

Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10

域配置指南



手册代码: C120-E680-16ZH
2017 年 7 月

版权所有© 2007, 2017, Fujitsu Limited。保留所有权利。

Oracle 和/或其附属公司对于本材料的部分内容提供了技术意见和审阅。

Oracle 和/或其附属公司与 Fujitsu Limited 各自拥有或控制与本文档所述产品和技术相关之知识产权，并且此类产品、技术及本文档受著作权法、专利和其他知识产权法及国际条约的保护。

本文档以及它所介绍的产品和技术按照限制其使用、复制、分发和反汇编的许可证进行分销。未经 Oracle 和/或其附属公司与 Fujitsu Limited 及其相应许可方（若有）的事先书面许可，严禁以任何形式、通过任何方式复制此产品或技术及本文档的任何部分。向您提供本文档并非授权您对于文档所述产品或技术之任何明示或默示之权利或许可，并且本文档不包含或表示 Oracle、Fujitsu Limited 或其各自任何附属公司方面的任何承诺。

本文档以及其所述的产品和技术可能融合了第三方的知识产权，这些产权的著作权属于这些供应商和/或是供应商授权给 Oracle 和/或其附属公司及 Fujitsu Limited，包括软件和字体技术。

依照 GPL 或 LGPL 的条款，在最终用户请求时，如果可行，则提供由 GPL 或 LGPL 管控的源代码的副本。请联系 Oracle 和/或其附属公司或 Fujitsu Limited。此分发可能包括第三方开发的材料。产品的一些部分可能源于 Berkeley BSD 系统，并获得了加州大学的授权。

UNIX 是 The Open Group 的注册商标。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。

Fujitsu 和 Fujitsu 徽标是 Fujitsu Limited 的注册商标。

SPARC Enterprise、SPARC64、SPARC64 徽标和所有 SPARC 商标是 SPARC International, Inc.在美国和其他国家/地区的商标或注册商标，并应按照许可证的规定使用。

其他名称可能是各自所有者的商标。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

免责声明：Oracle 和 Fujitsu Limited 和/或任何附属公司授予的与本文档或其所述任何产品或技术之仅有保证均在所提供相关产品或技术的许可证协议中明确说明。

除非此类协议明确规定，ORACLE 或 FUJITSU LIMITED 和/或其附属公司对于按原样提供的此类产品或技术或本文档不做任何类型（明示或默示）的陈述或保证。所有明示或默示之条件、陈述和保证，包括但不限于对适销性、特定用途适用性或非侵权的默示担保，均被免责，除非此类免责被法律视为无效。

除非此类协议明确规定，在适用法律允许的范围内，无论任何情况下，依据任何法理，Oracle 或 Fujitsu Limited 和/或其任何附属公司对于任何第三方的收入或利润损失、使用或数据损失、业务中断，或者任何间接、特殊、意外或连带损害，均不承担责任，即使事先被告知可能会发生此损害。

文档按原样提供，所有明示或默示之条件、陈述和保证，包括但不限于对适销性、特定用途适用性或非侵权的默示担保，均被免责，除非此类免责被法律视为无效。

目录

前言 xi

第1章 概括了解域配置 1

1.1 SPARC M12/M10 域配置 1

1.1.1 单机配置和模块化架构配置概述 1

1.1.2 逻辑域配置概述 3

1.1.3 域配置的特性 4

1.2 SPARC M12/M10 硬件资源 8

1.3 什么是物理分区? 11

1.3.1 了解物理分区组件 11

1.3.2 物理分区配置的操作 12

1.4 什么是逻辑域? 15

1.4.1 了解逻辑域组件 15

1.4.2 逻辑域配置的操作 16

1.4.3 重新配置物理 I/O 设备 18

1.5 什么是物理分区动态重新配置? 18

1.5.1 物理分区动态重新配置概述 18

1.5.2 使用物理分区动态重新配置 19

1.5.3 CPU 和内存的动态重新配置 19

1.5.4 软件条件和确认方法 20

1.6 混合使用 SPARC64 X+ 处理器和 SPARC64 X 处理器 22

1.6.1 CPU 操作类型和 CPU 运行模式 23

第2章 系统运行的条件和设置	25
2.1 XSCF 状态管理	25
2.1.1 物理分区状态	25
2.1.2 系统板状态	26
2.1.3 状态变化	27
2.2 XSCF 的条件和设置	31
2.2.1 XSCF 需要的配置条件	32
2.2.2 XSCF 的设置	32
2.2.3 内存镜像模式的注意事项	33
2.2.4 添加 CPU 模块时的注意事项	34
2.3 Oracle Solaris 的条件和设置	34
2.3.1 I/O 和软件要求	34
2.3.2 交换空间相关的注意事项	34
2.3.3 实时处理的注意事项	35
2.4 逻辑域的条件和设置	35
2.4.1 逻辑域配置注意事项	36
2.4.2 逻辑域重新配置注意事项	46
2.5 动态重新配置的条件和设置	48
2.5.1 动态重新配置系统配置时的注意事项	48
2.5.2 动态重新配置的系统操作注意事项	49
2.5.3 如何检查/设置 PPAR DR 模式	58
2.6 使用 SPARC64 X+ 处理器时的注意事项	60
2.6.1 更新 XCP 固件	60
2.6.2 CPU 运行模式设置	60
2.6.3 PPAR 内 SPARC64 X+ 处理器和 SPARC64 X 处理器的混合配置条件	64
2.6.4 CPU 运行模式和物理分区动态重新配置之间的关系	65
第3章 域配置的操作	67
3.1 与物理分区配置有关的操作和命令	67
3.1.1 检查物理分区配置信息	68
3.1.2 检查物理分区状态	70

3.1.3	检查系统板状态	71
3.1.4	检查设备设置信息	73
3.1.5	检查逻辑域状态	74
3.1.6	检查 CPU 激活密钥信息	76
3.1.7	检查 CPU 内核资源的使用状况	76
3.1.8	检查 CPU 激活注册和设置信息	78
3.1.9	检查物理分区运行模式	79
3.1.10	显示物理分区的逻辑域配置信息	80
3.1.11	设置内存镜像	81
3.1.12	设置物理分区配置信息	82
3.1.13	添加 CPU 激活密钥	83
3.1.14	分配物理分区中使用的 CPU 内核资源	84
3.1.15	添加系统板到模块化架构配置	86
3.1.16	删除模块化架构配置中的系统板	88
3.1.17	设置物理分区运行模式	90
3.1.18	指定物理分区的逻辑域配置信息	92
3.1.19	启动物理分区	92
3.1.20	停止物理分区	92
3.1.21	连接到控制域控制台	93
3.2	与逻辑域配置有关的操作和命令	93
3.2.1	确认逻辑域管理器正在运行	95
3.2.2	检查服务	95
3.2.3	基于 CPU 激活, 检查可分配的虚拟 CPU 的数量	96
3.2.4	检查资源的分配状况	96
3.2.5	检查资源的使用状况	96
3.2.6	检查各资源组的使用状况	98
3.2.7	检查各 CPU 插槽的资源使用状况	99
3.2.8	检查逻辑域状态	100
3.2.9	显示逻辑域配置信息	101
3.2.10	检查 I/O 设备的使用状况	101
3.2.11	启动延迟重新配置模式	103

3.2.12	设置默认服务	104
3.2.13	配置虚拟 CPU	105
3.2.14	配置虚拟内存	109
3.2.15	设置 CPU 插槽限制条件	112
3.2.16	设置逻辑域配置信息	112
3.2.17	创建逻辑域	113
3.2.18	配置 I/O 设备	113
3.2.19	创建或销毁 SR-IOV 虚拟功能	116
3.2.20	配置虚拟网络设备	119
3.2.21	配置虚拟磁盘服务器	120
3.2.22	配置虚拟磁盘	121
3.2.23	配置虚拟控制台	122
3.2.24	配置启动设备	123
3.2.25	绑定资源	123
3.2.26	启动来宾域	123
3.2.27	指定关闭组	124
3.2.28	重新配置设备	124
3.2.29	启用恢复模式	124
第4章	物理分区配置示例	125
4.1	物理分区配置流程	125
4.2	物理分区配置操作示例	126
第5章	逻辑域配置示例	133
5.1	逻辑域配置流程	133
5.2	逻辑域配置操作示例	134
5.2.1	登录控制域	135
5.2.2	设置默认服务	136
5.2.3	进行控制域的初始设置	137
5.2.4	配置来宾域	138
5.2.5	配置恢复模式	139
5.2.6	保存逻辑域配置信息	140
第6章	物理分区重新配置示例	141

6.1	物理分区重新配置流程	141
6.1.1	系统板添加流程	141
6.1.2	系统板删除流程	143
6.1.3	系统板移动流程	144
6.1.4	系统板更换流程	146
6.2	系统板添加示例	147
6.2.1	系统板分配示例	148
6.2.2	系统板整合示例	150
6.2.3	预约系统板整合的操作示例	153
6.3	删除系统板的操作示例	156
6.3.1	删除系统板分配的操作示例	156
6.3.2	释放系统板的操作示例	157
6.3.3	预约系统板取消分配的操作示例	161
6.4	移动系统板的操作示例	163
6.5	更换系统板的操作示例	169
6.6	变更物理分区编号注意事项	173
第7章	迁移来宾域	175
7.1	概述	175
7.1.1	实时迁移的要求	177
7.2	迁移来宾域	178
7.3	来宾域迁移示例	178
附录 A	使用物理分区动态重新配置的环境配置示例和步骤	181
A.1	将不支持物理分区动态重新配置的系统更新到支持物理分区动态重新配置的系统	182
A.1.1	配置示例	182
A.1.2	配置步骤	183
A.2	对于从 XCP 2220 或更高版本的新安装，逻辑域配置无可用硬件资源	201
A.2.1	配置示例	201
A.2.2	物理分区配置步骤示例	203

A.2.3	活动状态更换的步骤示例（适用于 Oracle VM Server for SPARC 3.1.x）	225
A.2.4	活动状态更换步骤示例（当 PCIe 总线的动态分配可用时）	247
A.3	对于从 XCP 2220 或更高版本的新安装，逻辑域配置有可用硬件资源	269
A.3.1	配置示例	269
A.3.2	物理分区配置步骤示例	271
A.3.3	活动状态更换的步骤示例（适用于 Oracle VM Server for SPARC 3.1.x）	296
A.3.4	活动状态更换步骤示例（当 PCIe 总线的动态分配可用时）	310
A.4	采用 XCP 2220 或更高版本新安装的系统从 1BB 至 2BB 配置的扩展	324
A.4.1	配置示例	325
A.4.2	扩展步骤（适用于 Oracle VM Server for SPARC 3.1.x）	327
A.4.3	扩展步骤（当 PCIe 总线的动态分配可用时）	343
A.5	向仅由 SPARC64 X 处理器配置的物理分区安装由 SPARC64X+ 处理器配置的系统板	359
A.5.1	配置示例	359
A.5.2	配置步骤	362
A.6	仅配置控制域的系统板的活动状态更换（适用于 Oracle VM Server for SPARC 3.2 或更高版本）	394
A.6.1	配置示例	394
A.6.2	物理分区配置步骤示例	395
A.6.3	活动状态更换的步骤示例	403
附录 B	使用动态重新配置补充资料	411
B.1	XSCF 重新启动或故障转移时间的注意事项	411
检查添加操作的完成状况		411
检查移除操作的完成状况		412
B.2	CPU 运行模式补充资料	413
B.3	其他注意事项	413
附录 C	信息含义及应对措施	417
C.1	命令信息	417

C.1.1	addboard	417
C.1.2	deleteboard	420
索引		425

前言

本文档介绍 Oracle 或 Fujitsu SPARC M12/M10 系统提供的域配置功能。本文档适用于精通计算机网络和 Oracle Solaris 的系统管理员。

在日本 Fujitsu 将 Fujitsu SPARC M12 作为 SPARC M12 销售。

Fujitsu SPARC M12 和 SPARC M12 是相同的产品。

在日本 Fujitsu 将 Fujitsu M10 作为 SPARC M10 销售。

Fujitsu M10 和 SPARC M10 是相同的产品。

适用对象

本文档适用于精通计算机网络和 Oracle Solaris 的系统管理员。

相关文档

在以下网址可以找到适用于您服务器的全部文档。

- Sun Oracle 软件相关文档（Oracle Solaris 等）

<http://docs.oracle.com/en/>

- Fujitsu 文档

全球网站

<http://www.fujitsu.com/global/products/computing/servers/unix/sparc/downloads/manuals/>

日语网站

<http://www.fujitsu.com/jp/products/computing/servers/unix/sparc/downloads/manual/>

有关使用 SPARC M12 的系统，请参见 "与 SPARC M12 相关的文档" 中列出的手册。

有关使用 SPARC M10 的系统，请参见 "与 SPARC M10 相关的文档" 中列出的手册。

与 SPARC M12 相关的文档

手册名称 (*1)
《Fujitsu SPARC M12 产品说明》
《Fujitsu SPARC M12 快速指南》
《Fujitsu SPARC M12 入门指南》 (*2)
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Important Legal and Safety Information》 (*2)
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Safety and Compliance Guide》
《Software License Conditions for Fujitsu SPARC M12 and Fujitsu M10/SPARC M10》
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Security Guide》
《Fujitsu SPARC Servers/SPARC Enterprise/PRIMEQUEST Common Installation Planning Manual》
《Fujitsu SPARC M12-2 安装指南》
《Fujitsu SPARC M12-2S 安装指南》
《Fujitsu SPARC M12 PCI Card Installation Guide》
《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》
《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 域配置指南》
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 RCIL User Guide》 (*3)
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF MIB and Trap Lists》
《Fujitsu SPARC M12-2/M12-2S Service Manual》
《Crossbar Box for Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Service Manual》
《PCI Expansion Unit for Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Service Manual》
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Glossary》
《External USB-DVD Drive user guide》

*1 列出的手册如有变更，恕不另行通知。

*2 本产品提供印刷的手册。

*3 本文档特别适用于 SPARC M12/M10 和 FUJITSU ETERNUS 磁盘存储系统。

与 SPARC M10 相关的文档

手册名称 (*1)
《Fujitsu M10/SPARC M10 系统 产品说明》
《Fujitsu M10/SPARC M10 系统 快速指南》
《Fujitsu M10/SPARC M10 系统 入门指南》 (*2)
《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Important Legal and Safety Information》 (*2)

手册名称 (*1)

- 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Safety and Compliance Guide》
- 《Software License Conditions for Fujitsu SPARC M12 and Fujitsu M10/SPARC M10》
- 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Security Guide》
- 《Fujitsu SPARC Servers/SPARC Enterprise/PRIMEQUEST Common Installation Planning Manual》
- 《Fujitsu M10-1/SPARC M10-1 安装指南》
- 《Fujitsu M10-4/SPARC M10-4 安装指南》
- 《Fujitsu M10-4S/SPARC M10-4S 安装指南》
- 《Fujitsu M10/SPARC M10 Systems PCI Card Installation Guide》
- 《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》
- 《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 域配置指南》
- 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 RCIL User Guide》 (*3)
- 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》
- 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF MIB and Trap Lists》
- 《Fujitsu M10-1/SPARC M10-1 Service Manual》
- 《Fujitsu M10-4/Fujitsu M10-4S/SPARC M10-4/SPARC M10-4S Service Manual》
- 《Crossbar Box for Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Service Manual》
- 《PCI Expansion Unit for Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Service Manual》
- 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Glossary》
- 《External USB-DVD Drive user guide》

*1 列出的手册如有变更，恕不另行通知。
*2 本产品提供印刷的手册。
*3 本文档特别适用于 SPARC M12/M10 和 FUJITSU ETERNUS 磁盘存储系统。

安全注意事项

- 在使用或处理 SPARC M12/M10 之前，请仔细阅读以下文档：
- 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Important Legal and Safety Information》
 - 《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Safety and Compliance Guide》

文本惯例

本手册使用以下字体和符号来表示特定类型的信息。

字体/符号	含义	示例
AaBbCc123	您输入的内容，对比屏幕上的计算机输出内容。 本字体用于表示命令输入。	XSCF> adduser jsmith
AaBbCc123	命令、文件和目录的名称；屏幕上的计算机输出。 此字体表示文框内的命令输出示例。	XSCF> showuser -P User Name: jsmith Privileges: useradm auditadm
《 》	表示参考手册的名称。	请参阅《Fujitsu M10-1/SPARC M10-1 安装指南》。
" "	表示章节名、项目、按钮或菜单。	请参阅 "第 2 章 网络连接"。

文本中的命令语法

当 XSCF 命令有 (8) 或 (1) 的节号时，从文本中被省略。
有关命令的详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

命令行界面 (CLI) 语法

命令语法如下：

- 必须将要求输入一个值的变量设为斜体。
- 可选元素必须括在 [] 中。
- 可选关键词的一组选项必须括在 [] 中并由 | 隔开。

文档的反馈

如果您对本文档有任何意见建议或要求，请抽出几分钟时间告知我们，指明手册代码、手册标题和页码，并在以下网站陈述您的观点：

- 全球网站
<http://www.fujitsu.com/global/contact/>
- 日语网站

<http://www.fujitsu.com/jp/products/computing/servers/unix/sparc/contact/>

概括了解域配置

本章概括介绍域配置。

- SPARC M12/M10 域配置
- SPARC M12/M10 硬件资源
- 什么是物理分区？
- 什么是逻辑域？
- 什么是物理分区动态重新配置？
- 混合使用 SPARC64 X+ 处理器和 SPARC64 X 处理器

1.1 SPARC M12/M10 域配置

本节介绍使用 SPARC M12/M10 可构建的域配置的概述和功能。

1.1.1 单机配置和模块化架构配置概述

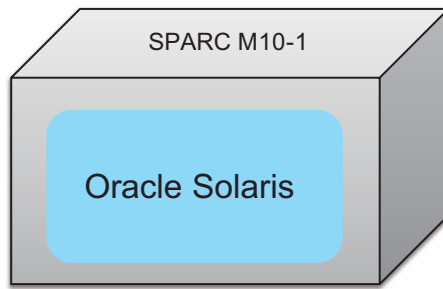
本节简要介绍由一个 SPARC M12/M10 单元组成的系统以及由多个 SPARC M12 或 SPARC M10 单元（具有模块化架构配置的系统）组成的系统。

- 单机配置的系统
- 模块化架构配置的系统 (SPARC M12-2S/M10-4S)

单机配置的系统

图 1-1 显示了 SPARC M10-1 的系统配置示例。在此示例中，系统上运行了一个 Oracle Solaris。

图 1-1 单机配置的系统（示例：SPARC M10-1）



模块化架构配置的系统

使用 SPARC M12-2S 或 SPARC M10-4S，模块化架构 (BB) 配置的系统可以连接多个 SPARC M12-2S 单元或 SPARC M10-4S 单元。该系统可以构建为一个独立的系统，或以 BB 为单位分成多个物理分区。

对于具有模块化架构配置的系统，多个 BB 可以包括单个物理分区。

注 - 对于模块化架构方法，SPARC M12-2S 和 SPARC M10-4S 不能混合构成系统。

图 1-2 显示了多个 SPARC M12-2S 单元的系统配置示例。在此示例中，系统上多个 SPARC M12-2S 单元运行一个 Oracle Solaris。

图 1-2 模块化架构配置的系统（不划分物理分区）

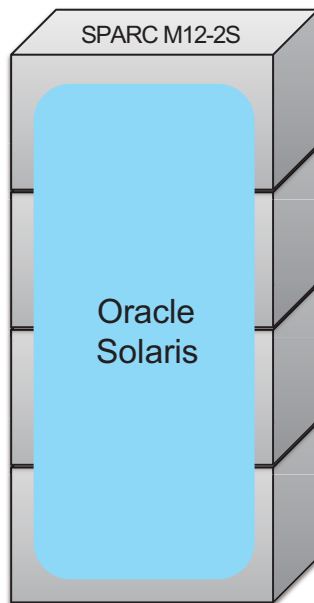
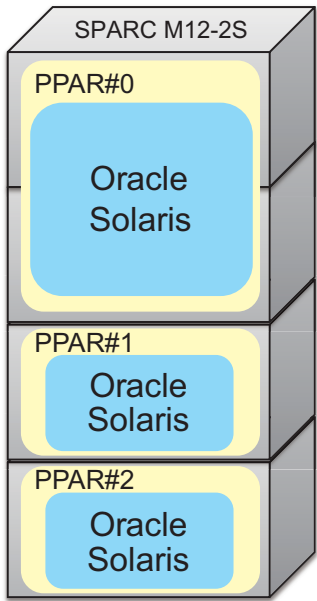


图 1-3 显示了四个 SPARC M12-2S 单元连接成划分为三个物理分区的系统的示例。在此示例中，系统划分为多个物理分区，每个分区运行一个 Oracle Solaris。

图 1-3 模块化架构配置的系统（划分物理分区）



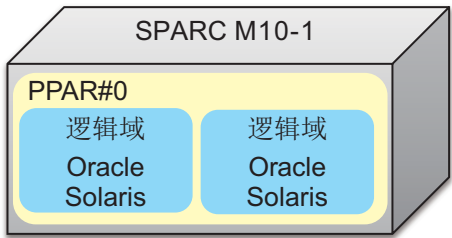
1.1.2 逻辑域配置概述

具有多个逻辑域的配置

可以在单个物理分区中构建多个逻辑域。

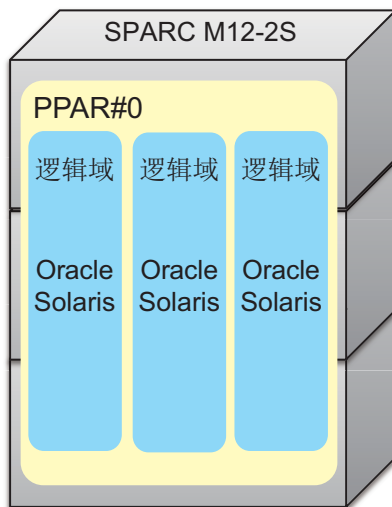
在单个物理分区中构建多个逻辑域可以使在每个逻辑域上运行一个 Oracle Solaris 成为可能。图 1-4 显示了 SPARC M10-1 的系统配置示例。此示例显示了具有在单个物理分区中构建的两个逻辑域的系统配置。

图 1-4 具有多个逻辑域的配置（单机配置）



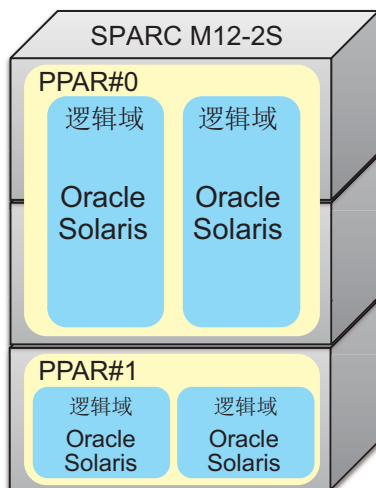
在模块化架构配置的系统构建多个逻辑域可以使在每个逻辑域上运行 Oracle Solaris 成为可能。图 1-5 显示了在一个物理分区中连接了三个 SPARC M12-2S 单元并构建了三个逻辑域的模块化架构配置的系统示例。

图 1-5 具有多个逻辑域的配置（模块化架构配置 -1）



模块化架构配置的系统可以划分为多个物理分区，每个物理分区上配置了多个逻辑域。此外，Oracle Solaris 可以在每个逻辑域上运行。图 1-6 显示了在两个物理分区中的每一个中连接了三个 SPARC M12-2S 单元并构建了两个逻辑域的模块化架构配置的系统示例。

图 1-6 具有多个逻辑域的配置（模块化架构配置 -2）



1.1.3 域配置的特性

本节介绍 SPARC M12/M10 系统的域配置、特性。

由逻辑域组成的系统的特性

SPARC M12/M10 系统可使用 Oracle VM Server for SPARC 构建多个逻辑域，并且允许独立的 OS 在每个逻辑域上运行。不同的逻辑域可以分别运行不同版本的操作系统。CPU、内存和 I/O 可以灵活分配给每个逻辑域，从而有效使用硬件资源。

图 1-7 显示了两个 CPU 内核和一部分内存资源在不同版本的 Oracle Solaris 实例之间移动的示例。

图 1-7 逻辑域之间硬件资源的有效使用（使用 SPARC M10-1 的单机配置示例）

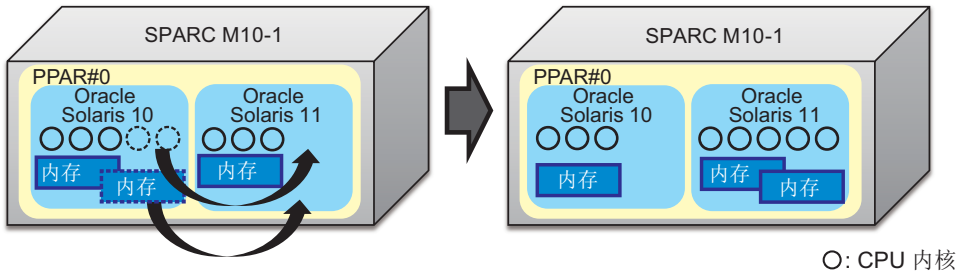
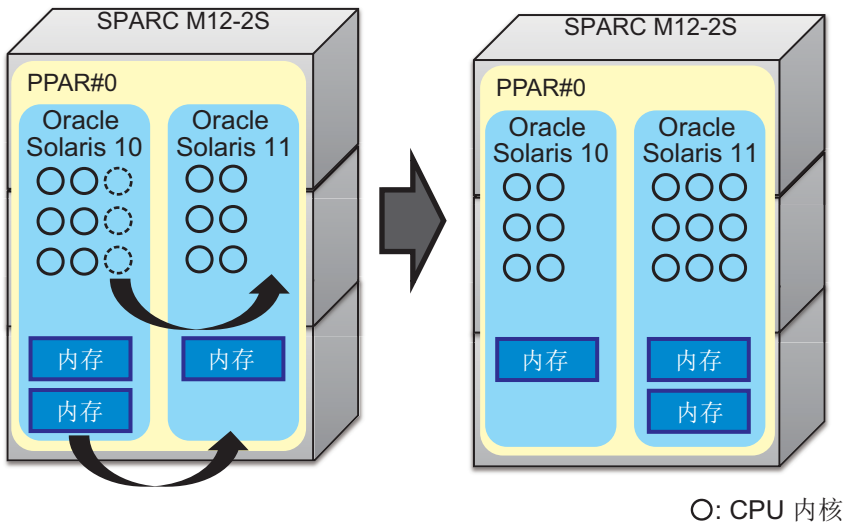


图 1-8 显示了三个 CPU 内核和一个内存资源在不同版本的 Oracle Solaris 实例之间移动的示例。

图 1-8 逻辑域之间硬件资源的有效使用（模块化架构配置）



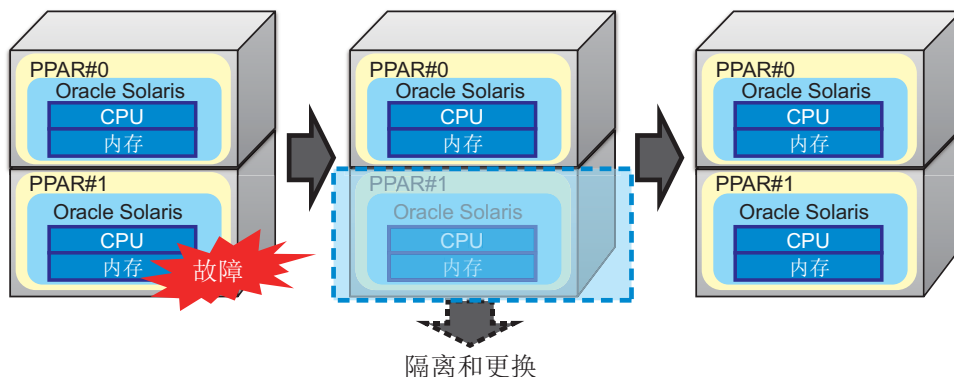
由物理分区组成的系统的特性

SPARC M12-2S/M10-4S 系统允许物理分区配置一个或多个机箱并允许独立的 OS 在每个物理分区上运行。

由物理分区组成的系统比由使用 Oracle VM Sever for SPARC 的逻辑域组成的系统提供了更高的硬件独立性。

图 1-9 显示了故障物理分区被隔离和更换以恢复故障之前配置的示例。

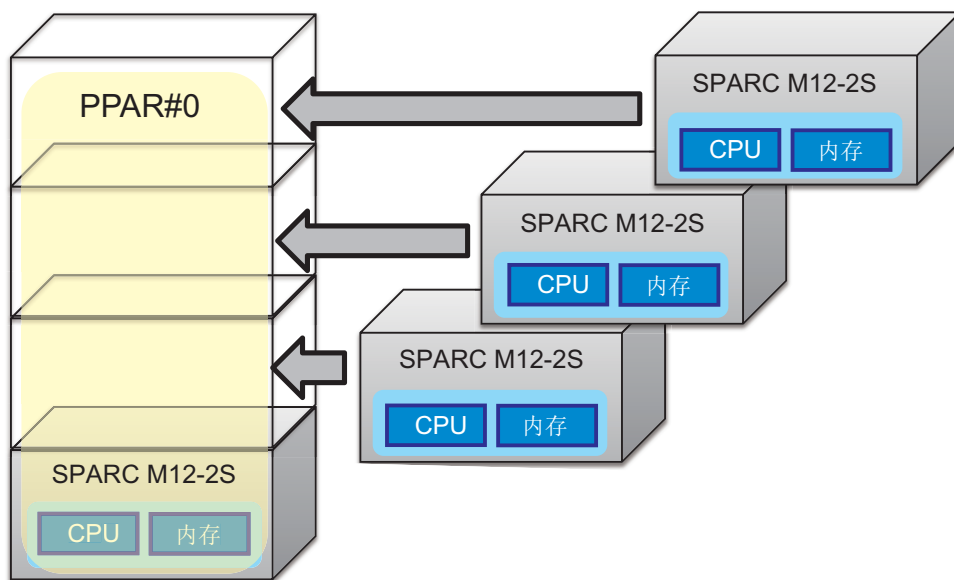
图 1-9 物理分区的特性（硬件独立）



SPARC M12-2S/M10-4S 系统允许通过堆叠多个机箱来扩展硬件资源。这使得能够根据环境的需要逐渐扩展小型系统。

图 1-10 显示了添加三个 SPARC M12-2S 单元并将四个单元用作单个物理分区的示例。

图 1-10 物理分区的特性（硬件可扩展性）

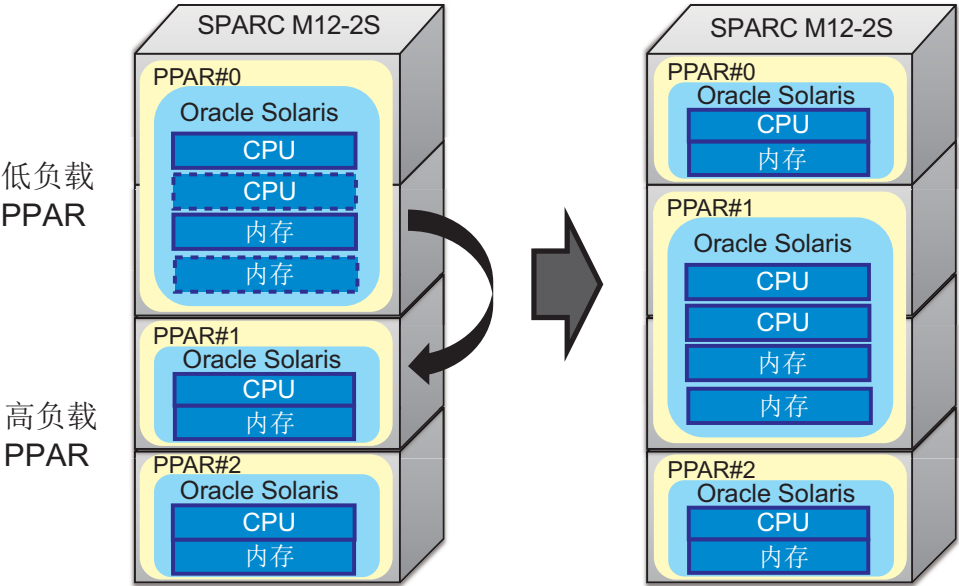


此外，CPU、内存和 I/O 可以以模块化架构模块 (PSB) 为单位添加到物理分区，也可从其移除。

此外，由多个物理分区组成的系统允许以集中的方式管理这些物理分区，并且硬件资源可以从低负载的物理分区移动到高负载的物理分区。

图 1-11 显示了一个机箱的资源从 PPAR#0 移到 PPAR#1 的示例。

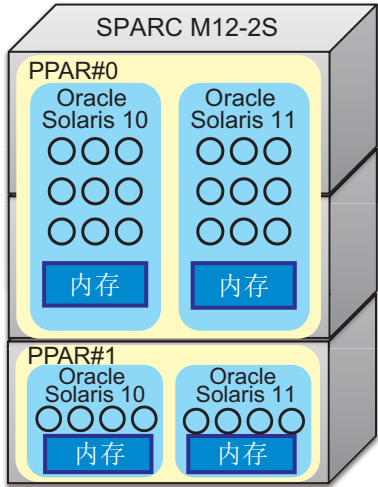
图 1-11 物理分区的特性（硬件灵活性）



此外，可以在每个物理分区中构建逻辑域。

图 1-12 显示了在两个物理分区中的每一个中连接了三个 SPARC M12-2S 单元并构建了两个逻辑域的示例。

图 1-12 在物理分区中构建逻辑域的示例



○: CPU 内核

1.2 SPARC M12/M10 硬件资源

本节介绍 SPARC M12/M10 系统的硬件资源。

SPARC M12/M10 系统允许灵活分配 CPU、内存和 I/O 等硬件资源，从而有效使用这些资源。

图 1-13 显示了有效使用 SPARC M12-1/M10-1 硬件资源的示例。

图 1-13 有效使用硬件资源 (SPARC M12-1/M10-1)

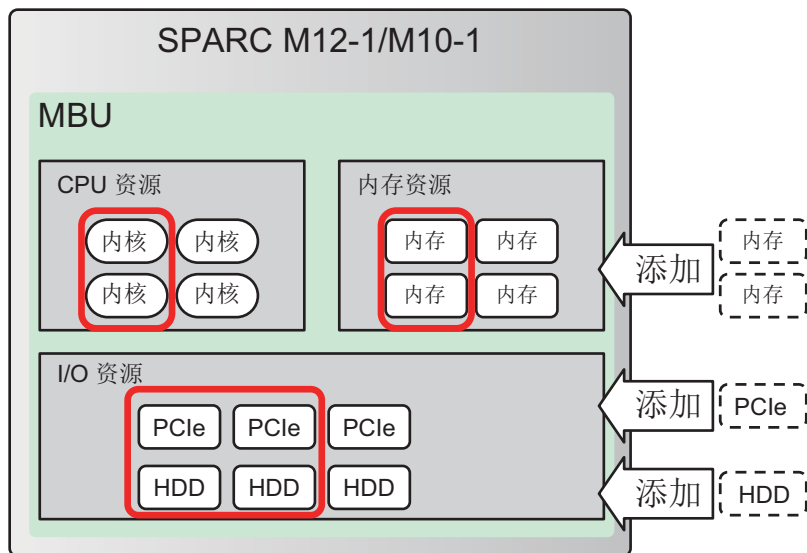
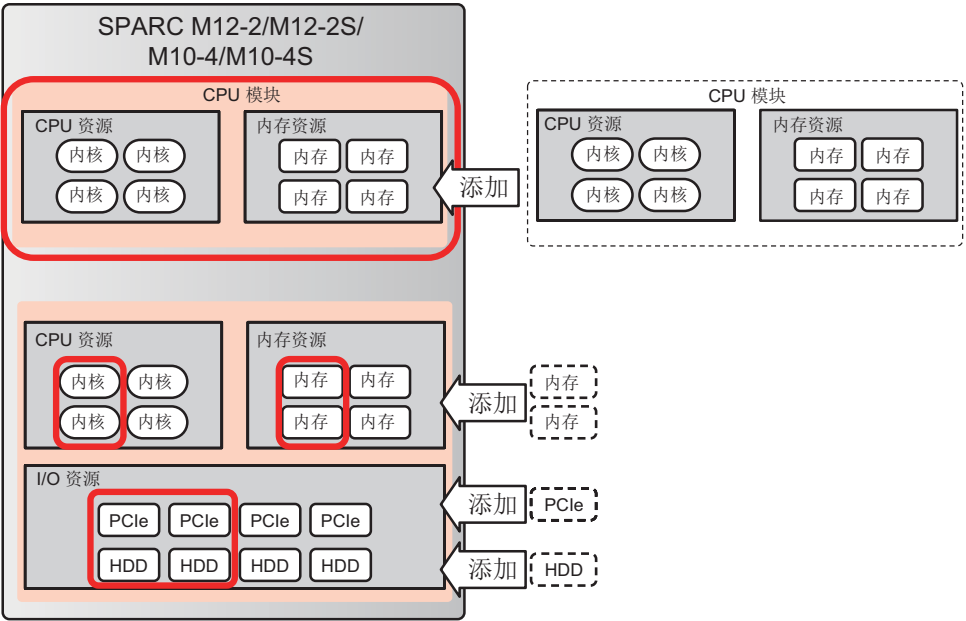


图 1-14 显示了 SPARC M12-2/M12-2S/M10-4/M10-4S 硬件资源有效使用的示例。

图 1-14 有效使用硬件资源 (SPARC M12-2/M12-2S/M10-4/M10-4S)



- **CPU 资源**
[SPARC M12-1]
CPU 激活提供的资源最多为 6 个 CPU 内核。
[SPARC M12-2/M12-2S]
CPU 激活提供的资源最多为 24 个 CPU 内核（2-CPU 配置）。
[SPARC M10-1]
CPU 激活提供的资源最多为 16 个 CPU 内核。
[SPARC M10-4/M10-4S]
CPU 激活提供的资源最多为 32 个 CPU 内核（4-CPU 配置）。
- **内存资源**
可以对所有型号配置内存镜像。在这种情况下，可用内存的数量减半。
[SPARC M12-1]
最多可安装 1 TB。
[SPARC M12-2/M12-2S]
最多可安装 2 TB（2-CPU 配置）。
[SPARC M10-1]
最多可安装 1 TB。
[SPARC M10-4/M10-4S]
最多可安装 4 TB（4-CPU 配置）。

■ I/O 资源

[SPARC M12-1]

最多可安装三个 PCIe 卡。通过连接 PCI 扩展单元，可以安装多达 33 个 PCIe 卡。最多可安装八个内部存储单元。

[SPARC M12-2/M12-2S]

SPARC M12-2/M10-4 最多可以安装 11 个 PCIe 卡。通过连接 PCI 扩展单元，可以安装多达 71 个 PCIe 卡。最多可安装八个内部存储单元。

使用 PHP 可以动态增加 PCIe 卡的数量。

[SPARC M10-4/M10-4S]

SPARC M12-2S/M10-4S 最多可以安装八个 PCIe 卡。通过连接 PCI 扩展单元，可以安装多达 58 个 PCIe 卡。最多可安装八个内部存储单元。

使用 PHP 可以动态增加 PCIe 卡的数量。

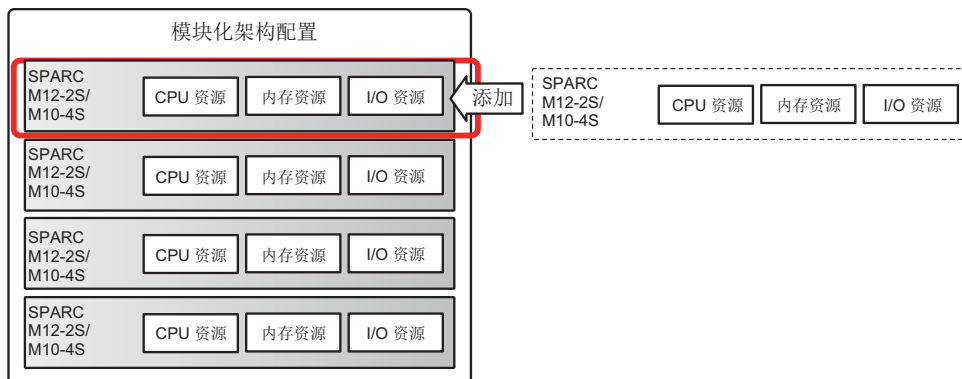
[SPARC M10-1]

SPARC M10-1 最多可以安装三个 PCIe 卡。通过连接 PCI 扩展单元，可以安装多达 23 个 PCIe 卡。最多可安装八个内部存储单元。

■ 模块化架构配置 (SPARC M12-2S/M10-4S)

由多个 SPARC M12-2S/M10-4S 单元组成的模块化架构配置的系统可以通过堆叠模块从 1 个模块逐渐扩展到 16 个模块，并将机箱用作模块。

图 1-15 模块化架构配置中硬件资源的有效使用



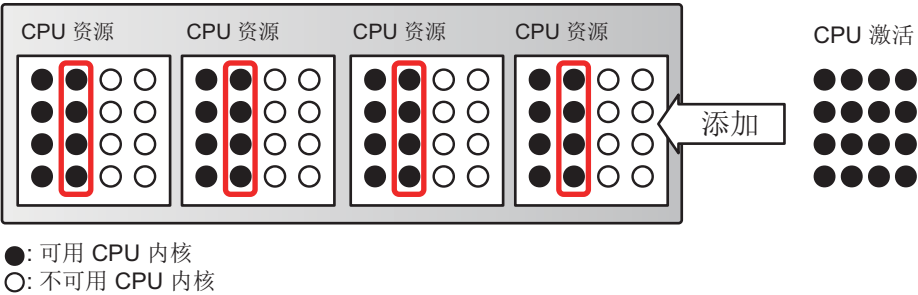
■ CPU 激活

CPU 激活功能可用于以比 CPU 芯片更小的 CPU 内核来增加 CPU 资源。

您可以从最低要求的 CPU 资源开始，并通过添加 CPU 激活逐渐扩大 CPU 资源。

有关 CPU 激活的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "第5章 CPU 激活"。

图 1-16 CPU 激活（SPARC M10-4 / M10-4S 的示例）



1.3 什么是物理分区？

本节介绍物理分区的组件以及物理分区相关的操作。

1.3.1 了解物理分区组件

物理分区由多个模块化架构组成。在不是模块化架构配置的配置中，一个服务器就是一个物理分区。配置物理分区时，XSCF 固件将一个模块化架构模块作为一个物理系统板 (PSB) 处理。物理系统板由三种类型的资源组成：CPU、内存和 I/O 设备。在物理分区内可以将物理系统板的硬件资源分配给逻辑系统板 (LSB)。

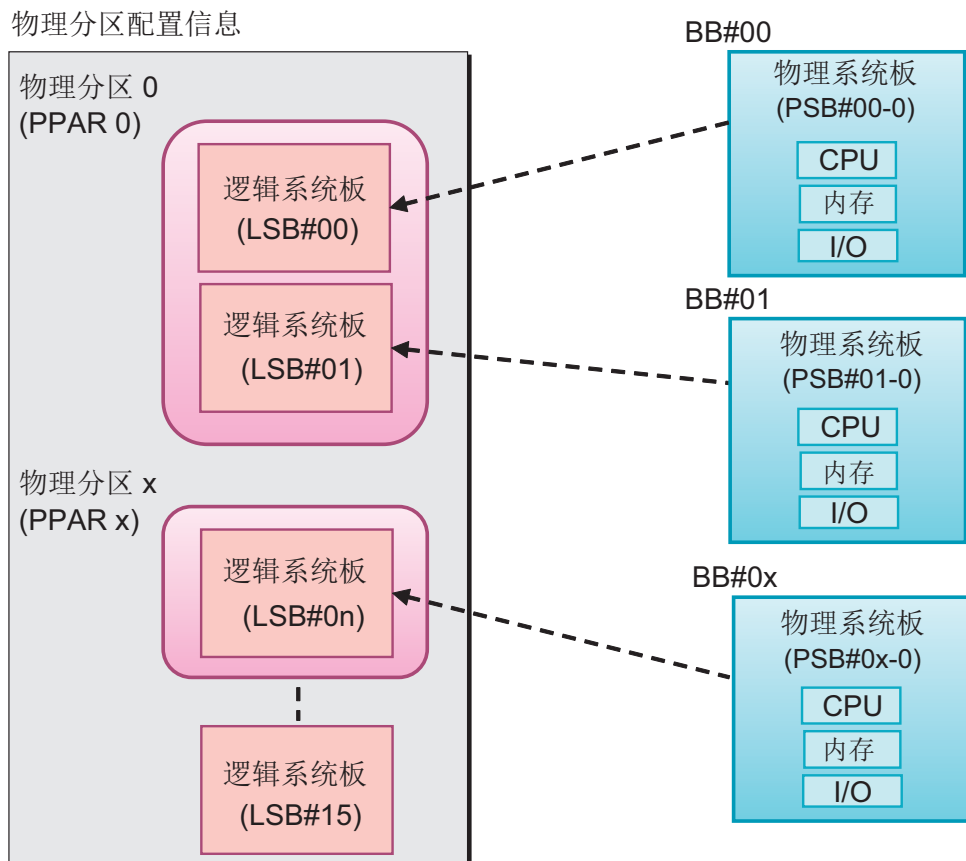
- 物理系统板 (PSB)
PSB 由安装在一个 SPARC M12/M10 机箱中的物理组件（CPU、内存和 I/O）组成。在 SPARC M12-1/M10-1 中，物理系统板是主板单元。在 SPARC M12-2/M12-2S/M10-4/M10-4S 中，物理系统板是 CPU 内存单元（包括下部 (CMUL) 及上部 (CMUU)）。物理系统板可用作代表机箱的单元，维护时用于添加/移除/更换 SPARC M12/M10 机箱。对于模块化架构配置的系统，物理系统板是指模块化架构 (BB)。
- 逻辑系统板 (LSB)
LSB 是分配给物理系统板的逻辑单元的名称。每个物理分区有一组分配给它的逻辑系统板。将一个 PSB 编号分配给物理分区中的逻辑系统板，即可在系统中对其进行识别。
LSB 号用于控制对于物理分区的资源（如内存）分配。
- 物理分区编号 (PPAR-ID)
PPAR-ID 是识别物理分区的编号。您可以分配 0 到 15 之间的一个数字。但是，对于物理分区的编号有以下限制条件：
 - 对于 SPARC M12-1/M12-2/M10-1/M10-4
物理分区编号固定为 0。您不能变更该数值。
 - 对于 SPARC M12-2S/M10-4S

您可以给物理分区编号分配与系统上所装机箱的 BB-ID 编号相同的号码。如果将未安装的机箱的 BB-ID 指定为物理分区编号，则该物理分区的上电将失败。

[示例] 对于安装有 BB#0 和 BB#1 的机箱，可将 0 或 1 分配为物理分区编号。如果您指定 2 作为物理分区编号，则该物理分区的上电将失败。

将物理分区上的逻辑和物理系统板之间的映射在物理分区配置信息（PPAR 配置信息）中进行设置。图 1-17 是这种映射的概念图。

图 1-17 逻辑系统板和物理系统板之间映射的概念图



作为动态重新配置目标的所有系统板必须在相关物理分区的物理分区配置信息（PPAR 配置信息）中进行登录。使用 `setpcl` 命令在 PPAR 配置信息中登录。

PPAR 配置信息将构成物理分区的逻辑系统板 (LSB) 映射到作为硬件资源信息的物理系统板 (PSB)，并使用 XSCF 进行管理。每条 PPAR 配置信息也存储了有关映射到物理分区和 LSB 的 PSB 的硬件资源的设置信息。

1.3.2 物理分区配置的操作

本节介绍如何操作具有多个 SPARC M12-2S 或 M10-4S 单元的物理分区配置。

如需设置物理分区配置并操作物理分区，请使用 XSCF 提供的 XSCF shell 和 XSCF Web。由 XSCF 对通过此类界面的操作进行集中管理。此外，XSCF 安全管理仅允许具有特定访问权限的管理员来执行相关物理分区配置的操作。有关物理分区配置的 XSCF shell 命令的详情，请参见“3.1 与物理分区配置有关的操作和命令”。有关 XSCF Web 的操作，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的“第3章 配置系统”。

登录/释放系统板

“登录系统板”是指提供以逻辑系统板编号（LSB 编号）显示一个模块化架构模块的物理系统板 (PSB) 并且在物理分区配置信息中对其进行设置。

“释放系统板”是指从物理分区配置信息中删除 PSB (BB)。物理系统板 (PSB) 在物理分区配置信息的登录可以使您能够执行 PSB (BB) 的操作，如将 PSB (BB) 分配和整合到物理分区。

添加系统板

“添加系统板”是指将已安装的模块化架构模块 (PSB) 或不属于任何物理分区的未使用的 PSB (BB) 整合到物理分区。添加 PSB (BB) 的过程将逐步执行，首先使用 "connect"，然后 "configure"。当您添加 PSB (BB) 时，指定的 PSB (BB) 首先连接到相关的物理分区。这之后是整合 PSB (BB) 的过程。PSB (BB) 的添加就此完成。

删除系统板

删除系统板是指从物理分区释放无需再配置的模块化架构模块 (PSB)。删除 PSB (BB) 的过程将逐步执行，首先使用 "unconfigure"，然后 "disconnect"。

将 PSB (BB) 分配到不同的物理分区，在删除过程中需要包含取消分配的操作。当您删除 PSB (BB) 时，将首先释放指定的 PSB (BB)。然后，从其所属的物理分区断开连接。PSB (BB) 的删除就此完成。

分配/取消分配系统板

“分配系统板”是指分配已安装的模块化架构模块 (PSB) 或不属于任何物理分区的未使用的 PSB (BB)，使其属于某个物理分区。“取消分配系统板”是指从其所属的物理分区释放模块化架构模块或 PSB。

将 PSB (BB) 分配到相关的物理分区将禁用不同物理分区上的 PSB (BB) 的操作。当您 将 PSB (BB) 分配到物理分区并且开启物理分区的电源时，将添加 PSB (BB)。当您关闭物理分区的电源时，将删除 PSB (BB) 并分配给物理分区。

当您将已经分配的系统板从其物理分区取消分配时，该系统板将不再属于任何物理分区，可以将其分配到不同的物理分区。

更换系统板

“更换系统板”是指为了维护而更换模块化架构模块 (PSB)。更换 CPU、内存和 I/O 设备的硬件资源时，您可以使用此操作。更换 PSB (BB) 的过程将逐步执行。当您更换 PSB (BB) 时，将首先删除指定的 PSB (BB)。完成删除 PSB (BB) 后，PSB (BB) 将移除准备就绪。此时可移除 PSB (BB)。然后，执行组件更换等的操作后，重新安装新的 PSB (BB) 从而进行更换操作。在此之后，安装完成后，PSB (BB) 的添加将完成更换操作。

系统板池功能

系统板池功能是一种使指定的 (PSB<BB>) 进入不属于任何物理分区的状态的功能。此功能对于根据需要在多个物理分区之间移动 PSB (BB) 有效。例如，您可以从系统板池获取一个 PSB (BB)，然后将其添加到一个 CPU 和内存负荷高的物理分区。您也可以在其不再被需要时将系统板返回系统板池。

只有在 PSB (BB) 被登录在物理分区配置信息（PPAR 配置信息）中时，您才可以将汇集的系统板分配到物理分区。换句话说，通过在多个 PPAR 的配置信息中登录相同的 PSB (BB)，您可以根据物理分区的运行状态灵活地通过整合或释放 PSB (BB) 来操作系统。然而，对于此类操作必须合理地管理汇集的 PSB (BB) 的使用。

您也可以将此功能与 "2.2 XSCF 的条件和设置" 中描述的内存无效选项和 I/O 无效选项相结合，从而方便地添加和删除 PSB (BB)。

预约物理分区配置变更

您不仅可以动态地在物理分区中添加或删除系统板 (PSB<BB>)，而且可以在该物理分区的下一个开启电源/关闭电源或重新启动时预约重新配置。

例如，您可以预约下列场合下的物理分区配置变更。

- 当动态重新配置由于任务和操作的便利性无法重新配置硬件资源时。
- 当您不需要立即变更物理分区配置时。
- 当您需要阻止动态重新配置变更设置和需要在物理分区重新启动后立即变更配置时：举例说明，当您删除设有不支持动态重新配置的驱动器或 PCI 卡的 PSB (BB) 时。

表 1-1 列出了组成物理分区的系统板 (PSB<BB>) 上可用的操作。

表 1-1 系统板上可用操作

术语	说明
登录	将系统板登录在物理分区配置信息中
释放	将系统板的登录从物理分区配置信息中释放
添加	将系统板添加到物理分区
删除	从物理分区删除系统板
分配	将系统板分配给物理分区
取消分配	从物理分区取消分配系统板
连接	将系统板连接到物理分区
断开	从物理分区断开系统板连接
配置	将系统板整合到物理分区配置
取消配置	从物理分区配置释放系统板
预约	物理分区电源打开或重新启动时，为将系统板整合到物理分区，预约物理分区。此操作同样预约物理分区中系统板的取消分配。
安装	将 SPARC M12/M10 机箱插入系统
移除	将 SPARC M12/M10 机箱从系统中移除
更换	移除 SPARC M12/M10 机箱进行维护检查等以及重新安装机箱或安装一个新机箱

1.4 什么是逻辑域？

本节介绍逻辑域的组件和逻辑域相关的操作。

1.4.1 了解逻辑域组件

逻辑域由虚拟 CPU、虚拟内存和虚拟 I/O 组成。

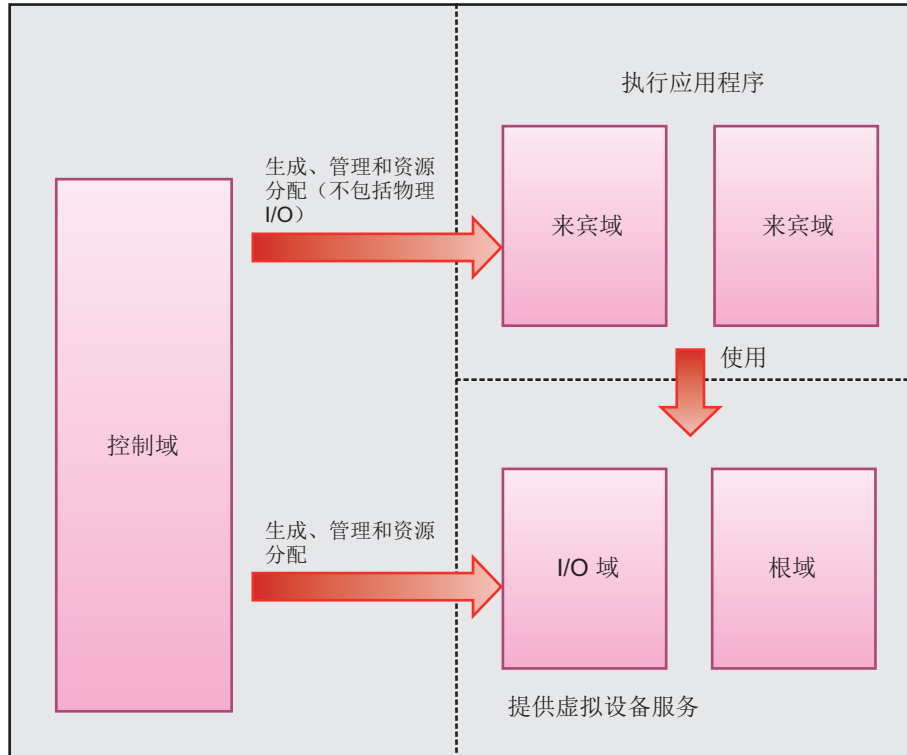
- 虚拟 CPU
可以将 CPU 分配给虚拟 CPU 单元中的逻辑域（线程）。在 SPARC M12/M10 系统中，一个物理 CPU（即 1 个插槽）有多个内核，每个有几个线程。这意味着物理 CPU 有与线程数相当的多个虚拟 CPU。可以将这些虚拟 CPU 分配给逻辑域。通常，考虑到逻辑域性能，将虚拟 CPU 以内核为单元分配给逻辑域。
- 虚拟内存
可以将内存以 256 MB 为单元分配给逻辑域。
- 虚拟 I/O
可以将 I/O 以虚拟 I/O 为单元分配给逻辑域。例如，虚拟磁盘是一个虚拟 I/O，可以是：
 - 物理磁盘
 - 物理磁盘切片
 - ZFS、UFS 等文件系统中的文件
 - ZFS 或其它卷管理器中的逻辑卷

根据所起的作用将逻辑域划分为：

- 控制域
控制域是创建和管理其它逻辑域，并将资源分配给逻辑域的逻辑域。在各物理分区中仅有一个控制域。Oracle VM Server for SPARC 已安装到运行着管理软件 Logical Domains Manager（逻辑域管理器）的控制域上。
- I/O 域
I/O 域 是提供虚拟设备服务的逻辑域。虚拟设备服务包括磁盘、网络和控制台。
- 根域
根域是已分配给 PCIe 根合成体的 I/O 域。PCIe 根合成体是整个 PCIe 总线，包括 PCIe 总线、所有 PCI 开关和设备。根域拥有物理 I/O 设备并直接访问它们。
- 来宾域
来宾域是由控制域控制的逻辑域。它使用 I/O 域中的虚拟设备服务。通常，中间件和应用程序在来宾域上运行。Oracle Solaris 独立在来宾域上运行，这样就可以不影响其它来宾域而启动和停止。可以在来宾域中动态添加和删除虚拟 CPU、虚拟内存或虚拟 I/O。
- 服务域
服务域是给来宾域提供服务的域的通用术语。尤其包括 I/O 和根域。

图 1-18 逻辑域之间关系的概念

物理分区



1.4.2 逻辑域配置的操作

Oracle VM Server for SPARC 的管理软件 Logical Domains Manager (逻辑域管理器) 用于执行逻辑域操作。您可以使用 Logical Domains Manager 配置或重新配置逻辑域。通过登录逻辑域中的控制域并使用 `ldm` 命令以及各种选项可执行这些操作。

本节介绍可用的逻辑域操作。

配置逻辑域

配置逻辑域时，使用 Oracle VM Server for SPARC 的管理软件：Logical Domains Manager (逻辑域管理器) 的 `ldm` 命令来执行以下操作。

- 创建逻辑域。
- 将硬件资源（如：虚拟 CPU、虚拟内存和虚拟 I/O）分配给逻辑域。
- 保存逻辑域配置信息。
- 启动/停止逻辑域。

对于除控制域以外的逻辑域，在操作系统和应用程序都在运行时，您也可以添加或删除

虚拟 CPU、内存和 I/O 硬件资源。这样的重新配置被称为逻辑域动态重新配置。然而，逻辑域动态重新配置的可用性取决于在逻辑域中运行的操作系统和应用程序。

可重新配置的硬件资源

可以对逻辑域中的以下硬件资源重新配置。

- 虚拟 CPU
CPU 可以虚拟 CPU（线程）为单元重新配置。但是，基于性能考虑，通常以内核为单元重新配置 CPU。
- 虚拟内存
可以 256 MB 为单元重新配置内存。
- 虚拟 I/O
虚拟 I/O 可在虚拟磁盘或虚拟网络单元内重新配置。重新配置分配给来宾域的虚拟 I/O 时，您不必重新启动来宾域。
在移除虚拟磁盘或虚拟网络之前，您需要执行以下任务。
 - 移除虚拟磁盘
卸载虚拟磁盘后，将其移除。
 - 移除虚拟网络
对虚拟网络接口执行取消连接后，移除虚拟网络。
- 物理 I/O
可将物理 I/O 在下述单元内分配给逻辑域。

表 1-2 Oracle VM Server for SPARC 的版本和可重新配置的物理 I/O

物理 I/O 的类型	静态重新配置	动态重新配置
PCIe SR-IOV 的虚拟功能 (Single Root I/O Virtualization)	3.0 或更高版本	3.1 或更高版本
PCIe 端点	3.0 或更高版本	3.1.1.1 或更高版本
PCIe 根合成体	3.1 或更高版本	3.2 或更高版本 (*1)

*1 Oracle Solaris 11.2 SRU11.2.8.4.0 或更高版本可添加或删除 PCIe 根合成体的逻辑域。

- 对于使用 SR-IOV、PCIe 端点分配设备（包括板载设备）的要求以及根合成体（PCIe 总线）的分配要求，请参见以下手册。
- 服务器的最新《产品说明》
 - 服务器的《PCI Card Installation Guide》
 - 所使用版本的《Oracle VM Server for SPARC Administration Guide》
 - 所使用版本的《Oracle VM Server for SPARC Release Notes》
- 静态重新配置分配给控制域和根域的物理 I/O 时，请切换到延迟重新配置模式。然后，分别重新启动控制域和根域，并应用重新配置的设置。
- 删除物理 I/O 之前，需要执行以下任务。
- 删除物理 I/O
如果是磁盘，将磁盘卸载，如果是网络，将网络取消连接。然后，在将物理 I/O 设备设置为不使用后，将其删除。
若需动态移动相关来宾域的硬件资源，Logical Domains DR 守护程序 (drd) 必须

在来宾域中运行。

1.4.3 重新配置物理 I/O 设备

添加或删除物理 I/O 设备

每个物理 I/O 设备的动态重新配置的需求因 Oracle VM Server for SPARC 的版本而异。有关详情，请参见“表 1-2”。

- 在控制域中添加或删除物理 I/O 设备之前，需要进入延迟重新配置模式。
- 在根域中添加或删除物理 I/O 设备之前，需要进入延迟重新配置模式或关闭根域 (ldm stop-domain)。
- 在 I/O 域中添加或删除物理 I/O 设备之前，需要关闭 I/O 域 (ldm stop-domain)。

设备驱动程序

需要动态重新配置的物理分区中的控制物理 I/O 设备的驱动程序必须支持 Oracle Solaris 的挂起和恢复功能。

支持 SR-IOV

当与 Oracle VM Server for SPARC 结合起来时，SPARC M12/M10 可以支持 PCIe SR-IOV 功能（单根 I/O 虚拟化功能）。

使用 SR-IOV 功能，您可以通过使用支持 SR-IOV 的 PCI Express 卡，针对一个物理功能创建多个虚拟功能。将所创建的虚拟功能分配给 I/O 域，可以更有效地在逻辑域共享 PCIe 卡带宽，从而减少服务域的系统开销。有关 SR-IOV 功能的详情，请参见

《Oracle VM Server for SPARC Administration Guide》和您的服务器的《PCI Card Installation Guide》。此外，关于可用的 Oracle Solaris 版本，请参见《Oracle VM Server for SPARC Release Notes》和《Oracle VM Server for SPARC Administration Guide》。

1.5 什么是物理分区动态重新配置？

本节简要介绍物理分区动态重新配置 (PPAR DR)。

1.5.1 物理分区动态重新配置概述

您可以使用 SPARC M12-2S 或 SPARC M10-4S 单元的模块化架构法配置物理分区。物理分区动态重新配置 (PPAR DR) 功能是一种将硬件资源（CPU、内存和 I/O）添加到物理分区和从物理分区移除的技术。此功能无需停止物理分区上的任何任务就可以起作用。

添加或移除的单元是一个 SPARC M12-2S 或 SPARC M10-4S（模块化架构模块 <BB>）。固件将本机作为系统板 (PSB) 进行处理。安装在 PSB (BB) 上的硬件资源可以在逻辑上并且动态地重新配置。

1.5.2 使用物理分区动态重新配置

物理分区动态重新配置功能（PPAR DR 功能）可用于如下情况：

- 增加硬件资源
为了应对任务扩大和系统负荷增加，您可以将系统板 (PSB<BB>) 添加到物理分区而无需停止物理分区上的任何任务。如果有必要，必须购买并配置额外的 CPU 激活。
- 减少硬件资源
若需根据任务规模的变更来有效地使用硬件资源，您可以不中断正在进行的任务而从物理分区删除 PSB (BB)。
- 移动硬件资源
若需暂时处理任务扩大和负荷增加，您可以暂时从另一个物理分区删除 PSB (BB)。然后将删除的 PSB (BB) 添加到需要增加硬件资源的物理分区。当两个物理分区的任务正在运行时，这将启用需要移动的硬件资源。可对系统进行配置以便灵活地应对系统负荷的变化。
- 部件故障时的活动状态更换
如果 PSB (BB) 上的硬件资源由于 PSB (BB) 的故障而降级，则可以从该物理分区暂时删除该 PSB (BB)，并更换故障部件，而无需停止物理分区上的任何任务。更换部件后，您可以通过将该 PSB (BB) 添加到物理分区来恢复初始配置，而无需停止任何任务。

1.5.3 CPU 和内存的动态重新配置

添加或删除 CPU

Oracle VM Server for SPARC 通过物理分区动态配置可自动识别和使用任何添加的 CPU 可用。

通过物理分区动态重新配置对整合的系统板 (PSB<BB>) 或添加的 CPU 执行以下其中一项操作。

- 如果将 PSB (BB) 添加到将逻辑域配置信息设置为 factory-default（出厂时的配置）的物理分区上，在所添加的 PSB (BB) 上安装的所有 CPU 都自动添加到控制域。
- 如果将一个 PSB (BB) 添加到一个已配置逻辑域或已保存逻辑域配置信息的物理分区，PSB (BB) 上安装的 CPU 将不被分配给任何逻辑域。若需使用这些 CPU，则需要使用 `ldm` 命令将其添加到逻辑域。
- 如果从配置了逻辑域的物理分区释放 PSB (BB)，则在更换 PSB (BB) 时，会自动删除逻辑域的 CPU 分配。在这种情况下，添加 PSB (BB) 将自动将和所删除 CPU 一样多的 CPU 添加到逻辑域。

若需通过动态重新配置删除 CPU，必须满足下列条件。如果条件未满足，物理分区的动态重新配置进程停止并出现消息。

- 没有正在运行的进程被绑定到需要删除的 CPU 上。
如果绑定了任何进程，您需要释放该进程的绑定或停止进程。
- 需要删除的 CPU 不属于任何处理器集。
如果 CPU 属于任意一个处理器集，则需要使用 `psrset` 命令提前从处理器集中将其删除。
- CPU 内核被分配给逻辑域，而无需以 `ldm add-core` 命令或 `ldm set-core` 命令指定内核 ID (CID)。

当在物理分区动态重新配置期间，将逻辑域资源删除和将系统板释放时，如果上述条件无法满足，物理分区动态重新配置处理将停止并出现一条消息。

添加或删除内存

通过物理分区动态重新配置对整合的系统板 (PSB<BB>) 或添加的内存执行以下其中一项操作。

- 如果将 PSB (BB) 添加到将逻辑域配置信息设置为 `factory-default`（出厂时的配置）的物理分区上，在所添加的 PSB (BB) 上安装的所有内存都自动添加到控制域。
- 如果将一个 PSB (BB) 添加到一个已配置逻辑域或已保存逻辑域配置信息的物理分区，PSB (BB) 上安装的内存将不被分配给任何逻辑域。若需使用此内存，则需要使用 `ldm` 命令将其添加到逻辑域。
- 如果从配置了逻辑域的物理分区释放 PSB (BB)，则在更换 PSB (BB) 时，会自动删除逻辑域的内存分配。在这种情况下，添加 PSB (BB) 将自动将和所删除内存相同容量的内存添加到逻辑域。
通过物理分区动态重新配置删除 PSB (BB) 之前，请注意以下事项：

- 如果具有指定物理地址的内存已经通过 `ldm add-memory` 命令或 `ldm set-memory` 命令的 `mblock` 选项被分配给逻辑域，将不通过物理分区动态重新配置删除 PSB (BB)。

1.5.4 软件条件和确认方法

需要启用物理分区动态重新配置的软件条件

表 1-3和表 1-4列出了需要执行物理分区动态重新配置的 XCP、Oracle Solaris 和需要的 SRU/补丁。若需使用物理分区动态重新配置功能，必须设置 XCP 固件和 Oracle VMServer for SPARC 软件版本的适当组合。

表 1-3 SPARC M12-2S 上物理分区动态重新配置所需的 Oracle Solaris 和所需的 SRU/ 补丁列表

OS 版本	域类型	根域	I/O 域	来宾域
	控制域 非虚拟环境			
Oracle Solaris 11	Oracle Solaris 11.3 (*1) SRU11.3.17.5.0 或更高版本	Oracle Solaris 11.3 或更高版本 (*2)	Oracle Solaris 11.3 或更高版本 (*2)	Oracle Solaris 11.3 或更高版本 (*2)
		Oracle Solaris 11.2 或更高版本 (*2)	Oracle Solaris 11.2 或更高版本 (*2)	Oracle Solaris 11.2 或更高版本 (*2)

表 1-3 SPARC M12-2S 上物理分区动态重新配置所需的 Oracle Solaris 和所需的 SRU/ 补丁列表 (续)

OS 版本	域类型			
	控制域 非虚拟环境	根域	I/O 域	来宾域
			Oracle Solaris 11.1 (*2) SRU1.4 或更高版本	Oracle Solaris 11.1 或更高版本 (*2)
Oracle Solaris 10	-	-	-	Oracle Solaris 10 1/13 (*3) 150310-03 或更高版本 Oracle Solaris 10 8/11 (*3)(*4) Oracle Solaris 10 1/13 SPARC 软件包 150310-03 或更新版本 Oracle Solaris 10 9/10 (*3)(*4) Oracle Solaris 10 1/13 SPARC 软件包 150310-03 或更新版本

- *1: system/Idoms 和 system/Idoms/Idomsmanager 软件包是必需的。这些软件包包含在 group/system/solaris-large-server 和 group/system/solaris-small-server 之中。
- *2: system/Idoms 软件包是必需的。此软件包包含在 group/system/solaris-large-server 和 group/system/solaris-small-server 之中。
- *3: 可以分配给来宾域中 CPU 的 LSB 编号没有限制。但是，当 Oracle Solaris 10 在来宾域上运行时，可以在一个来宾域中分配的最大 CPU (vcpu) 数为 1024。
- *4: 在安装 Oracle Solaris 10 9/10 或 Oracle Solaris 10 8/11 之前，需要执行其他步骤。有关详细信息，请参见最新版《产品说明》中的 "在来宾域上安装 Oracle Solaris 10 的注意事项" 中介绍的步骤。

表 1-4 SPARC M10-4S 上物理分区动态重新配置所需的 XCP/Oracle Solaris 和所需的 SRU/补丁列表

服务器	XCP	Oracle Solaris	需要的软件包 需要的产品	需要的 SRU 需要的补丁
SPARC M10-4S	2220 或更高版本	- Oracle Solaris 11.2或更高版本	系统/Idoms (*1) 系统/Idoms/Idomsmanager (*2)	无
		Oracle Solaris 11.1	系统/Idoms (*1) 系统/Idoms/Idomsmanager (*2)	SRU11.1.14.5.0 或更高版本 (*3)
		Oracle Solaris 10 1/13	Oracle VM for SPARC 3.1 (*4)	150817-02 或更高版本 (*4)(*5)

- *1 需要控制域和来宾域。包含在 group/system/solaris-large-server 和 group/system/solaris-small-server 中。
- *2 仅对于控制域是必需的。包含在 group/system/solaris-large-server 和 group/system/solaris-small-server 中。
- *3 需要控制域和来宾域。SRU11.1.14.5.0 包括 Oracle VM Server for SPARC 3.1.0.1。然而，为确保物理分区动态重新配置的稳定运行，需要 Oracle Solaris11.1 补丁 (CR:17709858)。该问题已经在 SRU11.1.15.4.0 和更高版本中解决。
- *4 仅对于控制域是必需的。
- *5 150400-01 和 150400-06 之间请勿使用补丁。

如何检查软件版本：

- 如何检查 XCP 固件的版本

1. 登录 **XSCF shell**。
2. 执行 **version -c xcp** 命令来检查固件版本信息。

```
XSCF> version -c xcp
BB#00-XSCF#0 (Master)
XCP0 (Current): 2220
XCP1 (Reserve): 2220
BB#01-XSCF#0 (Standby)
XCP0 (Current): 2220
XCP1 (Reserve): 2220
```

■ 如何检查 **Oracle VM Server for SPARC** 软件的版本

1. 登录到物理分区的控制域控制台。
有关如何登录控制域控制台的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "8.3 从 XSCF Shell 切换到控制域控制台"。
2. 对于 **Oracle Solaris 11**，执行 **pkg info entire** 命令来检查 **Oracle Solaris** 的 **SRU** 的版本。

```
# pkg info entire
Name: entire
Summary: entire incorporation including Support Repository Update (Oracle Solaris 11.1.14.5.0). <-- SRU 的版本
```

3. 对于 **Oracle Solaris 10**，执行 **ldm -V** 命令来检查 **Oracle VM Server for SPARC** 的版本。

```
# ldm -V
Logical Domains Manager (v 3.1.0.1) <-- Oracle VM Server for SPARC 的版本

Hypervisor control protocol v 1.9
Using Hypervisor MD v 1.3
```

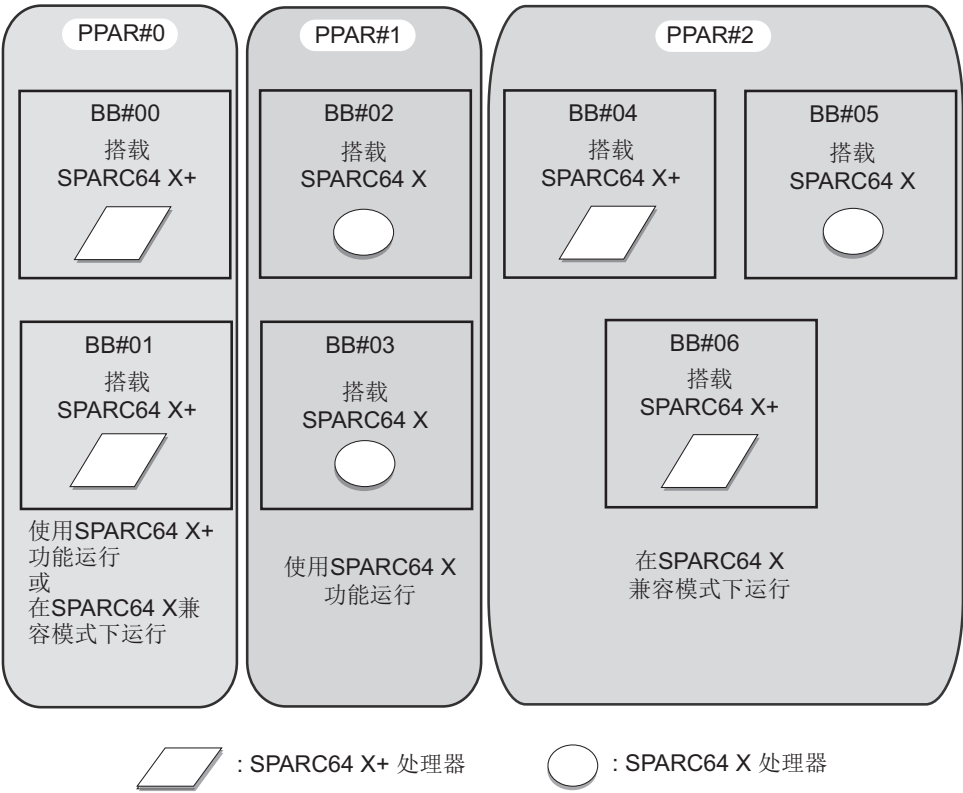
1.6 混合使用 SPARC64 X+ 处理器和 SPARC64 X 处理器

对于 SPARC M10-4S 系统而言，搭载 SPARC64 X+ 处理器的 SPARC M10-4S 和搭载 SPARC64 X 处理器的 SPARC M10-4S 可以采用混合的配置作为一个物理分区一起使用。

当您启动具有由 SPARC64 X+ 处理器和 SPARC64 X 处理器组成的混合配置的物理分区

时，所有的处理器都在 SPARC64 X 处理器兼容模式下运行。
有关 SPARC64 X+ 处理器和 SPARC64 X 处理器组合使用的详情，请参见 "2.6.3 PPAR 内 SPARC64 X+ 处理器和 SPARC64 X 处理器的混合配置条件"。

图 1-19 SPARC M10 系统中由处理器和物理分区所构成的配置示例



1.6.1 CPU 操作类型和 CPU 运行模式

在采用通过使用 XSCF 固件的 `setpparmode` 命令设置的 CPU 运行模式的 SPARC M10-4S 中，您可以指定每个物理分区的 CPU 操作类型。当 Oracle Solaris 下次启动时，此操作可自动确定 CPU 运行是使用 SPARC64 X+ 处理器功能还是使用 SPARC64 X 处理器功能。

CPU 的运行包括以下几种类型。

- 采用 SPARC64 X+ 功能运行
物理分区中所有的 CPU 都使用扩展的 SPARC64 X+ 处理器的功能运行。
- 采用 SPARC64 X 功能运行
物理分区中所有的 CPU 都使用 SPARC64 X 处理器的功能进行操作。无论搭载在配置物理分区的系统板上的 CPU 类型是什么，当采用 SPARC64 X 处理器功能让所有 CPU 运行时，使用此功能。SPARC64 X 处理器始终使用 SPARC64 X 处理器功能运行。

注 -有关 CPU 操作类型和 XCP 固件支持版本的详情，请参见最新 XCP 版本的最新《SPARC M10 系统 产品说明》。

注 -有关 CPU 运行模式的设置详情，请参见 "[2.6.2 CPU 运行模式设置](#)"。

注 - 有关 CPU 运行模式和物理分区动态重新配置的详情，请参见 "[2.6.4 CPU 运行模式和物理分区动态重新配置之间的关系](#)"。

系统运行的条件和设置

本章介绍在开始配置域之前您应该了解的内容。

- [XSCF 状态管理](#)
- [XSCF 的条件和设置](#)
- [Oracle Solaris 的条件和设置](#)
- [逻辑域的条件和设置](#)
- [动态重新配置的条件和设置](#)
- [使用 SPARC64 X+ 处理器时的注意事项](#)

2.1 XSCF 状态管理

若需正确执行物理分区重新配置，您需要根据显示为一个模块化架构模块的物理分区和系统板 (PSB<BB>) 的状态来执行操作。本节介绍由 XSCF 管理的物理分区和 PSB (BB) 的状态信息，并提供有关各种状态的注意事项。这样，您可以了解物理分区重新配置功能运行的条件。

2.1.1 物理分区状态

XSCF 管理物理分区状态。您可以使用 XSCF 提供的用户界面来显示和查阅物理分区状态。下面的示例使用 `showpparstatus` 命令从基于 XSCF shell 的命令行来检查物理分区状态。

```
XSCF> showpparstatus -p 0
PPAR-ID PPAR Status
00      Initialization Complete
```

有关用户界面的详情，请参见 ["第3章 域配置的操作"](#)。

XSCF 管理以下物理分区相关的状态。

表 2-1 物理分区状态

状态	含义
Powered Off	电源关闭状态
Initialization Phase	电源开启自检 (POST) 正在运行状态
Initialization Complete	POST 完成状态
Running	POST 处理完成后的运行状态
Hypervisor Aborted	虚拟机监控程序中止直到重置时的状态
-	上述以外的状态（未定义物理分区的状态）

如需执行物理分区动态重新配置，您需要确定如何根据相关物理分区状态操作系统板 (PSB<BB>)。

2.1.2 系统板状态

XSCF 管理系统板 (PSB) 单元中一个模块化架构的状态。您可以使用 XSCF 提供的用户界面来显示和查阅 PSB 状态。下面的示例使用 `showboards` 命令从基于 XSCF shell 的命令行来确认 PSB 状态。

XSCF> showboards -av								
PSB	R	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	*	00 (00)	Assigned	n	n	n	Unmount	Normal
01-0		00 (01)	Assigned	y	y	n	Passed	Normal
02-0		00 (02)	Assigned	y	y	n	Passed	Normal
04-0		00 (04)	Assigned	y	y	n	Passed	Normal
05-0		01 (00)	Assigned	y	y	n	Passed	Normal
06-0		01 (01)	Assigned	y	y	n	Passed	Normal
09-0		00 (09)	Assigned	y	y	n	Passed	Normal
13-0		13 (00)	Assigned	y	y	n	Passed	Normal
15-0		13 (01)	Assigned	y	y	n	Passed	Normal

有关用户界面的详情，请参见 "第3章 域配置的操作"。

XSCF 管理以下系统板相关的状态。

表 2-2 系统板状态

显示项目	说明	含义
PSB	xx-y (xx: 00 至 15 之间的整数; y: 固定为 0)	表示一个模块化架构 (BB) 的物理系统板 (PSB) 编号（其中 xx 是 BB-ID）
R (*1)	*	物理分区重新启动时，将 PSB (BB) 整合到或释放当前物理分区配置以及配置变更时的状态
PPAR-ID	0 到 15 之间的整数	分配 PSB (BB) 的物理分区的编号

表 2-2 系统板状态 (续)

显示项目	说明	含义
	SP	系统板池状态下的 PSB (BB)
	Other	物理分区配置信息被设置在具有用户权限的物理分区中并且属于不具有用户权限的物理分区的状态
LSB	00 到 15 之间的整数	物理分区所使用的逻辑系统板编号
Assignment (正在分配给 PPAR 的状态)	Unavailable	未分配给物理分区的 PSB (BB) 和处于以下任意一种状态 (包括无法识别 PSB 的情况): 系统板池、未诊断、正在诊断中、异常诊断中的 PSB (BB)
	Available	系统板池状态下的 PSB (BB), 诊断可正常结束
	Assigned	PSB (BB) 已分配
Pwr (系统板的电源开启状态)	n	电源关闭状态
	y	电源开启状态
Conn (整合到 PPAR 配置中的状态)	n	PSB (BB) 从物理分区配置中被释放的状态
	y	PSB (BB) 被整合到物理分区配置中的状态
Conf (Oracle Solaris 中的运行状态)	n	Oracle Solaris 不在 PSB (BB) 上运行的状态
	y	Oracle Solaris 在 PSB (BB) 上运行的状态
Test (诊断状态)	Unmount	已卸载或未定义的状态
	Unknown	未诊断的状态
	Testing	正在诊断中
	Passed	诊断正常结束的状态
	Failed	诊断检测出异常, 并且 PSB (BB) 不工作的状态
Fault (已降级的状态)	Normal	正常状态
	Degraded	存在已降级部件的状态 (PSB (BB) 可以运行)
	Faulted	由于异常, PSB (BB) 不工作或由于通讯故障无法被管理的状态

*1 仅在指定 -v 选项时才显示此项目。

XSCF 根据情况, 变更/设置 PSB (BB) 状态。此种状态包括: 安装/移除 PSB (BB)、注册物理分区配置信息以及启动/停止物理分区。在物理分区动态重新配置过程中, 当添加或删除 PSB (BB) 时, 状态同样发生变更。

如需执行物理分区动态重新配置, 您需要确定如何根据相关 PSB (BB) 状态操作 PSB (BB)。

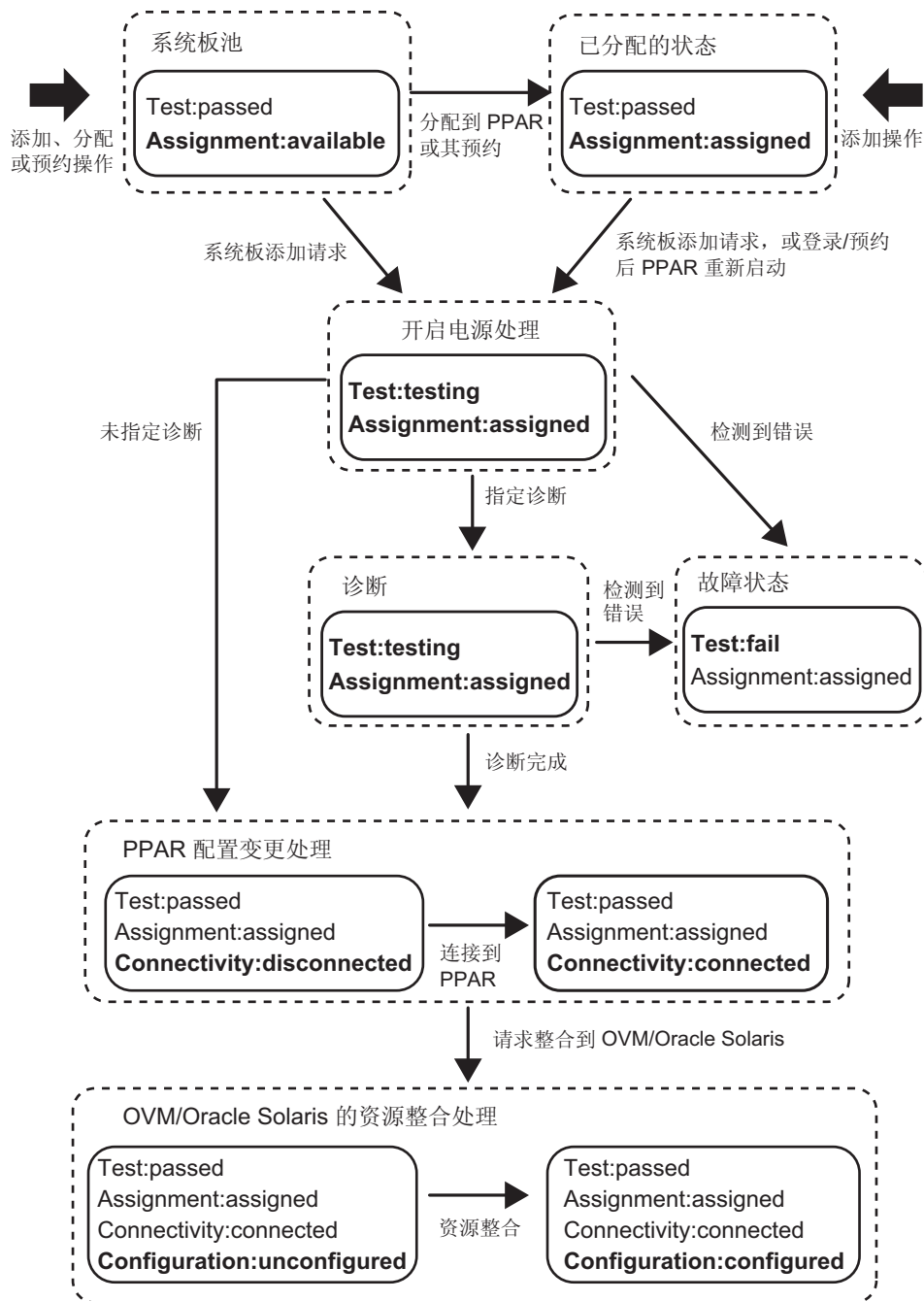
2.1.3 状态变化

本节介绍随着每个物理分区配置的变更, 处理流程和系统板 (PSB<BB>) 状态的变化。

系统板添加处理

本节使用示意图来描述在 PSB (BB) 添加或 PSB (BB) 添加预约操作时的处理流程和系统板 (PSB<BB>) 状态的变化。图 2-1 显示了主要的 PSB (BB) 状态变化。

图 2-1 系统板添加处理

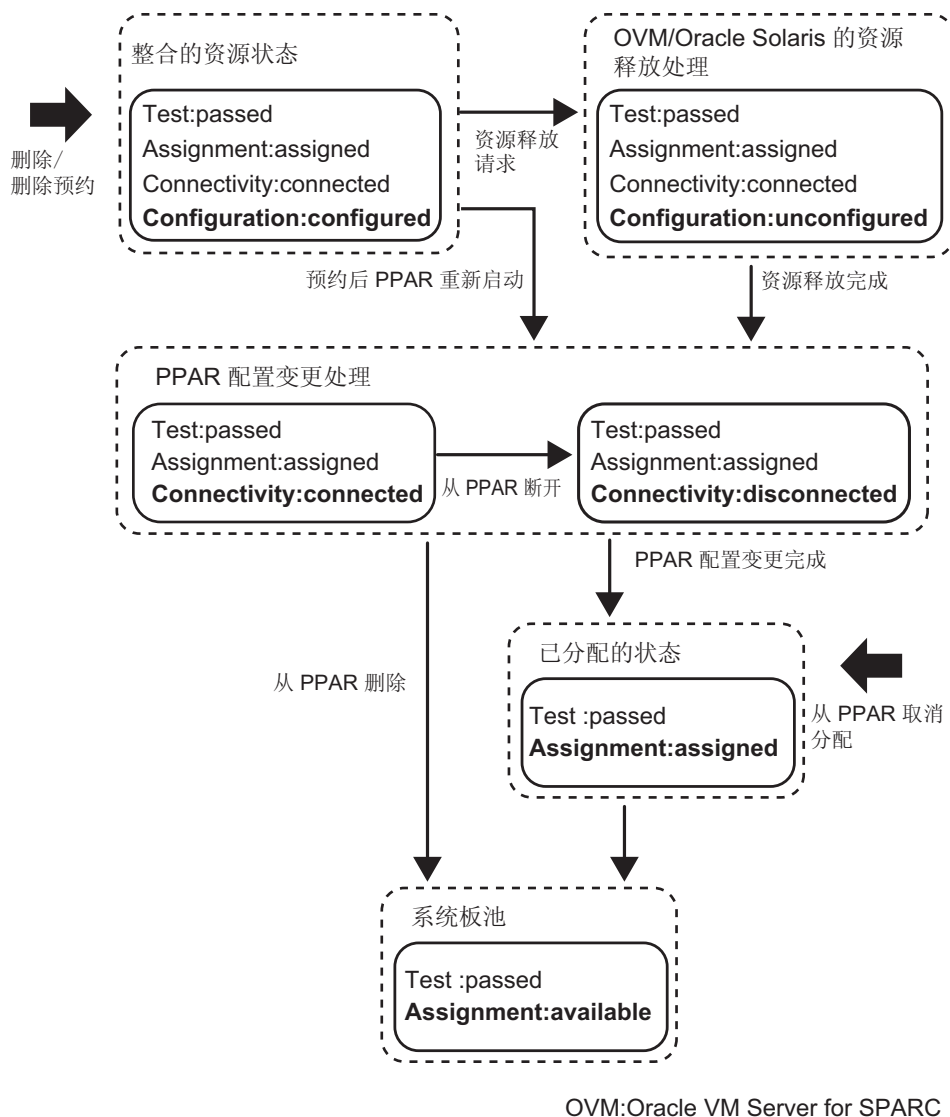


OVM:Oracle VM Server for SPARC

系统板删除处理

本节使用示意图来描述在 PSB (BB) 删除或 PSB (BB) 删除预约操作时的处理流程和系统板 (PSB<BB>) 状态的变化。图 2-2 显示了主要的 PSB (BB) 状态变化。

图 2-2 系统板删除处理

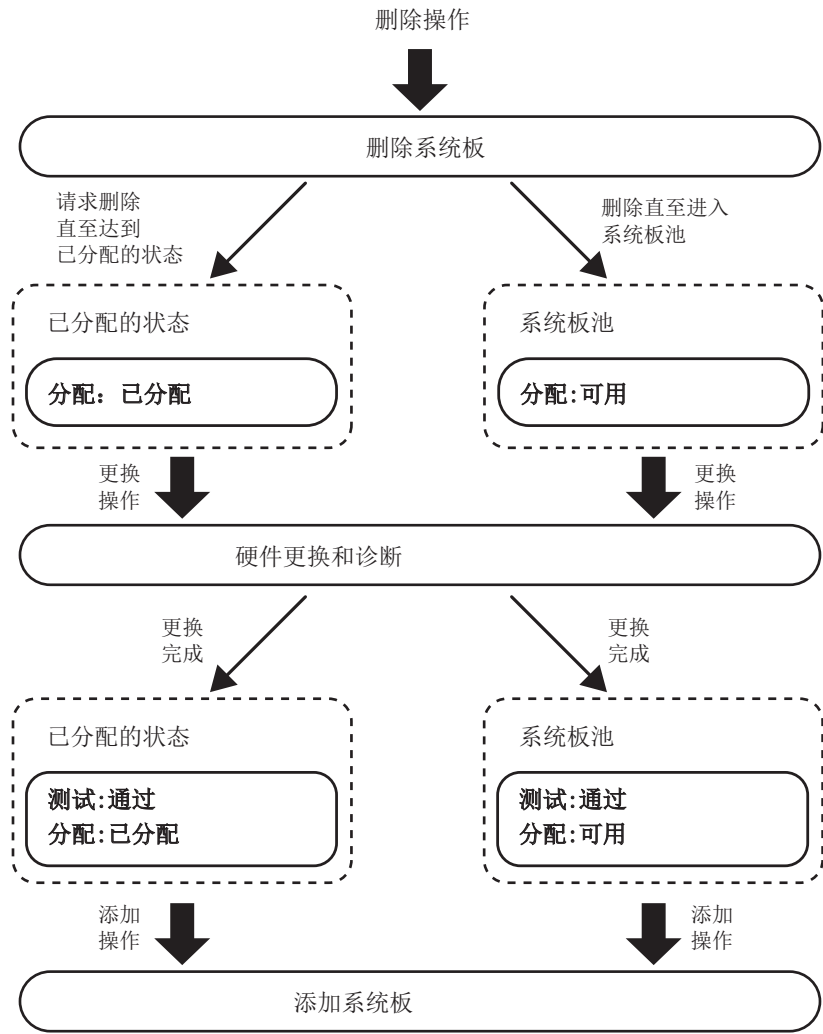


系统板更换处理

本节使用示意图来描述在 PSB (BB) 更换操作时的处理流程和系统板 (PSB<BB>) 状态的变化。图 2-3 显示了主要的 PSB (BB) 状态变化。硬件更换之前和之后的状态的示例显示如下。当您更换硬件时，其状态可能不会与此处显示对应的状态相匹配。有关 PSB

(BB) 添加和删除处理流程以及处理过程中的状态的详情，请参见 "系统板添加处理" 和 "系统板删除处理"。有关硬件更换操作的详情，请参见服务器的《Service Manual》。

图 2-3 系统板更换处理



2.2 XSCF 的条件和设置

本节介绍 XSCF 需要的条件以及使用 XSCF 所进行的设置。

2.2.1 XSCF 需要的配置条件

当添加系统板 (PSB<BB>) 时，您不能简单地通过安装和连接 SPARC M12-2S/M10-4S 将系统板插入物理分区。请使用 XSCF shell 或 XSCF Web，以便在物理分区配置信息（PPAR 配置信息）中登录 PSB (BB)。这使您能够在物理分区上执行操作。当操作物理分区时，在整合 PSB (BB) 之前您需要确认添加的 PSB (BB) 是否在 PPAR 配置信息中登录。

当删除或更换 PSB (BB) 时，PSB (BB) 应该已经在 PPAR 配置信息中登录。您不必确认系统板是否在 PPAR 配置信息中登录。

2.2.2 XSCF 的设置

为了更加顺利地重新配置物理分区和持续运行，与物理分区配置相关的功能包括一些选配的功能。该功能可防止有关 Oracle Solaris 重新配置、内存分配等的复杂性。使用 XSCF shell 或 XSCF Web 来配置这些选配功能。本节介绍以下选配功能：

- 配置策略选项
- 内存无效选项
- I/O 无效选项

使用 setpcl 命令来设置这些选项。

设置配置策略选项

当将物理分区开启电源或采用物理分区动态重新配置添加系统板 (PSB<BB>) 时，硬件需要进行开启电源自检，以确保 PSB (BB) 的安全插入。配置策略是一个选项，指定如果开启电源自检检测到错误时所应用的降级范围。

可设置的降级单元如下：检测到硬件错误的部件、该部件所安装的系统板 (PSB) 单元（BB 单元）、或物理分区。表 2-3 列出了此选项可设置的设置值和降级的单元。此选项的默认值是 "fru"。

表 2-3 降级的单元

设置值	降级的单元
fru	以部件（如 CPU 或内存）为单元执行降级。
psb	系统板 (PSB) 中以单元（BB 单元）进行降级。
system	降级的单元是物理分区。将物理分区停止，无降级。

注 - 设置配置策略选项之前，停止相关物理分区。

内存无效选项

此选项可防止需要整合到物理分区的系统板 (PSB<BB>) 上的内存存在逻辑域中被使用。

您可以组合使用此选项和 I/O 无效选项，以便仅将添加到目标物理分区的系统板上的 CPU 整合到逻辑域而不将内存整合到逻辑域。此选项可选择 "true" 或 "false"。前者使内存无效，而后者将其禁用。默认值是 "false"。

注 -即使启用内存无效选项后，您需要实际加载内存来诊断和管理 PSB (BB)。启用内存无效选项时，物理分区可用内存将减少并且性能可能会下降。设置此选项之前，您需要考虑对任务的影响。

注 -当相关 PSB (BB) 处于系统板池的状态，或当物理分区电源关闭时，设置此内存无效选项。

I/O 无效选项

此选项可防止 PCI 卡、磁盘驱动器和系统板 (PSB<BB>) 上的基本局域网 (LAN) 端口以及 PCI 扩展单元被插入到目标物理分区。如果在物理分区上仅需要使用 PSB (BB) 上的 CPU 和内存，请指定 "true" 来启用 I/O 无效选项。如果在物理分区上需要使用 PSB (BB) 上的 PCI 卡和 I/O 设备，请将此选项设置为 "false"。在这种情况下，您需要充分了解使用这些 I/O 设备的限制。在采用动态重新配置删除 PSB (BB) 之前，您还需要停止所有软件（如应用程序或守护程序）和使用 I/O 设备的 I/O 域。

如果采用配置为 **factory-default** 设置的逻辑域配置信息启动物理分区，物理分区将根据 I/O 无效选项的设置来启动。假设采用 **factory-default** 设置以外的逻辑域配置信息启动物理分区，并且根合成体和相关系统板的 I/O 设备都属于任意一个逻辑域。系统将判断该物理分区配置已经改变，并且通过切换到 **factory-default** 设置来启动物理分区。如果上述不属于任何逻辑域，将不执行切换到出厂默认设置。如果通过动态重新配置的方法添加 PSB (BB)，物理分区的插入处理将根据 I/O 无效选项进行的设置执行。

I/O 无效选项的默认设置是 "false"。

注 -当相关 PSB (BB) 处于系统板池的状态，或当物理分区电源关闭时，设置此 I/O 无效选项。

2.2.3 内存镜像模式的注意事项

内存镜像模式是复制内存以确保内存硬件可靠性的一种功能。即使某部分内存发生故障，只要可能从故障中恢复，您就可以启用内存镜像模式来继续操作物理分区。当使用物理分区动态重新配置的情况下，启用内存镜像模式不会限制物理分区动态重新配置的功能。

但是，需要注意物理分区的配置和操作。例如，如果在动态重新配置中启用了内存镜像模式的系统板 (PSB<BB>) 上执行操作，则在删除 PSB (BB) 时，Oracle VM Server for SPARC 和 Oracle Solaris 使用的内存资源将被移到其他的 PSB (BB)。此时，如果目标 PSB (BB) 禁用内存镜像模式，Oracle VM Server for SPARC 和 Oracle Solaris 使用的内存资源将被正确地移动，但继续运行将会带来降低内存可靠性的后果。请考虑域配置、操作等方面的需求并且提前制定计划，以便设置内存镜像模式。

有关内存镜像设置的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "第14章 配置高可靠性系统"。

2.2.4 添加 CPU 模块时的注意事项

如果在启用 I/O 总线重新配置时添加 CPU 模块 (CMUU)，则使用中的 I/O 设备的设备路径会发生变化。

通常，应禁用 (false) 可使用 `setpparmode` 命令的 `-m` 功能选项来设置 I/O 总线重新配置 (`ioreconfigure`) 功能。

启用 I/O 总线重新配置功能 (true) 可能需要重新安装 Oracle Solaris。

有关 `setpparmode` (8) 命令的详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

2.3 Oracle Solaris 的条件和设置

本节介绍物理分区重新配置后，启动操作需要的条件和设置。

2.3.1 I/O 和软件要求

冗余配置中的 I/O 设备

如果使用多个机柜来配置物理分区或者如果由于 I/O 设备故障阻碍了任务的处理，建议使用软件来配置 I/O 设备的冗余配置和切换 I/O 设备。

2.3.2 交换空间相关的注意事项

可用的虚拟内存容量是系统上安装的内存和磁盘上交换空间的容量的总和。您需要确保系统有足够的可用内存容量来执行所有必要的操作。

添加 BB (PSB) 时的注意事项

默认情况下，Oracle Solaris 也使用交换空间，用于存储系统崩溃转储的目的。您应该使用转储专用的设备来代替。有关详情，请参见 Oracle Solaris 的手册页的 `dumpadm`。用于存储崩溃转储的交换空间的默认容量将根据所安装内存的容量变化。

存储崩溃转储的转储设备的容量必须大于所安装内存的容量。如果添加了系统板，并且安装的内存发生增加，需要重新配置转储设备。有关详情，请参见 Oracle Solaris 的手册页的 `dumpadm`。

删除 BB (PSB) 时的注意事项

如果删除 BB (PSB) 也删除了域中的内存，BB (PSB) 上的内存将被磁盘上的交换空间所交换。然后，内存将减少与删除的内存容量相同的容量。执行命令来删除 BB (PSB) 之前，检查整个交换空间，以便确认是否有足够的交换空间容量用于存储 BB (PSB) 的物理内存数据。另外请注意，在有些情况下将连接到需要删除的 BB (PSB) 上的磁盘作为交换空间的一部分。换句话说，如果删除 BB (PSB) 减少了交换空间，您还需要计算失

去的交换空间。

- 假设（可用交换空间：1.5 GB）>（删除的内存容量：1 GB）。删除后，可用的内存容量成为 0.5 GB。
- 假设（可用交换空间：1.5 GB）<（删除的容量：2 GB）。BB (PSB) 删除处理失败。

若需查找当前可用交换空间的容量，请在 Oracle Solaris 上执行 `swap -s` 命令，然后查看显示为 "available" 的容量。有关详情，请参见 Oracle Solaris 的手册页的 `swap`。

2.3.3 实时处理的注意事项

当物理分区动态重新配置删除系统板时，Oracle Solaris 将在分配正在被删除的系统板上的内存的逻辑域上被挂起。请注意，如果系统具有实时性要求（显示所有实时处理），该处理可能对其处理过程具有相当大的影响。

2.4 逻辑域的条件和设置

注 - Logical Domains Manager（逻辑域管理器）是 Oracle VM Server for SPARC 的管理软件，用于配置逻辑域。配置逻辑域时，请参见所用版本的 Oracle VM Server for SPARC 手册，并找出所需的软件。

注 - 本节仅提供配置逻辑域所必需的信息。有关 Oracle VM Server for SPARC 中可用功能的详细信息，请参见所用版本的 Oracle VM Server for SPARC 手册，并找出所需的软件。

配置逻辑域时设置以下内容。

控制域的初始配置

在初始状态下，所有硬件资源都被分配给控制域（出厂默认配置）。因此，您应检查分配给控制域的硬件资源以便确保来宾域的硬件资源。因此，请执行以下任务：

- 设置默认服务
- 设置延迟重新配置模式
- 释放虚拟 CPU 资源
- 释放虚拟内存资源
- 释放 I/O 资源或虚拟 I/O 资源
- 保存控制域设置

配置来宾域

查看分配给控制域的硬件资源后，将硬件资源分配给各来宾域。因此，请执行以下任务：

- 分配虚拟 CPU
- 分配虚拟内存
- 设置虚拟网络

- 设置虚拟磁盘
- 设置启动设备
- 设置自动重启
- 保存来宾域设置

2.4.1 逻辑域配置注意事项

本节介绍配置逻辑域时应该考虑的内容。

注意逻辑域的数量

可以配置的逻辑域最大数量与可以被分配给逻辑域的虚拟 CPU 数（线程）的数量相同。此外，每个物理分区的最大逻辑域的数量由软件限制。表 2-4 按照型号列出了可配置的逻辑域的最大数量。

表 2-4 按照型号逻辑域可以配置的最大数量

型号	逻辑域可以配置的最大数量
SPARC M12-1	48
SPARC M12-2	192
SPARC M12-2S	每个物理分区的线程数或 256，取其较小的数
SPARC M10-1	32
SPARC M10-1 系统(3.7 GHz)	16
SPARC M10-4	128
SPARC M10-4 机柜(3.7 GHz)	64
SPARC M10-4S	每个物理分区的线程数或 256，取其较小的数

通常，可以将虚拟 CPU 以内核为单元分配给逻辑域。

创建超过物理分区内核数量的逻辑域之前，请执行操作测试等验证。

使用 CPU 自动更换功能时的注意事项

使用 Oracle VM Server for SPARC 3.0 或更高版本支持的 CPU 自动更换功能时，您需要考虑 CPU 的配置。有关详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "10.7 设置故障的 CPU 自动更换"。

Logical Domains DR 守护程序的操作

如需将硬件资源分配给逻辑域，Logical Domains DR 守护程序 (drd) 必须在管理该逻辑域的控制域上运行。有关 drd 守护程序的详情，请参见《Oracle VM Server for SPARC Reference Manual》。

注册 CPU 激活密钥

在将 CPU 内核分配给逻辑域之前，必须通过将在 XSCF 中注册的 CPU 激活密钥分配

给物理分区来实现。

CPU 激活密钥在服务器部署时随附的 CD-ROM 上标记为 "<model name> CPU Activation" 上可以查看到。登录 CPU 激活密钥之前，将 CD-ROM 放在方便随时查看的位置。

CPU 激活密钥包含在 CD-ROM 上的 "ACTIVATION_KEY" 文件夹中的文本文件中。提供了按批处理的登录密钥的文件 (XXXXXX_XX.TXT)，和另一个一次一个地登录密钥的文件 (XXXXXX_XX_001.TXT等)。根据需要使用其中一个文件。

注 - 系统导入时的 XSCF 可能已经包含了 CPU 激活密钥。登录密钥可能与系统随附 CD-ROM 上包含的密钥相同。这种情况下，您不必将密钥登录到 XSCF。

注 - 如果由 CPU 激活密钥引起的可用 CPU 内核数量减少，需重新分配 CPU 内核，以便分配给物理分区的 CPU 内核的数量等于或少于可用 CPU 内核的数量。

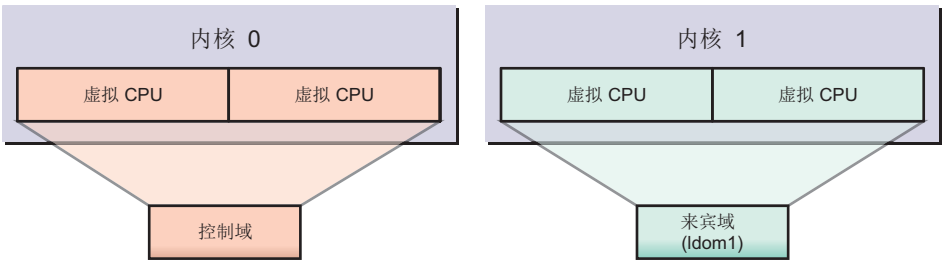
控制域初始设置的注意事项

进行控制域初始设置时，请勿使用 ldm 命令的内存动态重新配置 (DR) 功能。请使用延迟重新配置模式进行控制域的初始设置。此模式通过逻辑域重新启动反应配置信息。

虚拟 CPU 分配的单元

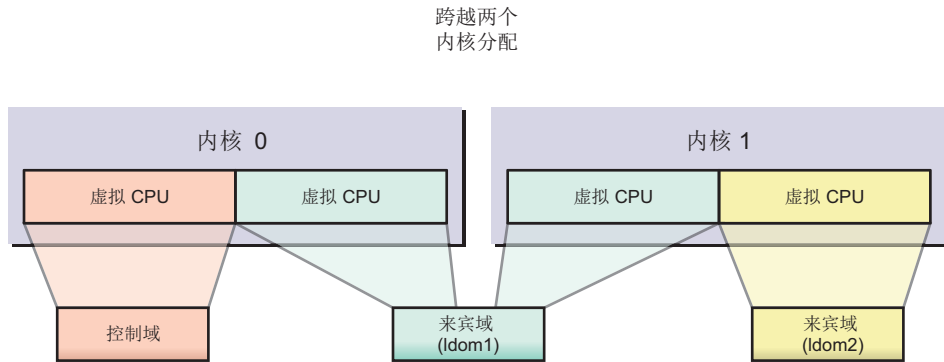
尽管可以以内核或线程为单元分配虚拟 CPU，我们建议您通常以内核为单元分配它们。[图 2-4](#) 是正常分配的示例。

图 2-4 正常分配的示例



应该注意到，在虚拟 CPU 未以内核为单元分配的情况下，性能可能会下降。[图 2-5](#) 是性能下降的示例。

图 2-5 不合理分配的示例



CPU 内核 ID 指定

如需分配或删除使用 `ldm` 命令指定逻辑域的 CPU 内核 ID (CID) 的已命名的内核，您需要在理解其与 CPU 内核物理位置关系的同时设置该 CID。

注 - 如果您指定 CID 以便将 CPU 内核分配给逻辑域时，您无法使用下列功能。充分理解内容后再进行设置。

- CPU 的动态重新配置
- CPU 的自动更换功能
- CPU 动态资源管理 (DRM)

在 Oracle VM Server for SPARC 中，内核 ID (CID) 由物理 CPU 编号 (PID) 和线程数决定。

下图显示各种型号的 SPARC M12/M10 中 CPU 的物理位置。

图 2-6 SPARC M12-1 的 CPU 位置

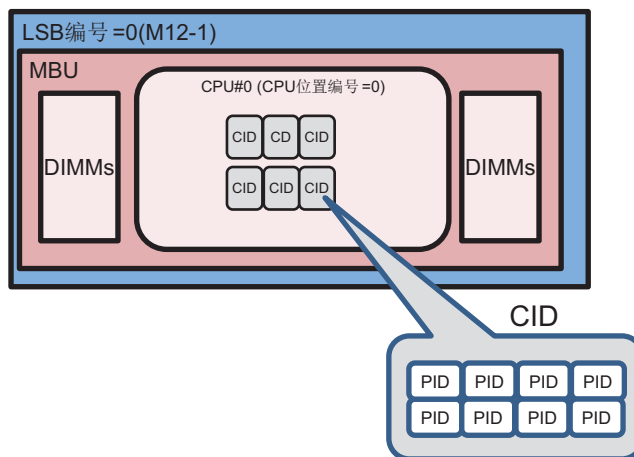


图 2-7 SPARC M12-2 或 SPARC M12-2S 的 CPU 位置（安装 1 个 CPU）

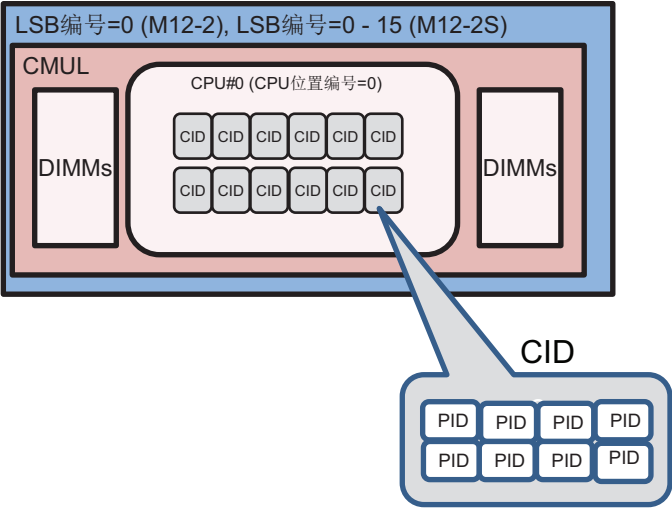


图 2-8 SPARC M12-2 或 SPARC M12-2S 的 CPU 位置（安装 2 个 CPU）

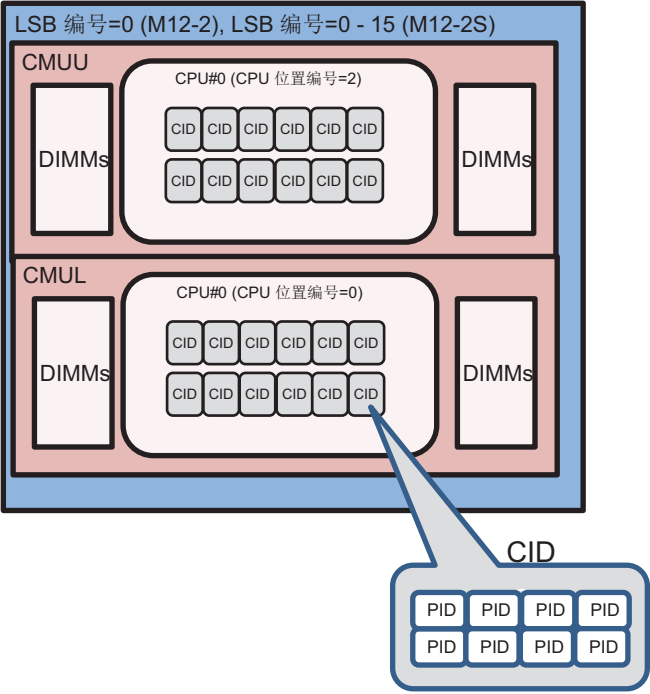


图 2-9 SPARC M10-1 的 CPU 位置

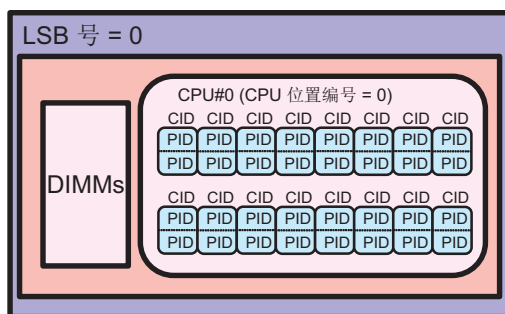


图 2-10 SPARC M10-4 或 SPARC M10-4S 的 CPU 位置（安装 2 个 CPU）

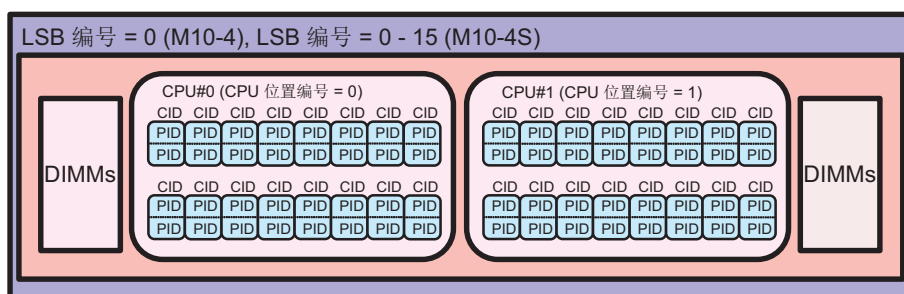
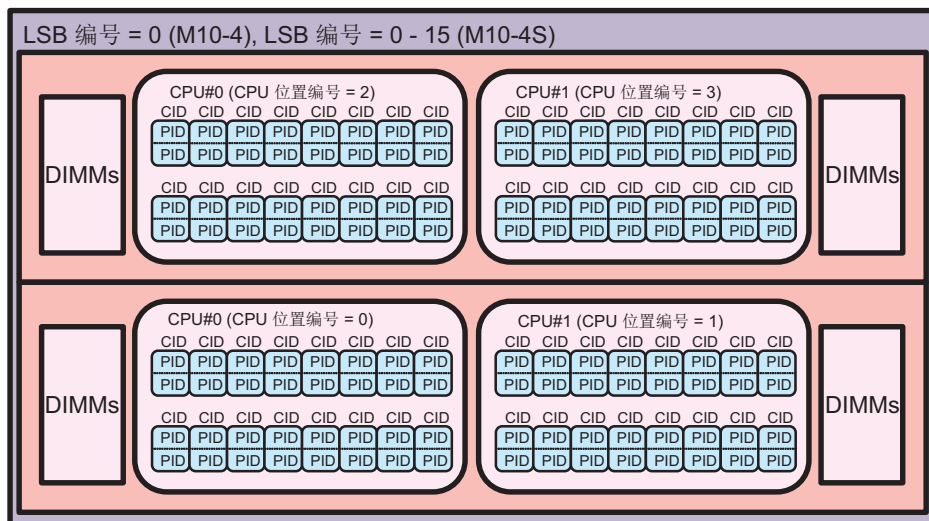


图 2-11 SPARC M10-4 或 SPARC M10-4S 的 CPU 位置（安装 4 个 CPU）



对于 SPARC M12/M10 系统，CPU 的物理位置和 CID 之间的关系如下所示。

在本节中，作为表示 CPU 位置的参数，逻辑系统板 (LSB) 上的 CPU 序列号被称为 CPU 位置编号。

$$\begin{aligned} \text{pid} &= \text{LSB 数} \times 1024 + \text{CPU 位置数} \times 256 + \text{CPU 内核数} \times 8 + \text{线程数} \\ \text{CID} &= \text{PID} / \text{线程数} \end{aligned}$$

上式中使用的各编号的范围如下所示。

表 2-5 表示 CPU 物理位置的各编号的范围 (SPARC M12)

编号类型	范围
LSB 号	0 至 15 (*1)
CPU 位置编号	0、2 (*2)
CPU 内核编号	0、1、2、4、5、6、8、9、10、12、13、14 (0、1、2、4、5、6) (*3)
线程编号	0 至 7
线程的数量	8

*1 对于 SPARC M12-1/M12-2，LSB 编号固定为 0。对于 SPARC M12-2S，使用 XSCF 固件的 setpcl 命令可设置 LSB 号。
*2 当 SPARC M12-1/M12-2 或 SPARC M12-2S 中安装一个 CPU 时，CPU 位置编号为 0。当安装了两个 CPU 时，CMUL 的 CPU#0 为 0，CMUU 的 CPU#0 为 2。
*3 括号中的数字是 SPARC M12-1 的 CPU 内核数。

表 2-6 表示 CPU 物理位置的各编号的范围 (SPARC M10)

编号类型	范围
LSB 号	0 至 15 (*1)
CPU 位置编号	0 至 3 (*2)
CPU 内核编号	0 至 15 (0 至 7) (*3)
线程编号	0 至 1
线程的数量	2

*1 对于 SPARC M10-1 和 M10-4，LSB 号固定为 0。对于 SPARC M10-4S，使用 XSCF 固件的 setpcl 命令可设置 LSB 号。
*2 当 SPARC M10-4 或 M10-4S 中安装两个 CPU 时，CPU 位置编号为 0 至 1。当安装四个 CPU 时，CPU 位置编号如下：CMUL CPU#0 和 #1 分别为 0 和 1，CMUU CPU#0 和 #1 分别为 2 和 3。
*3 当 SPARC64 X+8 核 3.7 GHz 处理器搭载在 SPARC M10-1/M10-4 上时，括号中的数字是 CPU 内核数。

因此，CPU 内核 ID 和各种型号的 SPARC M12 系统的物理位置之间的关系如 [表 2-7](#) 中所示。

CPU 内核 ID 和各种型号的 SPARC M10 系统的物理位置之间的关系如 [表 2-8](#) 中所示。

表 2-7 各型号的 CPU 内核 ID 列表 (SPARC M12)

型号	LSB 号	CPU 位置 编号	PID	CID
SPARC M12-1	0	0	0 至 55	0 至 6
SPARC M12-2	0	0	0 至 119	0 至 14
		2	512 至 631	64 至 78
SPARC M12-2S	0	0	0 至 119	0 至 14

表 2-7 各型号的 CPU 内核 ID 列表 (SPARC M12) (续)

型号	LSB 号	CPU 位置 编号	PID	CID
		2	512 至 631	64 至 78
	1	0	1024 至 1143	128 至 142
		2	1536 至 1655	192 至 206
	2	0	2048 至 2167	256 至 270
		2	2560 至 2679	320 至 334
	3	0	3072 至 3191	384 至 398
		2	3584 至 3703	448 至 462
	4	0	4096 至 4215	512 至 526
		2	4608 至 4727	576 至 590
	5	0	5120 至 5239	640 至 654
		2	5632 至 5751	704 至 718
	6	0	6144 至 6263	768 至 782
		2	6656 至 6775	832 至 846
	7	0	7168 至 7287	896 至 910
		2	7680 至 7799	960 至 974
	8	0	8192 至 8311	1024 至 1038
		2	8704 至 8823	1088 至 1102
	9	0	9216 至 9335	1152 至 1166
		2	9728 至 9847	1216 至 1230
	10	0	10240 至 10359	1280 至 1294
		2	10752 至 10871	1344 至 1358
	11	0	11264 至 11383	1408 至 1422
		2	11776 至 11895	1472 至 1486
	12	0	12288 至 12407	1536 至 1550
		2	12800 至 12919	1600 至 1614
	13	0	13312 至 13431	1664 至 1678
		2	13824 至 13943	1728 至 1742
	14	0	14336 至 14455	1792 至 1806
		2	14848 至 14967	1856 至 1870
	15	0	15360 至 15479	1920 至 1934
		2	15872 至 15991	1984 至 1998

表 2-8 各型号的 CPU 内核 ID 列表 (SPARC M10)

型号	LSB 号	CPU 位置编号	PID	CID
SPARC M10-1	0	0	0 至 121 (0 至 57) (*1)	0 至 60 (0 至 28) (*1)
SPARC M10-4	0	0	0 至 121 (0 至 57) (*1)	0 至 60 (0 至 28) (*1)
		1	256 至 377 (256 至 313) (*1)	128 至 188 (128 至 156) (*1)
		2	512 至 633 (512 至 569) (*1)	256 至 316 (256 至 284) (*1)
		3	768 至 889 (768 至 825) (*1)	384 至 444 (384 至 412) (*1)
SPARC M10-4S	0	0	0 至 121	0 至 60
		1	256 至 377	128 至 188
		2	512 至 633	256 至 316
		3	768 至 889	384 至 444
SPARC M10-4S	1	0	1024 至 1145	512 至 572
		1	1280 至 1401	640 至 700
		2	1536 至 1657	768 至 828
		3	1792 至 1913	896 至 956
SPARC M10-4S	2	0	2048 至 2169	1024 至 1084
		1	2304 至 2425	1152 至 1212
		2	2560 至 2681	1280 至 1340
		3	2816 至 2937	1408 至 1468
SPARC M10-4S	3	0	3072 至 3193	1536 至 1596
		1	3328 至 3449	1664 至 1724
		2	3584 至 3705	1792 至 1852
		3	3840 至 3961	1920 至 1980
SPARC M10-4S	4	0	4096 至 4217	2048 至 2108
		1	4352 至 4473	2176 至 2236
		2	4608 至 4729	2304 至 2364
		3	4864 至 4985	2432 至 2492
SPARC M10-4S	5	0	5120 至 5241	2560 至 2620
		1	5376 至 5497	2688 至 2748
		2	5632 至 5753	2816 至 2876
		3	5888 至 6009	2944 至 3004
SPARC M10-4S	6	0	6144 至 6265	3072 至 3132
		1	6400 至 6521	3200 至 3260
		2	6656 至 6777	3328 至 3388
		3	6912 至 7033	3456 至 3516
SPARC M10-4S	7	0	7168 至 7289	3584 至 3644
		1	7424 至 7545	3712 至 3772
		2	7680 至 7801	3840 至 3900
		3	7936 至 8057	3968 至 4028
SPARC M10-4S	8	0	8192 至 8313	4096 至 4156
		1	8448 至 8569	4224 至 4284
		2	8704 至 8825	4352 至 4412
		3	8960 至 9081	4480 至 4540
SPARC M10-4S	9	0	9216 至 9337	4608 至 4668
		1	9472 至 9593	4736 至 4796
		2	9728 至 9849	4864 至 4924
		3	9984 至 10105	4992 至 5052

表 2-8 各型号的 CPU 内核 ID 列表 (SPARC M10) (续)

型号	LSB 号	CPU 位置编号	PID	CID
SPARC M10-4S	10	0	10240 至 10361	5120 至 5180
		1	10496 至 10617	5248 至 5308
		2	10752 至 10873	5376 至 5436
		3	11008 至 11129	5504 至 5564
SPARC M10-4S	11	0	11264 至 11385	5632 至 5692
		1	11520 至 11641	5760 至 5820
		2	11776 至 11897	5888 至 5948
		3	12032 至 12153	6016 至 6076
SPARC M10-4S	12	0	12288 至 12409	6144 至 6204
		1	12544 至 12665	6272 至 6332
		2	12800 至 12921	6400 至 6460
		3	13056 至 13177	6528 至 6588
SPARC M10-4S	13	0	13312 至 13433	6656 至 6716
		1	13568 至 13689	6784 至 6844
		2	13824 至 13945	6912 至 6972
		3	14080 至 14201	7040 至 7100
SPARC M10-4S	14	0	14336 至 14457	7168 至 7228
		1	14592 至 14713	7296 至 7356
		2	14848 至 14969	7424 至 7484
		3	15104 至 15225	7552 至 7612
SPARC M10-4S	15	0	15360 至 15481	7680 至 7740
		1	15616 至 15737	7808 至 7868
		2	15872 至 15993	7936 至 7996
		3	16128 至 16249	8064 至 8124

*1 当 SPARC64 X+8 核 3.7 GHz 处理器搭载在 SPARC M10-1/M10-4 上时，括号中的数字是 pid 和 cid。

如需查看未分配给逻辑域的 CPU 的 CID 和 PID，使用 `ldm list-devices core` 命令。

下面是执行 `ldm list-devices core` 命令的示例。

[ID] 表示 CPU 内核 ID (CID)，[CPUSET] 表示与 CPU 内核 ID 相关联 CPU 的物理 CPU 编号 (PID)。

# ldm list-devices core		
CORE		
ID	%FREE	CPUSET
920	100	(1840, 1841)
924	100	(1848, 1849)
936	100	(1872, 1873)
940	100	(1880, 1881)
944	100	(1888, 1889)
<u>948</u>	100	<u>(1896, 1897)</u>
CID		PID

如需查看分配给逻辑域的 CPU 的 CID 和 PID，使用 `ldm list-domain -o core` 命令。

下面是分配给控制域的 CPU 内核的 CID 和 PID 的查看示例。

```
# ldm list-domain -o core primary
NAME
primary

CORE
  CID      CPUSET
  0         (0, 1)
  4         (8, 9)
  8         (16, 17)
  --- 省略 ---
  896       (1792, 1793)
  900       (1800, 1801)
  904       (1808, 1809)
  908       (1816, 1817)
  912       (1824, 1825)
  916       (1832, 1833)
  CID      PID
```

逻辑域内存分配的指导原则

将 4 GB 或更多的虚拟内存分配给逻辑域。

逻辑域的内存容量

如果变更逻辑域的内存未使用动态重新配置功能，待分配给逻辑域的内存容量可以 4 MB 为单位进行设置。
如果您需要使用动态重新配置变更逻辑域的内存，待分配给逻辑域的内存容量必须以 256 MB 为单位进行设置。

保存逻辑域配置信息

完成逻辑域配置后，将逻辑域配置信息保存到 XSCF。执行有关配置逻辑域的操作之前，也需保存逻辑域配置信息。执行操作前保存配置信息可确保逻辑域可以无误地恢复到其配置前的状态。

硬件故障或相似问题可能会造成逻辑域配置信息的缺失。因此，更改了来宾域配置信息后，请使用 Oracle VM Server for SPARC 的管理软件 Logical Domains Manager 的 ldm 命令来获取来宾域配置信息并将其保存到安全位置。将获取的配置信息复制到磁带设备或文件服务器上以防数据从磁盘上丢失。您也可以使用 ldm 命令根据获取的配置信息重新配置来宾域。有关详情，请参见所使用版本的《Oracle VM Server for SPARC Administration Guide》和《Oracle VM Server for SPARC Reference Manual》。

您可以备份 XSCF 中保存的配置信息。如需将配置信息备份到文件，请使用 XSCF 固件的 dumpconfig 命令。要从备份文件中恢复配置信息，请使用 restoreconfig 命令。

配置逻辑域时指定的项目

若需配置逻辑域，您需要定义不同名称和指定不同资源以及其编号。以下列表显示配置逻辑域时应定义或指定的项目。开始工作前，请确定这些项目。

- 虚拟控制台终端收集和分配设备的名称
- 虚拟磁盘服务器的名称

- 虚拟交换机服务的名称
- 虚拟网络设备接口的名称
- 虚拟设备的名称
- 虚拟磁盘的名称
- 来宾域的名称
- 逻辑域配置信息的名称
- 虚拟控制台终端端口号范围
- 用于虚拟交换机服务的设备
- 用于虚拟磁盘服务的设备
- 分配给来宾域的 CPU 数量
- 分配给来宾域的内存容量
- 分配给来宾域的虚拟控制台终端端口号

逻辑域关闭的顺序

若需使用顺序关闭功能并使用 XSCF 的 `poweroff` 命令指定各逻辑域的关闭顺序，请设置所创建的来宾域的关闭组。关闭组可指定逻辑域的关闭顺序。

关闭组范围从 0 到 15。号码越大的组，关闭越早。默认情况下，来宾域属于组 15。控制域始终属于组 0，并且不能更改。根据逻辑域的从属关系设置关闭顺序。

2.4.2 逻辑域重新配置注意事项

重新配置分配给逻辑域的硬件资源时，您需要注意以下事项。

设备侧的准备工作

重新配置虚拟 I/O 设备中的虚拟磁盘和虚拟网络之前，需执行以下操作：

- 卸载虚拟磁盘
- 对虚拟网络接口执行 `unplumb`

控制域和根域的延迟重新配置

如需将物理 I/O 添加到控制域和根域，或从控制域和根域删除物理 I/O，请通过延迟重新配置进行操作。一旦您通过延迟重新配置对分配给目标控制域和根域的物理 I/O 进行重新配置，将推迟控制域和根域的任何其他重新配置请求，直到控制域和根域被重新启动。

延迟重新配置的限制

对控制域或根域设置延迟重新配置模式以便重新配置硬件资源时，您将无法重新配置来宾域的硬件资源。在完成针对控制域或根域的重新配置操作后，对来宾域执行重新配置操作。

取消延迟重新配置

使用 `ldm cancel-operation reconf` 命令来取消对控制域或根域的延迟重新配置操作。有关 `ldm` 命令的详情，请参见《Oracle VM Server for SPARC Reference Manual》。

逻辑域的要求

- 当动态地重新配置物理分区时，确保控制域上的 Oracle Solaris 已经启动。
- 在控制域上配置系统卷的磁盘必须处于冗余配置中。在通过动态重新配置删除的系统板和通过动态重新配置未删除的系统板的磁盘上需要执行冗余配置。如果磁盘未处于冗余配置中，当已提前释放 I/O 设备以便删除搭载磁盘的系统板时，将无法启动控制域。

分配给虚拟机监控程序的内存容量

逻辑域可以使用的内存量小于机柜中实际安装的内存量。这是因为虚拟机监控程序操作使用的内存被预约在每个机柜。在系统配置中，设计域配置时应减去此虚拟机监控程序使用的内存。分配给虚拟机监控程序的内存量取决于型号、模块化架构配置和 PPAR-DR 模式设置。有关 PPAR DR 模式的详情，请参见 ["2.5.3 如何检查/设置 PPAR DR 模式"](#)。

分配给各物理分区的虚拟机监控程序的内存如下：

表 2-9 分配给虚拟机监控程序的内存量

型号	PPAR DR 模式	
	启用	禁用
SPARC M12-1/M12-2	2.5 GB	
SPARC M12-2S	2.5 GB + (1.5 GB x (包含物理分区的机箱的数量 - 1))	
SPARC M10-1/M10-4	-	2 GB
SPARC M10-4S	2 GB + (1.25 GB x (机柜数 - 1))	2 GB + (1.0 GB x (机柜数 - 1))

CPU 内核和内存的配置

在物理分区动态重新配置时，为了能够断开系统板，需要注意 CPU 内核数量的分配以及内存配置。有关详情，请参见 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中的 ["CPU 内核和内存的配置"](#)。

2.5 动态重新配置的条件和设置

2.5.1 动态重新配置系统配置时的注意事项

本节介绍使用物理分区动态重新配置 (PPAR DR) 功能配置系统时，必须考虑的事项。仅 SPARC M12-2S/M10-4S 支持动态重新配置功能。

更新 XCP 固件

XCP 2220 和更高版本支持物理分区动态重新配置 (PPAR DR) 功能。有关如何更新 XCP 固件的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统操作和管理指南》中的 "第16章 更新 XCP 固件"。

检查 PPAR DR 模式

若需使用物理分区动态重新配置 (PPAR DR) 功能，必须启用 PPAR DR 模式。使用 XSCF 固件的 `setpparmode` 命令可以设置此模式。有关如何检查 PPAR DR 模式设置的详情，请参见 "2.5.3 如何检查/设置 PPAR DR 模式"。

设备驱动程序和 I/O 设备的要求

若需使用物理分区动态重新配置 (PPAR DR) 功能，搭载到将要进行物理分区动态重新配置的系统板上的所有 PCIe 卡和 I/O 设备接口必须支持 PPAR DR。此外，所有已安装的 I/O 设备驱动程序和复制的软件必须支持物理分区动态重新配置。有关支持 PPAR DR 的 PCIe 卡、驱动程序和复制的软件，请参见《PCI Card Installation Guide》中的 "附录 A Cards That Support PCI Hot Plug and Dynamic Reconfiguration"。

如果搭载到将要进行 PPAR DR 的系统板上的所有 PCIe 卡和 I/O 设备不支持 PPAR DR，则对该系统板将无法执行物理分区动态重新配置。

软件要求

如果使用物理分区动态重新配置 (PPAR DR) 功能，逻辑域可能会暂时停止 (suspend)。因此，软件需要采取变更超时检测时间和暂时停止超时监控等措施。这些软件产品包括使用计时器或类似检测错误的软件，如：集群软件和系统监控软件。

分配给虚拟机监控程序的内存容量

逻辑域可以使用的内存量小于机柜中实际安装的内存量。这是因为虚拟机监控程序操作使用的内存被预约在每个机柜。在系统配置中，设计域配置时应减去此虚拟机监控程序使用的内存。分配给虚拟机监控程序的内存量取决于型号、模块化架构配置和 PPAR-DR 模式设置。有关 PPAR DR 模式的详情，请参见 "2.5.3 如何检查/设置 PPAR DR 模式"。

有关分配给各物理分区的虚拟机监控程序内存的详情，请参见 表 2-9。

内存容量的设置单元

当 Oracle VM Server for SPARC 的版本早于 3.2 时，若需启用物理分区动态重新配置，请将需要分配给逻辑域的内存容量设置为“分配给逻辑域的内核数量 × 256 MB”的倍数。有关设置步骤的示例详情，请参见“附录 A 使用物理分区动态重新配置的环境配置示例和步骤”。

CPU 内核和内存的配置

在物理分区动态重新配置时，为了能够断开 SPARC M12-2S/M10-4S，需要注意 CPU 内核数量的分配以及内存配置。有关详情，请参见“2.5.2 动态重新配置的系统操作注意事项”、“CPU 内核和内存的配置”。

时间同步

当执行物理分区动态重新配置时，逻辑域的系统时间可能会延迟。使用物理分区动态重新配置 (PPAR DR) 功能之前，请使用 NTP (Network Time Protocol) 同步时间。有关 Oracle Solaris NTP 服务器设置的详情，请查阅《Oracle Solaris System Management (Network Service)》(Oracle Solaris 10) 或《Oracle Solaris 11 Network Service Introduction》(Oracle Solaris 11)。

动态添加/删除 PCIe 根合成体

在 SPARC M10 的情况下，您可以添加或删除 PCIe 根合成体，而无需在运行 XCP 2240 或更高版本和 Oracle VM Server for SPARC 3.2 或更高版本的根域上停止 Oracle Solaris。在这种情况下，您仅可添加或删除安装了 Oracle Solaris 11.2 SRU11.2.8.4.0 或更高版本的根域。

因此，如需添加/删除仅被配置有 Oracle Solaris 10 的控制域的 PCIe 根合成体，在延迟重新配置后，必须重新启动 Oracle Solaris。所以，无法使用物理分区的动态重新配置。关于此功能的详情，请参见 Oracle VM Server for SPARC 3.2 或更高版本的《Oracle VM Server for SPARC Administration Guide》中的 "Assigning a Dynamic PCIe Bus"。

为了使用物理分区的动态重新配置来配置系统，包含进一步详细注意事项和最佳做法的文档已发布到 Fujitsu SPARC 服务器文档网站。

<http://www.fujitsu.com/global/products/computing/servers/unix/sparc/downloads/documents/>

请参见《Building High Availability System on Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Servers (Maintenance procedure)》中的 "1.2 Overview of PPAR DR" 和 "附录 A PPAR DR deleteboard Best Practice"。

2.5.2 动态重新配置的系统操作注意事项

本节介绍动态重新配置的系统操作中需要考虑的事项。

关于安装了 Oracle Solaris 11.3 或更高版本的逻辑域和内核区

- 如果 Oracle Solaris 11.3 安装在控制域，则需提前将以下行添加到控制域中的 /etc/system 文件后重新启动 SPARC M10-4S 系统。然而，如果 XCP 固件版本是 XCP 2260 或更高版本，则此步骤没有必要。

```
set enable_user_tick_stick_emulation = 0
```

- 如果安装了 Oracle Solaris 11.3 或更高版本的逻辑域和 Oracle Solaris 内核区（不包括控制域）位于 SPARC M12-2S/M10-4S 配置的物理分区中，则需在将以下行添加到适用的逻辑域和 Oracle Solaris 内核区中 /etc/system 文件后，重新启动逻辑域和 Oracle Solaris 内核区。

```
set uhrt_enable = 0x0
```

deleteboard 命令执行过程中 Oracle Solaris 的停止时间

当执行 `deleteboard` 命令时，逻辑域上的 Oracle Solaris 可能会暂时停止 (suspend)。在此期间，物理设备通讯和 I/O 设备可能会停止，并且在 Oracle Solaris 上运行的应用程序的操作也可能会停止，这可能会对任务处理，如使用远程设备关闭网络产生影响。因此，有必要通过提前确定各逻辑域可能停止的时间（挂起时间）来确定操作及其他事项。

最大挂起时间是，内存移动所造成的挂起/恢复处理时间和各 I/O 设备所需的挂起/恢复处理的时间。

近似值可以根据安装在将要执行 `deleteboard` 命令的机柜上的内存，以及安装在物理分区的物理 I/O 设备的类型和数量来计算。

使用以下公式进行计算。

挂起时间 = 内存移动时间 + 板载设备挂起/恢复时间之和 + PCI 卡挂起/恢复时间之和

由于内存的移动，挂起时间是每 TB 168 秒。为了进行计算，请使用 `showpparinfo` 命令检查安装在将要执行 `deleteboard` 命令的机柜内的内存数量。

在下面显示的输出示例中，如果系统板 (BB#01) 断开连接，由于 PSB 01-0 的安装值为 128 GB（显示在 "Memory:" 下），则挂起时间是 21 秒。

$$128 \text{ (GB)} \times 168 \text{ (秒)} / 1024 \text{ (GB)} = 21 \text{ (秒)}$$

```
XSCF> showpparinfo -p 0
PPAR#00 Information:
-----
....
Memory:
-----
install
PID PSB size GB
00 00-0 128
00 01-0 128
IO Devices:
-----
PID PSB device
....
```

从所有安装在物理分区的物理 I/O 设备的类型和数量计算物理 I/O 设备的挂起时间。板载设备的挂起/恢复处理时间如下所示。

- 对于 SPARC M12

每个机箱 5.2 秒
示例) 2BB 配置是 10.4 秒
- 对于 SPARC M10
每个机箱 21.4 秒
示例) 2BB 配置是 42.8 秒

有关 PCI 卡挂起/恢复处理时间的详细信息, 请参见您的服务器的《PCI Card Installation Guide》中的 "附录 A Cards That Support PCI Hot Plug and Dynamic Reconfiguration"。

组合实时迁移

请勿同时执行实时迁移和物理分区动态重新配置。

CPU 内核和内存的配置

如果 CPU 或内存配置不符合下列条件, 执行 **deleteboard** 命令可能会显示以下信息。采取措施, 同时查阅各条件。

Some domain will be reduced with DR operation. But reducing resource is not allowed.

- 当仅配置了控制域时:
如果您使用 **deleteboard** 命令通过物理分区动态重新配置删除系统板并且以下两种情况都适用, 将出现上述消息, 此操作将出现错误, 异常结束:
 - 当逻辑域配置信息仅配置了控制域时, 如出厂默认
 - 当多个系统板上的 CPU 内核和内存区域被分配给控制域时。

为了防止发生此种情况, 请执行以下操作:

- 指定 **deleteboard** 命令中的 **-m unbind=resource** 选项来动态删除 CPU 内核和内存区。
- 使用 **ldm remove-core** 或 **ldm remove-memory** 命令提前删除 CPU 和内存, 以便获取需要移除的 BB (PSB) 的可用资源空间。

若需执行 PPAR DR 而不减少控制域使用的 CPU 内核数量或内存量, 则有必要设置该配置, 以便提前获取分配给域的 CPU 内核数量和内存区容量的可用空间。

Oracle VM Server for SPARC 3.2 或更高版本支持 **deleteboard** 命令的 **-m unbind=resource** 选项。在 Oracle VM Server for SPARC 3.4 或更高版本中支持 PPAR DR 策略。有关策略的详细信息, 请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "8.15 设置物理分区动态重新配置策略"。

- 当配置了控制域和逻辑域时:

[CPU 内核]

如果使用 **deleteboard** 命令删除后, 分配给逻辑域的 CPU 内核数量超过剩余的 CPU 内核数量, **deleteboard** 命令将失败。这是因为未分配的 CPU 内核是已分配的 CPU 内核的目标, 该未分配的 CPU 内核的数量不足。

为了防止发生此种情况，请执行以下操作：

- 当未分配的 CPU 内核与使用 `deleteboard` 命令将要删除 CPU 内核剩余的数量相同时，请提前配置逻辑域。
- 指定 `deleteboard` 命令中的 `-m unbind=resource` 选项来动态删除 CPU 内核和内存区。
- 使用 `ldm remove-core` 或 `ldm remove-memory` 命令提前删除 CPU 内核和内存，以便获取需要移除的系统板的资源可用空间。

Oracle VM Server for SPARC 3.2 或更高版本支持 `deleteboard` 命令的 `-m unbind=resource` 选项。在 Oracle VM Server for SPARC 3.4 或更高版本中支持 PPAR DR 策略。有关策略的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "8.15 设置物理分区动态重新配置策略"。

检查 CPU 内核数量的分配状况，如下所示：

1.检查分配给各逻辑域的 CPU 内核的总数。

这是 CPU 内核的总数，"%FREE" 字段包含除 "100" 以外的数值。您可以通过在控制域上执行 `ldm list-devices -a core` 命令来检查此数量。下面的示例执行了 `ldm list-devices -a -p core` 命令来显示分配给逻辑域的 CPU 内核的总数。

```
# ldm list-devices -a core
CORE
  ID      %FREE    CPUSSET
  0        0      (0, 1)
  4        0      (8, 9)
  8        0     (16, 17)
 12        0     (24, 25)
(省略)
# ldm list-devices -a -p core | egrep -v "CORE|VERSION|free=100" | wc -l
112
```

2.检查尚未移除的系统板上的 CPU 内核的总数。

这是尚未删除的 PSB 数量的 "内核" 总数。您可以通过在 XSCF 上执行 `showpparinfo` 命令来检查此数量。以下为 SPARC M10-4S 执行 `showpparinfo` 命令的示例。

```
XSCF> showpparinfo -p 0
PPAR#00 Information:
-----
CPU(s)                :          8
CPU Cores              :         128
CPU Threads            :         256
Memory size (GB)       :         256
CoD Assigned (Cores)   :         256
CPU(s) :
-----
PID  PSB    CPU#   Cores   Threads
00   00-0    0       16       32
00   00-0    1       16       32
00   00-0    2       16       32
00   00-0    3       16       32
```

00	01-0	0	16	32
00	01-0	1	16	32
00	01-0	2	16	32
00	01-0	3	16	32
(省略)				

使用下面的公式，计算释放 SPARC M12-2S/M10-4S 机箱后，将要发生的 CPU 内核不足的数量。

CPU 内核不足的数量 = 逻辑域中使用的内核数量 (1) - 释放后的物理内核的数量 (2)

如果没有足够的内核，您必须使用 `ldm remove-core` 命令删除分配给逻辑域的 CPU 内核来减少内核的数量。

[内存]

假设当将需要删除的系统板上的内存区分配给逻辑域时，您通过物理分区动态重新配置执行了 `deleteboard` 命令。在这种情况下，分配给逻辑域内存区的内容将被重新分配给尚未被删除的系统板上的内存区，以便移动该内容。

因此，如果在移动目标（在未删除的系统板）上不存在比所移动内存容量更大的可用空间，`deleteboard` 命令将出现错误而异常终止。

为了防止发生此种情况，请执行以下操作：

- 指定 `deleteboard` 命令中的 `-m unbind=resource` 选项来动态删除 CPU 内核和内存区。
- 使用 `ldm remove-memory` 命令提前删除内存，以便获取充足的需要移除的系统板的资源可用空间。

Oracle VM Server for SPARC 3.2 或更高版本支持 `deleteboard` 命令的 `-m unbind=resource` 选项。在 Oracle VM Server for SPARC 3.4 或更高版本中支持 PPAR DR 策略。有关策略的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "8.15 设置物理分区动态重新配置策略"。

检查内存区的使用状况，如下所示：

- 1.检查内存的连续区域（内存块）的使用状况。
执行 `prtdiag` 命令来检查内存的物理地址和具有模块化架构配置的 SPARC M10-4S 之间的对应关系。

```
# prtdiag
(省略)
===== Physical Memory Configuration =====
Segment Table:
```

Base Segment Interleave Bank Contains					
Address Size Factor Size Modules					

0x7e0000000000	32	GB	4	8 GB	/BB0/CMUL/CMP0/MEM00A
(省略)					
0x7c0000000000	32	GB	4	8 GB	/BB0/CMUL/CMP1/MEM10A
(省略)					
0x7a0000000000	32	GB	4	8 GB	/BB0/CMUU/CMP0/MEM00A
(省略)					
0x780000000000	32	GB	4	8 GB	/BB0/CMUU/CMP1/MEM10A
(省略)					
0x760000000000	32	GB	4	8 GB	/BB1/CMUL/CMP0/MEM00A
(省略)					
0x740000000000	32	GB	4	8 GB	/BB1/CMUL/CMP1/MEM10A
(省略)					
0x720000000000	32	GB	4	8 GB	/BB1/CMUU/CMP0/MEM00A
(省略)					
0x700000000000	32	GB	4	8 GB	/BB1/CMUU/CMP1/MEM10A
(省略)					

本示例的结果按内存中物理地址的升序重新排列。下表列出了物理地址和 SPARC M10-4S 之间的对应关系。

表 2-10 物理地址和 SPARC M10-4S 之间对应关系的示例

基地址（物理地址）	SPARC M10-4S
0x700000000000...	模块化架构 BB-ID#01
0x720000000000...	模块化架构 BB-ID#01
0x740000000000...	模块化架构 BB-ID#01
0x760000000000...	模块化架构 BB-ID#01
0x780000000000...	模块化架构 BB-ID#00
0x7a0000000000...	模块化架构 BB-ID#00
0x7c0000000000...	模块化架构 BB-ID#00
0x7e0000000000...	模块化架构 BB-ID#00

接下来，在控制域上执行 `ldm list-devices -a memory` 命令来显示分配给各逻辑域的内存区，以及所有未使用的内存区。

```
# ldm list-devices -a memory
MEMORY
PA SIZE BOUND
0x700000000000 24G root-dom1
0x700600000000 8G
0x720000000000 32G guest0
0x740000000000 32G guest1
0x760000800000 1272M _sys_
0x760050000000 24G root-dom0
```

0x760650000000	6912M
0x780000000000	32G
0x7a0000000000	32G
0x7c0000000000	32G
0x7e0000800000	1272M _sys_
0x7e0050000000	512M _sys_
0x7e0070000000	256M _sys_
0x7e0080000000	14G primary
0x7e0400000000	16G

从上述结果和在 "表 2-10 物理地址和 SPARC M10-4S 之间对应关系的示例" 中的物理位置，可以确定内存块的使用状况，如下所示。

表 2-11 内存块使用状况的示例

SPARC M10-4S	物理地址	容量	逻辑域
模块化架构 BB-ID#01 (需要更换)	0x700000000000	24 GB	root-dom1
	0x700600000000	8 GB	未分配
	0x720000000000	32 GB	guest0
	0x740000000000	32 GB	guest1
	0x760050000000	24 GB	root-dom0
	0x760650000000	6912 MB	未分配
模块化架构 BB-ID#00	0x780000000000	32 GB	未分配
	0x7a0000000000	32 GB	未分配
	0x7c0000000000	32 GB	未分配
	0x7e0080000000	14 GB	primary
	0x7e0400000000	16 GB	未分配

- 2.检查移动源内存块的容量和数量。
- 参照内存块使用状况的检查结果时，检查分配给需要更换的 SPARC M10-4S 的内存块（在下文中称为 "源内存块"）。
在 "表 2-11 内存块使用状况的示例" 中，您可以确定在模块化架构 BB-ID#01 侧分配给逻辑域的内存块的数量是 32 GB × 2（分配给 guest0 和 guest1）和 24 GB × 1（root-dom0）。

注 - 如果分配模块化架构 BB-ID#01 的 I/O 的根域 root-dom1 未绑定并且处于 inactive 状态，在该模块化架构断开前，您可以将 root-dom1 排除在移动之外。

- 3.检查空内存块。
- 接下来，根据在步骤 1 中获得的检查结果，检查未被分配给未断开的 SPARC M10-4S 上的逻辑域的内存块（在下文中称为 “空内存块”）。
对于 "表 2-11 内存块使用状况的示例" 中给出的示例，可以确定空内存块的数量是 32 GB × 3 和 16 GB × 1。
- 4.检查内存块是否可移动。
- 使用在步骤 2 和步骤 3 中获得的检查结果，检查源内存块是否可以移动到空内存
- 第2章 系统运行的条件和设置

55

块。
如果空内存块的容量等于或大于源内存块的容量，则可以移动。

当目标包含空资源时

在 "表 2-11 内存块使用状况的示例" 示例中，有 32 GB × 3 空内存块，分别是 guest0 (32 GB)、guest1 (32 GB) 和 root-dom0 (24 GB) 的目标。因此，可以确定需要配置内存，以便可断开模块化架构 BB-ID#01。在 "表 2-12 目标备选内存块" 中进行汇总。

表 2-12 目标备选内存块

SPARC M10-4S	容量	逻辑域	目标备选
模块化架构 BB-ID#01 (需要更换)	24 GB	root-dom1	-
	8 GB	未分配	-
	32 GB	guest0	模块化架构 BB-ID#00 的 32 GB
	32 GB	guest1	模块化架构 BB-ID#00 的 32 GB
	24 GB	root-dom0	模块化架构 BB-ID#00 的 32 GB
	6912 MB	未分配	-
模块化架构 BB-ID#00	32 GB	未分配	被移到此处
	32 GB	未分配	被移到此处
	32 GB	未分配	被移到此处
	14 GB	primary	-
	16 GB	未分配	由于容量低排除目标备选

目标没有空资源时

例如，在 "表 2-13 无目标的内存块配置示例" 的配置中，源内存块的数量是 32 GB × 2 和 24 GB × 2。同时，目标处的可用内存块是 32 GB × 3 和 16 GB × 1。

因此，可移动一个 32 GB (guest0) 内存块和两个 24 GB（两个 guest1、guest2 和 root-dom0）内存块。

然而，剩下的空目标内存块的数量是 16 GB×1 和 8 GB×2，在 24-GB 内存块已经被移动到 32 GB 的内存块后而获取。因此，无法移动分配 24-GB 内存块的任何 guest1、guest2 和 root-dom0。在这种情况下，您需要减小逻辑域中可能不可移动的内存块的容量为等于或小于目标中内存块的容量。

在上述示例中，您必须执行 ldm remove-memory 命令来将 guest1、guest2 和 root-dom0 的任意一个从 24 GB 更改为 16 GB 或更小。

表 2-13 无目标的内存块配置示例

SPARC M10-4S	容量	逻辑域	目标备选
模块化架构 BB-ID#01 (需要更换)	24 GB	guest2	无法移动
	8 GB	root-dom1	-
	32 GB	guest0	模块化架构 BB-ID#00 的 32 GB
	32 GB	guest1	无法移动
	24 GB	root-dom0	无法移动
	6912 MB	未分配	-
模块化架构 BB-ID#00	32 GB	未分配	被移到此处
	32 GB	未分配	guest1、guest2 和 root-dom0 (24 GB) 中的任意一个移到此处，剩余 8 GB。
	32 GB	未分配	guest1、guest2 和 root-dom0 (24 GB) 中的任意一个移到此处，剩余 8 GB。
	14 GB	primary	-
	16 GB	未分配	由于容量低排除目标备选

恢复模式启用时的动态重新配置操作

- 假设您在域配置已经在降级的配置中被恢复的条件下，添加使用物理分区的动态重新配置的系统板。添加的资源不会自动分配给任何逻辑域。请手动分配添加的资源。另外，执行 `ldm set-sconfig` 命令选择初始域配置，然后使用 XSCF 固件的 `poweron` 和 `poweroff` 命令重新启动物理分区。
- 假设域配置在降级的配置中被恢复后，在物理分区 (PPAR) 接通电源时，您使用 `deleteboard` 命令删除系统板 (PSB)。如果 Oracle VM Server for SPARC 的版本低于 3.2，`deleteboard` 命令可能失败。域配置在降级的配置中被恢复后，请勿使用物理分区的动态重新配置删除系统板。

组合 Oracle Solaris 内核区域

当 Oracle Solaris 内核区域在物理分区 (PPAR) 的任何逻辑域上运行时，将无法执行物理分区动态重新配置。请停止 Oracle Solaris 内核区域，然后执行物理分区动态重新配置。

对于分配虚拟服务的逻辑域

如果分配虚拟服务的逻辑域是以下任何一种，有必要事先分别使用 `ldm remove-vdsdev` 命令、`ldm remove-vsw` 命令、`ldm remove-vdisk` 命令以及 `ldm remove-vnet` 命令移除分配给物理 I/O 的虚拟磁盘服务器设备 (vdsdev)、虚拟网络交换机 (vsw)、以及分配给它们的虚拟磁盘 (vdisk) 和虚拟网络 (vnet)。

- 当使用 `ldm remove-io` 命令动态删除 PCIe 端点时
- 当通过物理分区的动态重新配置动态删除逻辑域的物理 I/O 时

PPAR DR 操作之后变更 USB 设备路径

如果您使用物理分区动态重新配置功能删除了一个系统板 (PSB<BB>), 然后添加了一个, 则删除/添加 PSB (BB) 的 USB 设备路径的 "usb@4,1" 部分变更为 "usb@4", 如下例所示。

- 外部 DVD 驱动器 (前) 示例

删除之前: `/pci@8000/pci@4/pci@0/pci@1/pci@0/usb@4,1/hub@2/cdrom@1/disk@0,0:a`

添加之后: `/pci@8000/pci@4/pci@0/pci@1/pci@0/usb@4/hub@2/cdrom@1/disk@0,0:a`

- 远程存储用 DVD 驱动器示例

删除之前: `/pci@8000/pci@4/pci@0/pci@1/pci@0/usb@4,1/storage@3/disk@0,0:a`

添加之后: `/pci@8000/pci@4/pci@0/pci@1/pci@0/usb@4/storage@3/disk@0,0:a`

但是, 在重新启动分配设备路径发生变化的 USB 设备的逻辑域后, 设备的路径返回到删除前的状态。

对于使用物理分区动态重新配置功能未删除/添加的 PSB (BB), 连接到该系统板的 USB 设备不会发生此现象。

为了使用物理分区的动态重新配置来配置系统, 包含更多详细注意事项和最佳做法的文档已发布到 Fujitsu SPARC 服务器文档网站。

<http://www.fujitsu.com/global/products/computing/servers/unix/sparc/downloads/documents/>

请参见《Building High Availability System on Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 Servers (Maintenance procedure)》中的 "1.2 Overview of PPAR DR" 和 "附录 A PAR DR deleteboard Best Practice"。

2.5.3 如何检查/设置 PPAR DR 模式

本节介绍如何检查 PPAR DR 模式, 以及如何进行设置。

重要 - 变更 PPAR DR 模式设置之前, 首先保存逻辑域配置信息。

假设您使用 `setpparmode` 命令, 将 PPAR DR 模式设置从禁用变更为启用, 或从启用变更为禁用。若需应用该设置, 您必须重置物理分区。此重置将逻辑域配置信息恢复到出厂默认设置。重新配置逻辑域需要以前已保存的逻辑域配置信息。因此, 在变更 PPAR DR 模式设置之前始终需要保存数据。有关如何保存逻辑域配置信息以及重新配置逻辑域的详情, 请参见 "A.1 将不支持物理分区动态重新配置的系统更新到支持物理分区动态重新配置的系统" 中给出的配置示例。

设置 PPAR DR 模式的流程如下:

1. 保存逻辑域配置信息。

在下面的示例中, 将逻辑域配置信息保存为 XML 文件, 称作 `file.xml`。

注 - 如果尚未配置逻辑域, 没有必要进行此步骤。

```
# ldm list-constraints -x > file.xml
```

2. 执行 `showpparmode` 命令来检查 PPAR DR 模式设置。

PPAR DR (Current) 的显示表示当前 PPAR DR 模式值。PPAR DR (Next) 表示所

设置的在物理分区下次启动后 PPAR DR 模式值。

在下面的示例中，当物理分区下次启动时，将关闭物理分区的电源，禁用 PPAR DR 功能。

```
XSCF> showpparmode -p PPAR_ID
Host-ID                :9006000a
Diagnostic Level        :min
Message Level           :normal
Alive Check             :on
Watchdog Reaction       :reset
Break Signal            :off
Autoboot (Guest Domain) :on
Elastic Mode            :off
IOreconfigure           :false
PPAR DR (Current)       :-
PPAR DR (Next)          :off
```

3. 执行 **setpparmode** 命令来设置 PPAR DR 模式。

下面的示例启用 PPAR DR 功能。

```
XSCF> setpparmode -p PPAR_ID -m ppar_dr=on
Diagnostic Level        :max      -> -
Message Level          :normal   -> -
Alive Check            :on       -> -
Watchdog Reaction       :reset    -> -
Break Signal           :on       -> -
Autoboot (Guest Domain) :on      -> -
Elastic Mode           :off      -> -
IOreconfigure          :true     -> -
PPAR DR                :off      -> on
The specified modes will be changed.
Continue? [y|n] :y
configured.
Diagnostic Level        :max
Message Level          :normal
Alive Check            :on (alive check:available)
Watchdog Reaction       :reset (watchdog reaction:reset)
Break Signal           :on (break signal:non-send)
Autoboot (Guest Domain) :on
Elastic Mode           :on
IOreconfigure          :false
PPAR DR                :on
```

4. 执行 **poweron** 命令来开启物理分区的电源。

```
XSCF> poweron -p PPAR_ID
```

注 - 逻辑域配置信息恢复到出厂默认设置。重新启动物理分区后，使用在步骤 1 中保存的逻辑域配置信息重新配置逻辑域。有关如何采用配置信息重新配置逻辑域的详情，请参见 "A.1 将

不支持物理分区动态重新配置的系统更新到支持物理分区动态重新配置的系统"中给出的配置示例。

5. 执行 **showpparmode** 命令来检查当前和下一个设置值。

下面的示例显示, PPAR DR 功能当前已启用, 并且当下一次启动物理分区时会保持启用状态。

```
XSCF> showpparmode -p PPAR_ID
Host-ID                :9006000a
Diagnostic Level        :min
Message Level          :normal
Alive Check            :on
Watchdog Reaction       :reset
Break Signal           :off
Autoboot (Guest Domain) :on
Elastic Mode           :off
IOreconfigure          :false
PPAR DR (Current)      :on
PPAR DR (Next)         :on
```

2.6 使用 SPARC64 X+ 处理器时的注意事项

本节介绍当配置包含 SPARC64 X+ 处理器的系统时, 应该考虑的事项。

注 - 这些注意事项适用于 SPARC M10。它们不适用于 SPARC M12。

2.6.1 更新 XCP 固件

有关支持将 SPARC64 X+ 处理器整合到 SPARC M10 的系统的 XCP 固件和 Oracle Solaris 的版本详情, 请参见 ["1.5.4 软件条件和确认方法"](#) 和最新的《SPARC M10 系统 产品说明》。

2.6.2 CPU 运行模式设置

在 SPARC M10 中, SPARC64 X+ 处理器可以提供与 SPARC64 X 处理器相同的功能。本节介绍关于 CPU 运行模式的设置和需要考虑的事项。

CPU 运行模式设置流程

有关 CPU 运行模式的详情, 请参见 ["1.6.1 CPU 操作类型和 CPU 运行模式"](#)。通过执行 XSCF 固件的 setpparmode 命令, 对各个物理分区均可配置 CPU 运行模式。

使用 `setpparmode` 命令可设置两种 CPU 运行模式 (cpumode): "auto" 和 "compatible"。默认情况下 CPU 运行模式设置为 "auto"。
各模式的操作如下。

- **auto 模式**
此种模式启用数据库加速指令，是由 SPARC64 X+ 处理器提供的扩展功能。然而，在 SPARC64 X+ 处理器和 SPARC64 X 处理器的混合配置中，SPARC64 X+ 处理器可在 SPARC64 X 兼容模式下运行。如果您在 SPARC64 X+ 处理器和 SPARC64 X 处理器的混合配置中通过动态重新配置执行维护，请设置为 `compatible` 模式。
- **compatible 模式**
SPARC64 X+ 处理器可在 SPARC64 X 兼容模式下运行。若需配置混合使用 SPARC64 X+ 处理器和 SPARC64 X 处理器的物理分区，设置为此模式。

表 2-14 显示了物理分区配置、CPU 运行模式设置和 CPU 运行之间的关系。

表 2-14 物理分区中 CPU 配置和 CPU 运行模式

物理分区配置	CPU 运行模式设置	CPU 操作
SPARC64 X+	auto	采用 SPARC64 X+ 功能运行
SPARC64 X+	compatible	在 SPARC64 X 兼容模式下运行
SPARC64 X+/X	auto 或 compatible	SPARC64 X+ 在 SPARC64 X 兼容模式下运行
SPARC64 X	auto 或 compatible	采用 SPARC64 X 功能运行

当目标物理分区电源关闭时，变更 CPU 运行模式设置。

重要 - 变更 CPU 运行模式设置之前，首先保存（备份）逻辑域配置信息。
在 Oracle VM Server for SPARC 中的逻辑域配置信息包括有关逻辑域中使用的 CPU 运行模式的信息。如果您将采用 SPARC64 X+ 功能运行的逻辑域的逻辑域配置信息应用到采用 SPARC64 X 功能运行的物理分区上，将发生配置信息不匹配。然后，XSCF 将逻辑域配置信息切换到出厂默认设置，并启动物理分区。
若需将已经切换到出厂默认设置的逻辑域配置信息重新配置到原来的配置设置，需要以前已经保存（备份）的逻辑域配置信息。因此，在使用 `setpparmode` 命令变更 CPU 运行模式之前，您必须保存（备份）逻辑域配置信息。有关如何保存（备份）逻辑域配置信息以及重新配置逻辑域的详情，请参见 "A.1 将不支持物理分区动态重新配置的系统更新到支持物理分区动态重新配置的系统" 中给出的配置示例。有关在哪些情况下逻辑域需要重新配置的详情，请参见 "B.2 CPU 运行模式补充资料"。

若需变更 CPU 运行模式，请使用 `setpparmode` 命令。下面显示将 CPU 运行模式变更为 "compatible" 的程序示例。

1. 关闭物理分区的电源。

```
XSCF> poweroff -p PPAR_ID
```

2. 执行 `showpparmode` 命令，检查当前 CPU 运行模式（CPU 模式）设置。

```
XSCF> showpparmode -p PPAR_ID
Host-ID                               :9007002b
Diagnostic Level                       :min
```

Message Level	:normal
Alive Check	:on
Watchdog Reaction	:reset
Break Signal	:on
Autoboot(Guest Domain)	:on
Elastic Mode	:off
IOreconfigure	:false
CPU Mode	:auto
PPAR DR(Current)	:-
PPAR DR(Next)	:on

3. 执行 **setpparmode** 命令，将 CPU 运行模式（CPU 模式）从 "auto" 变更为 "compatible"。

```
XSCF> setpparmode -p PPAR_ID -m cpumode=compatible
```

4. 执行 **showpparmode** 命令，检查 CPU 运行模式（CPU 模式）是否已经设置为 "compatible"。

```
XSCF> showpparmode -p PPAR_ID
Host-ID                :9007002b
Diagnostic Level        :min
Message Level           :normal
Alive Check             :on
Watchdog Reaction       :reset
Break Signal            :on
Autoboot(Guest Domain)  :on
Elastic Mode            :off
IOreconfigure           :false
CPU Mode                :compatible
PPAR DR(Current)        :-
PPAR DR(Next)           :on
```

5. 开启物理分区的电源。

```
XSCF> poweron -p PPAR_ID
```

检查 CPU 操作类型

如需检查 CPU 操作类型，请在 Oracle Solaris 上执行 **prtdiag** 或 **psrinfo** 命令。

当采用 SPARC64 X+ 功能 CPU 运行时，**prtdiag** 命令在 [Virtual CPUs] 内输出 "SPARC64-X+"。当采用 SPARC64 X 功能 CPU 运行时，该命令在 [Virtual CPUs] 内输出 "SPARC64-X"。

指定 **-pv** 选项，执行 **psrinfo** 命令。当采用 SPARC64 X+ 功能 CPU 运行时，在所提供的物理处理器信息内输出 "SPARC64-X+"。同时，当采用 SPARC64 X 功能 CPU 运行时，将输出 "SPARC64-X"。

以下显示了 **prtdiag** 和 **psrinfo** 命令的示例。

■ 采用 **SPARC64 X+** 功能运行时
[prtdiag 命令的显示示例]

```
# prtdiag
System Configuration:      Oracle Corporation      sun4v  SPARC  M10-4S
Memory size:  391168 Megabytes

===== Virtual CPUs =====

CPU ID Frequency Implementation      Status
-----
0      3700 MHz SPARC64-X+            on-line
:
```

[psrinfo 命令的显示示例]

```
# psrinfo -pv
The physical processor has 16 cores and 32 virtual processors
(0-31)
    The core has 2 virtual processors (0 1)
    The core has 2 virtual processors (2 3)
    The core has 2 virtual processors (4 5)
    The core has 2 virtual processors (6 7)
    The core has 2 virtual processors (8 9)
    The core has 2 virtual processors (10 11)
    The core has 2 virtual processors (12 13)
    The core has 2 virtual processors (14 15)
    The core has 2 virtual processors (16 17)
    The core has 2 virtual processors (18 19)
    The core has 2 virtual processors (20 21)
    The core has 2 virtual processors (22 23)
    The core has 2 virtual processors (24 25)
    The core has 2 virtual processors (26 27)
    The core has 2 virtual processors (28 29)
    The core has 2 virtual processors (30 31)
    SPARC64-X+ (chipid 0, clock 3700 MHz)
    :
    :
```

■ 采用 **SPARC64 X** 功能运行时
[prtdiag 命令的显示示例]

```
# prtdiag
System Configuration:      Oracle Corporation      sun4v  SPARC  M10-4S
Memory size:  391168 Megabytes

===== Virtual CPUs =====

CPU ID Frequency Implementation      Status
-----
0      3700 MHz SPARC64-X            on-line
:
```

```
# psrinfo -pv
The physical processor has 16 cores and 32 virtual processors
(0-31)
  The core has 2 virtual processors (0 1)
  The core has 2 virtual processors (2 3)
  The core has 2 virtual processors (4 5)
  The core has 2 virtual processors (6 7)
  The core has 2 virtual processors (8 9)
  The core has 2 virtual processors (10 11)
  The core has 2 virtual processors (12 13)
  The core has 2 virtual processors (14 15)
  The core has 2 virtual processors (16 17)
  The core has 2 virtual processors (18 19)
  The core has 2 virtual processors (20 21)
  The core has 2 virtual processors (22 23)
  The core has 2 virtual processors (24 25)
  The core has 2 virtual processors (26 27)
  The core has 2 virtual processors (28 29)
  The core has 2 virtual processors (30 31)
    SPARC64-X (chipid 0, clock 3700 MHz)
The physical processor has 16 cores and 32 virtual processors
(32-63)
  The core has 2 virtual processors (32 33)
```

2.6.3 PPAR 内 SPARC64 X+ 处理器和 SPARC64 X 处理器的混合配置条件

使用 SPARC M10-4S 时，您可以将搭载在一个 SPARC M10-4S 上的 SPARC64 X+ 处理器和搭载在其他 SPARC M10-4S 上的 SPARC64 X 处理器一起用于一个物理分区中。对于混合的配置，您需要将支持 SPARC64 X+ 处理器的 XCP 固件应用到组成模块化架构模块的所有 SPARC M10-4S 单元上。根据您的系统配置，支持 SPARC64 X+ 处理器的 XCP 固件将自动被应用。

如果您有一个由搭载 SPARC64 X 处理器的 SPARC M10-4S 单元组成的物理分区，并希望将搭载 SPARC64 X+ 处理器的 SPARC M10-4S 添加到此物理分区，则您必须事先更新搭载 SPARC64 X 处理器的 SPARC M10-4S 上的 XCP 固件。

如果您有一个由搭载 SPARC64 X+ 处理器的 SPARC M10-4S 单元组成的物理分区，并希望将搭载 SPARC64 X 处理器的 SPARC M10-4S 添加到此物理分区，则搭载 SPARC64 X 处理器的 SPARC M10-4S 上的 XCP 固件将自动更新到适用于主 XSCF 的 XCP 固件的版本。

注 -有关支持整合 SPARC64 X+ 处理器的系统的 XCP 固件的版本详情，请参见最新 XCP 版本的《Fujitsu SPARC M10 系统 产品说明》。

2.6.4 CPU 运行模式和物理分区动态重新配置之间的关系

本节介绍 CPU 运行模式和通过物理分区动态重新配置添加或删除 SPARC M10-4S 之间的关系。

CPU 运行模式和通过物理分区动态重新配置的添加之间的关系

搭载 SPARC SPARC64 X+ 处理器的系统板只能通过动态重新配置添加到 SPARC M10 中以 SPARC SPARC64 X+ 功能运行的物理分区。
您无法动态地整合不支持 SPARC64 X+ 功能的 SPARC64 X 处理器。
如果您尝试通过动态重新配置添加一个搭载 SPARC64 X 处理器的系统板，添加处理将失败，并显示以下信息。

The current configuration does not support this operation.

对于采用 SPARC64 X 功能运行的物理分区，通过动态重新配置您可以动态地添加搭载 SPARC64 X 或 SPARC64 X+ 处理器的系统板。

表 2-15 列出了通过动态重新配置可添加的系统板的物理分区配置、CPU 操作类型和 CPU 配置之间的关系。

表 2-15 CPU 运行模式和通过物理分区动态重新配置的添加处理之间的关系 (SPARC M10)

物理分区配置	CPU 操作类型	搭载 SPARC64 X+ 处理器的系统板的动态重新配置	搭载 SPARC64 X 处理器的系统板的动态重新配置
SPARC64 X+	采用 SPARC64 X+ 功能运行	是	否
SPARC64 X+	在 SPARC64 X 兼容模式下运行	是	是
混合使用 SPARC64 X+/SPARC64 X	SPARC64 X+ 在 SPARC64 X 兼容模式下运行	是	是
SPARC64 X	采用 SPARC64 X 功能运行	是	是

CPU 运行模式和通过物理分区动态重新配置的删除之间的关系

当 CPU 采用 SPARC64 X+ 或 SPARC64 X 功能在 SPARC M10 中运行时，通过动态重新配置可删除系统板。

表 2-16 列出了通过动态重新配置可删除的系统板的物理分区配置、CPU 操作类型和 CPU 配置之间的关系。

表 2-16 CPU 运行模式和通过物理分区动态重新配置的删除处理之间的关系 (SPARC M10)

物理分区配置	CPU 操作类型	搭载 SPARC64 X+ 处理器的系统板的动态重新配置	搭载 SPARC64 X 处理器的系统板的动态重新配置
SPARC64 X+	采用 SPARC64 X+ 功能运行	是	- (*1)
SPARC64 X+	在 SPARC64 X 兼容模式下运行	是	- (*1)

表 2-16 CPU 运行模式和通过物理分区动态重新配置的删除处理之间的关系 (SPARC M10) (续)

物理分区配置	CPU 操作类型	搭载 SPARC64 X+ 处理器的系 统板的动态重新配置	搭载 SPARC64 X 处理器的系统 板的动态重新配置
混合使用 SPARC64 X+/SPARC64 X	在 SPARC64 X 兼容模式 下运行	是	是 (*2)
SPARC64 X	采用 SPARC64 X 功能运行	- (*1)	是

*1 在此配置中，此操作不可用。

*2 对于 SPARC64 X+ 处理器和 SPARC64 X 处理器混合配置的物理分区，请执行 setpparmode 命令将 CPU 运行模式设置为 "compatible"。假设您通过动态重新配置删除搭载 SPARC64 X 处理器的系统板后，将一个仅配置有搭载 SPARC64 X+ 处理器的系统板的系统的 CPU 运行模式设置为 "auto"。此种情况下，如果您关闭电源，然后开启电源，或重置物理分区，CPU 将采用 SPARC64 X+ 的功能重新启动。这样，即使您尝试通过动态重新配置添加搭载 SPARC64 X 处理器的系统板，也无法这样操作，因为系统正在采用 SPARC64 X+ 的功能运行。因此，如果您考虑某项操作，该操作涉及组合了 SPARC64 X+ 处理器和 SPARC64 X 处理器混合配置的物理分区系统板的活动状态更换，则将 CPU 运行模式设置为 "compatible"。

域配置的操作

- 本章介绍域配置的操作和命令。
- 与物理分区配置有关的操作和命令
 - 与逻辑域配置有关的操作和命令

3.1 与物理分区配置有关的操作和命令

XSCF 提供两种物理分区配置的用户界面：基于 XSCF shell 的命令行和基于 XSCF Web 的网页浏览器。本节介绍主要用于物理分区配置相关操作的 XSCF shell 命令。

有两种用于配置物理分区的 XSCF shell 命令：与显示相关的命令和与操作相关的命令。

表 3-1 与显示相关的命令

命令的名称	功能概述
showpcl	显示物理分区配置信息。
showpparstatus	显示物理分区状态。
showboards	显示系统板状态。
showfru	显示设备设置信息。
showdomainstatus	显示逻辑域状态。
showcodactivation	显示 CPU 激活密钥信息。
showcodusage	显示 CPU 内核资源的使用状况。
showcod	显示 CPU 激活注册和设置信息。
showpparmode	显示物理分区运行模式。
showdomainconfig	显示物理分区的逻辑域配置信息。

表 3-2 与操作相关的命令

命令的名称	功能概述
setupfru	设置内存镜像。
setpcl	设置物理分区配置信息。
addcodactivation	添加 CPU 激活密钥。
setcod	分配在物理分区 (PPAR) 中使用的 CPU 内核资源。
addboard	将系统板 (PSB) 添加到物理分区。
deleteboard	从物理分区删除 PSB。
setpparmode	设置物理分区运行模式。
setdomainconfig	指定物理分区的逻辑域配置信息。
poweron	启动物理分区。
poweroff	停止物理分区。
console	连接到控制域控制台。

本节通过使用示例详细介绍物理分区配置所使用的与显示相关的命令和与操作相关的命令。有关各个命令的选项、操作数、使用等详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

注 - XSCF shell 和 XSCF Web 用户界面提供给一些管理员，这些管理员需要管理员权限来配置物理分区。在多个物理分区管理员共享一个系统板的情况下，需要准备和有意识地规划以便更安全地操作系统。

3.1.1 检查物理分区配置信息

使用 `showpcl` 命令来显示物理分区配置信息 (PPAR 配置信息)。该命令以列表形式显示 PPAR 配置信息，如 PPAR-ID、在 PPAR 配置信息中登录的 PSB 编号以及物理分区的状态等。

在执行涉及到物理分区动态重新配置的操作之前，请使用 `showpcl` 命令来确定物理分区是否可以被动态地重新配置，并检查物理分区配置信息中目标系统板的登录状态。动态重新配置操作后，也可使用该命令来检查物理分区的状态和配置。

注 - 有下列任何权限的用户帐户可执行 `showpcl` 命令：
`platadm`, `platop`, `fieldeng`, `pparadm`, `pparmgr`, `pparop`

```
XSCF> showpcl [-v] -a
XSCF> showpcl [-v] -p ppar_id
```

- `-v` 选项
除了默认显示，此选项也将显示 `[Cfg-policy]`、`[No-Mem]` 和 `[No-IO]` 栏。

- -a 选项
此选项显示所有物理分区的 PPAR 配置信息。
- -p ppar_id 选项
指定物理分区编号。此选项显示指定物理分区的 PPAR 配置信息。

表 3-3 列出了 showpcl 命令的显示项目。

表 3-3 showpcl 的显示项目

显示项目	说明	含义
PPAR-ID	0 到 15 之间的整数	物理分区编号
LSB	0 到 15 之间的整数	逻辑系统板编号
PSB	xx-y (xx: 00 至 15 之间的整数; y: 固定为 0)	物理系统板编号表示一个模块化架构 (BB), 支持 LSB。
Status (PPAR 操作状态)	Powered Off	电源关闭状态
	Initialization Phase	正在进行 POST 操作的状态
	Initialization Complete	POST 完成状态
	Running	POST 处理完成后的运行状态
Cfg-policy (*1) (PPAR 配置信息的配置策略)	Hypervisor Aborted	虚拟机监控程序中止直到重置时的状态
	FRU	以部件 (如 CPU 或内存) 为单元执行降级
	PSB	以 PSB 为单元执行降级
	System	以 PPARs 为单元执行降级
No-Mem (*1) (PPAR 配置信息的 no-mem 标记)	True	不允许逻辑域使用安装在 LSB 中的内存。
	False	允许逻辑域使用安装在 LSB 中的内存。
No-IO (*1) (PPAR 配置信息的 no-io 标记)	True	不允许逻辑域使用安装在 LSB 中的 I/O 设备。
	False	允许逻辑域使用安装在 LSB 中的 I/O 设备。

*1 仅在指定 -v 选项时才显示此项目。

下面的示例显示物理分区 0 的 PPAR 配置信息。

```
XSCF> showpcl -p 0
PPAR-ID  LSB  PSB  Status
00
          00    00-0  Running
          04    01-0
          08    02-0
          12    03-0
```

如需显示配置策略、no-mem、no-io 等的详情, 请指定 -v 选项。下面的示例显示物理分区 0 的 PPAR 配置信息的详情。

```
XSCF> showpcl -v -p 0
PPAR-ID  LSB  PSB  Status  No-Mem  No-IO  Cfg-policy
00
          00    00-0  Running
```

System			
00	-		
01	-		
02	-		
03	-		
04	01-0	False	False
05	-		
06	-		
07	-		
08	02-0	True	False
09	-		
10	-		
11	-		
12	03-0	False	True
13	-		
14	-		
15	-		

3.1.2 检查物理分区状态

使用 `showpparstatus` 命令来显示物理分区状态。

注 - 有下列任何权限的用户帐户可执行 `showpparstatus` 命令：
`useradm`、`platadm`、`platop`、`fieldeng`、`pparadm`、`pparmgr`、`pparop`

```
XSCF> showpparstatus -a
XSCF> showpparstatus -p ppar_id
```

- `-a` 选项
此选项显示所有物理分区的状态。
- `-p ppar_id` 选项
指定物理分区编号。此选项显示指定物理分区的状态。

表 3-4 列出了 `showpparstatus` 命令的显示项目。

表 3-4 物理分区状态

显示项目	说明	含义
PPAR-ID	0 到 15 之间的整数	物理分区编号
PPAR Status (PPAR 状态)	Powered Off	电源关闭状态
	Initialization Phase	正在进行 POST 操作的状态
	Initialization Complete	POST 完成状态
	Running	POST 处理完成后的运行状态
	Hypervisor Aborted	虚拟机监控程序中止直到重置时的状态
	-	上面以外的状态 (PSB 未分配给 PPAR 的状态)

下面的示例显示操作物理分区 0 的状态。

```
XSCF> showpparstatus -p 0
PPAR-ID          PPAR Status
00               Running
```

3.1.3 检查系统板状态

使用 showboards 命令显示系统板 (PSB<BB>) 的分配给物理分区的状态。

执行物理分区动态重新配置操作之前，使用 showboards 命令如下所示：检查 PSB (BB) 的状态来确认是否可在其上执行动态重新配置操作，检查 PSB (BB) 所属物理分区的编号等。也可使用此命令来检查动态重新配置操作后 PSB (BB) 的状态。

注 - 有下列任何权限的用户帐户可执行 showboards 命令：
platadm, platop, fieldeng, pparadm, pparmgr, pparop

```
XSCF> showboards [-v] -a [-c sp]
XSCF> showboards [-v] -p ppar_id [-c sp]
XSCF> showboards [-v] psb
```

注 - showboards 命令仅对 BB 配置有效。

- -v 选项
此选项显示详细信息。指定此选项来检查预约状态以便添加或删除 PSB (BB)。
- -a 选项
此选项显示所有物理分区的信息。

- **-c sp** 选项
此选项只显示处于系统板池状态的 PSB (BB)。系统板池状态是系统板不属于任何物理分区的状态。
- **-p ppar_id** 选项
指定物理分区编号。此选项显示指定物理分区的信息。
- **psb**
指定 PSB (BB) 编号。

表 3-5 列出了 showboards 命令的显示项目。

显示项目	说明	含义
PSB	xx-y (xx: 00 至 15 之间的整数; y: 固定为 0)	表示一个模块化架构 (BB) 的物理系统板 (PSB) 编号 (其中 xx 是 BB-ID)
R (*1)	*	物理分区重新启动时, 将 PSB (BB) 整合到或释放当前物理分区配置以及配置变更时的状态
PPAR-ID	0 到 15 之间的整数	分配 PSB (BB) 的物理分区的编号
	SP	系统板池状态下的 PSB (BB)
	Other	物理分区配置信息被设置在具有用户权限的物理分区中并且属于不具有用户权限的物理分区的状态
LSB	00 到 15 之间的整数	物理分区所使用的逻辑系统板编号
Assignment (正在分配给 PPAR 的状态)	Unavailable	未分配给物理分区的 PSB (BB) 和处于以下任意一种状态 (包括已卸载 PSB (BB) 的状态): 系统板池、未诊断、正在诊断中、异常诊断中的 PSB (BB)
	Available	系统板池状态下的 PSB (BB), 诊断可正常结束
	Assigned	PSB (BB) 已分配
Pwr (系统板的电源开启状态)	n	电源关闭状态
	y	电源开启状态
Conn (整合到 PPAR 配置中的状态)	n	PSB (BB) 从物理分区配置中被释放的状态
	y	PSB (BB) 被整合到物理分区配置中的状态
Conf (Oracle Solaris 中的运行状态)	n	PSB (BB) 未在 Oracle Solaris 下运行的状态
	y	PSB (BB) 在 Oracle Solaris 下运行的状态
Test (诊断状态)	Unmount	已卸载或未定义的状态
	Unknown	未诊断的状态
	Testing	正在诊断中
	Passed	诊断正常结束的状态
	Failed	诊断检测出异常, 并且 PSB (BB) 不工作的状态

表 3-5 系统板状态 (续)

显示项目	说明	含义
Fault (已降级的状态)	Normal	正常状态
	Degraded	存在已降级部件的状态 (PSB (BB) 可以运行)
	Faulted	由于异常, PSB (BB) 不工作或由于通讯故障无法被管理的状态

*1 仅在指定 -v 选项时才显示此项目。

在以下示例中, 执行 `showboards` 命令以显示所有物理分区中所有 PSB (BB) 的分配状态。在 [PPAR-ID] 栏的下面, PSB 显示 "SP" 是处于系统板池状态。系统板池状态是 PSB 不属于任何物理分区的状态。

XSCF> **showboards -a**

PSB	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00 (00)	Assigned	y	y	y	Passed	Normal
01-0	00 (01)	Assigned	y	y	y	Passed	Normal
02-0	SP	Available	n	n	n	Passed	Normal
03-0	SP	Available	n	n	n	Passed	Normal

若需显示预约状态以添加或删除 PSB, 请指定 -v 选项。下面的示例显示所有已安装 PSB 的详细信息。

XSCF> **showboards -v -a**

PSB	R	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	*	00 (00)	Assigned	y	y	y	Passed	Normal
01-0		SP	Unavailable	n	n	n	Testing	Normal
02-0		Other	Assigned	y	y	n	Passed	Degraded
03-0		SP	Unavailable	n	n	n	Failed	Faulted

[R] 栏下显示的 "*" 代表预约状态。在上面示例中, 您可以看到 PSB 00-0 已做删除预约, 因为已处于分配状态。

3.1.4 检查设备设置信息

使用 `showfru` 命令显示设备设置信息。

注 - 有 `platadm` 或 `fieldeng` 权限的用户帐户可执行 `showfru` 命令。

XSCF> **showfru device location**

- `device`
将 `sb` (系统板 (PSB<BB>)) 或 `cpu` (PSB (BB) 上的 CPU) 指定为设备名称。

- location
如果设备名称是 **sb**, 请指定为 **BB-ID-0**。如果设备名称是 **cpu**, 请指定为 **BB-ID-0-CPU**。在此, **BB-ID** 的范围是 0 至 15, **CPU** 的范围是 0 到 3。

表 3-6 列出了 **showfru** 命令的显示项目。

表 3-6 showfru 命令显示项目

显示项目	说明	含义
Device (设备)	sb	系统板 (PSB)
	cpu	CPU
Location (所安装设备的位置)	如果 Device 是 sb BB-ID-0 (BB-ID: 0 到 15 之间的整数)	PSB 号
	如果 Device 是 cpu BB-ID-0-x (BB-ID: 0 至 15 之间的整数; x: 0 至 3 之间的整数)	CPU 编号
Memory Mirror Mode (内存镜像模式)	yes	内存镜像模式
	no	非内存镜像模式

下面的示例显示模块化架构 **BB-ID#01** 中 **PSB** 的设备设置信息。

```
XSCF> showfru sb 01-0
Device Location Memory Mirror Mode
sb      01-0
      cpu 01-0-0    yes
      cpu 01-0-1    yes
      cpu 01-0-2    yes
      cpu 01-0-3    yes
```

下面的示例显示模块化架构 **BB-ID#01** 中 **PSB** 上 **CPU#03** 的设置信息。

```
XSCF> showfru cpu 01-0-3
Device Location Memory Mirror Mode
sb      01-0
      cpu 01-0-3    yes
```

3.1.5 检查逻辑域状态

使用 **showdomainstatus** 命令显示逻辑域状态。

注 - 有下列任何权限的用户帐户可执行 **showdomainstatus** 命令:
useradm、platadm、platop、fieldeng、ppradm、pparmgr、pparop

```
XSCF> showdomainstatus -p ppar_id [-g domainname]
```

- -p ppar_id 选项
指定要显示的物理分区编号。
 - -g domainname 选项
指定要显示的逻辑域的名称。
- 显示项目如 表 3-7所示。

表 3-7 逻辑域状态

显示项目	说明	含义
Logical Domain Name	-	逻辑域名称
Status（当前逻辑域运行状态）	Host stopped	逻辑域停止状态
	Solaris booting	逻辑域 Oracle Solaris 正在启动的状态
	Solaris running	逻辑域 Oracle Solaris 正在运行的状态
	Solaris halting	逻辑域 Oracle Solaris 处于停止的状态
	Solaris suspended	逻辑域 Oracle Solaris 被挂起的状态
	Solaris powering down	逻辑域 Oracle Solaris 处于电源关闭的状态
	Solaris rebooting	逻辑域 Oracle Solaris 正在重新启动的状态
	Solaris panicking	逻辑域 Oracle Solaris 处于崩溃的状态
	Solaris debugging	停止的状态并带有逻辑域的 kmdb 提示（内核正在调试）
	OpenBoot initializing	逻辑域的 OpenBoot PROM 正在初始化的状态
	OpenBoot Running	逻辑域的 OpenBoot PROM 初始化已经完成或者停止并有 ok 提示的状态
	OpenBoot Primary Boot Loader	Oracle Solaris 正在逻辑域上加载的状态
	OpenBoot Running OS Boot	正在迁移到逻辑域上的 Oracle Solaris 的状态
	OS Started.No state support.	SUNW、soft-state-supported CIF 尚未实施，SUNW、set-trap-table CIF 正在运行的状态
	OpenBoot Running Host Halted	init 0 正在从逻辑域 Oracle Solaris 被执行的状态
	OpenBoot Exited	reset-all 正在从逻辑域的 ok 提示被执行的状态
	OpenBoot Host Received Break	enter 服务从逻辑域 Oracle Solaris 被调用的状态
	OpenBoot Failed	逻辑域的 OpenBoot PROM 初始化时发生错误
	Unknown	未知的状态下，未查找到与用户选项指定的逻辑域名称相匹配的逻辑域的未知状态（包括 add-spconfig 命令尚未在 Logical Domains Manager（逻辑域管理器）中执行的状态）
-		上面以外的状态（PSB 未分配给 PPAR 的状态）

下面的示例显示物理分区 0 上所有逻辑域的状态。

```
XSCF> showdomainstatus -p 0
Logical Domain Name      Status
primary                  Solaris running
guest00                  Solaris running
guest01                  Solaris booting
guest02                  Solaris powering down
guest03                  Solaris panicking
guest04                  Shutdown Started
guest05                  OpenBoot initializing
guest06                  OpenBoot Primary Boot Loader
```

3.1.6 检查 CPU 激活密钥信息

使用 `showcodactivation` 命令查看在 XSCF 中登录的 CPU 激活密钥的信息。

注 - 有 `platadm` 或 `platop` 权限的用户帐户可执行 `showcodactivation` 命令。

```
XSCF> showcodactivation
```

下面的示例显示 SPARC M10（2 组）中的四个 CPU 内核资源的 CPU 激活密钥。

```
XSCF> showcodactivation
Index   Description Count
-----
0 PROC          2
1 PROC          2
```

注 - 在 SPARC M10 中，您可以在 XSCF 中注册一组 2 个 CPU 激活，并启用 2 个 CPU 内核资源。在 SPARC M12 中，您可以在 XSCF 中注册一组 1 个 CPU 激活，并启用 1 个 CPU 内核资源。

3.1.7 检查 CPU 内核资源的使用状况

使用 `showcodusage` 命令显示 CPU 内核资源的使用状况。

注 - 有下列任何权限的用户帐户可执行 `showcodusage` 命令：
`platadm`, `platop`, `fieldeng`, `pparadm`, `pparmgr`, `pparop`

```
XSCF> showcodusage [-v] [-p {resource|ppar|all}]
```

- -v 选项
此选项显示详细信息。
- -p {resource|ppar|all} 选项
指定 "all" 来显示所有 CPU 内核资源的使用状况。指定 "resource" 来按照资源显示该信息。指定 "ppar" 来按照物理分区显示该信息。

下面的示例指定 showcodusage 命令中的 -p resource 选项，以便按照资源显示 CPU 内核的使用状况。
如此处所示，该系统已安装 16 个 CPU 内核资源，已注册 4 个 CPU 激活，4 个 CPU 内核资源正在使用中，当前未在使用的 CPU 激活的数量是 0。

```
XSCF> showcodusage -p resource
Resource In Use Installed CoD Permitted Status
-----
PROC          4          16          4 OK: 0 cores available

Note:
  Please confirm the value of the "In Use" by the ldm command of
  Oracle VM Server for SPARC.

  The XSCF may take up to 20 minutes to reflect the "In Use" of
  logical domains.
```

表 3-8 列出了此命令的显示项目的含义。

表 3-8 showcodusage -p resource 命令显示的详细信息

显示项目	含义
Resource	可用 CPU 内核资源（PROC: CPU，目前仅针对 CPU）
In Use	系统当前使用的 CPU 内核资源的数量（无法与虚拟机监控程序建立通讯时，返回 0。）
Installed	系统上安装的 CPU 内核资源的数量
CoD Permitted	系统上注册的 CPU 激活的数量
Status	正在使用的 CPU 内核资源的 CPU 激活可用性（OK：可用，VIOLATION：不可用）

下面的示例指定 SPARC M10 中 showcodusage 命令中的 -p ppar 选项来显示物理分区中 CPU 内核资源的使用状况。此示例表明物理分区 0 中已安装 32 个 CPU 内核资源，分配给该分区的 CPU 激活的数量为 4，当前有 3 个正在使用的 CPU 内核资源。此示例也表明物理分区 1 中已安装 32 个 CPU 内核资源，分配给该分区的 CPU 激活的数量为 4，当前有 4 个正在使用的 CPU 内核资源。从本示例可以知晓系统（在 XSCF 中注册但未分配给 PPAR）中当前未使用的 CPU 激活的数量是 12。

```
XSCF> showcodusage -p ppar
PPAR-ID/Resource In Use Installed Assigned
-----
0 - PROC          3          32          4 cores
1 - PROC          4          32          4 cores
2 - PROC          0           0          0 cores
3 - PROC          0           0          0 cores
```

```
4 - PROC          0          0          0 cores
5 - PROC          0          0          0 cores
6 - PROC          0          0          0 cores
7 - PROC          0          0          0 cores
8 - PROC          0          0          0 cores
9 - PROC          0          0          0 cores
10 - PROC         0          0          0 cores
11 - PROC         0          0          0 cores
12 - PROC         0          0          0 cores
13 - PROC         0          0          0 cores
14 - PROC         0          0          0 cores
15 - PROC         0          0          0 cores
Unused - PROC     0          0          12 cores

Note:
    Please confirm the value of the "In Use" by the ldm command of
    Oracle VM Server for SPARC.

    The XSCF may take up to 20 minutes to reflect the "In Use" of
    logical domains.
```

如果在 `showcodusage` 命令中指定 `-p ppar` 选项，可以确认 [表 3-9](#)中所示的项目。

表 3-9 `showcodusage -p ppar` 命令显示的项目

显示项目	含义
PPAR-ID/Resource	物理分区编号和 CPU 内核资源的类型。显示为 Unused 的 CPU 内核资源在物理分区内未使用。
In Use	物理分区中当前使用的 CPU 内核资源的数量
Installed	物理分区中安装的 CPU 内核资源的数量
Assigned	已分配到物理分区的 CPU 激活的数量

3.1.8 检查 CPU 激活注册和设置信息

使用 `showcod` 命令来检查 CPU 激活注册和设置信息。

注 - 有下列任何权限的用户帐户可执行 `showcod` 命令：
`platadm`, `platop`, `pparadm`, `pparmgr`, `pparop`

```
XSCF> showcod [-v]
XSCF> showcod [-v] -s cpu
XSCF> showcod [-v] -p ppar_id
```

- `-v` 选项
此选项显示详细信息。显示 CPU 激活的细目。
- `-s cpu` 选项
指定 CPU 编号。

- **-p ppar_id** 选项
指定物理分区编号。

下面的示例使用 **showcod** 命令来显示所有 CPU 激活注册和设置信息。

```
XSCF> showcod
PROC Permits installed: 6 cores
PROC Permits assigned for PPAR 0: 4
```

3.1.9 检查物理分区运行模式

使用 **showpparmode** 命令显示物理分区运行模式。

注 - 有下列任何权限的用户帐户可执行 **showpparmode** 命令：
platadm, fieldeng, pparadm

```
XSCF> showpparmode -p ppar_id [-v]
```

- **-p ppar_id** 选项
指定物理分区。
- **-v** 选项
此选项显示详细信息。也显示物理分区的以太网 (MAC) 地址。

表 3-10 列出了 **showpparmode** 命令的显示项目。

表 3-10 showpparmode 命令显示项目

显示项目	含义
Host-ID	主机 ID（如果主机 ID 未分配，则为连字号 "-"）(*1)
Diagnostic Level	POST 诊断级别（None[无]/normal[标准]/maximum[最大]）
Message Level	POST 诊断控制台信息的详细级别（None[无]/extract[摘要]/normal[标准]/maximum[最大]/debug[调试]）
Alive Check	启用/禁用 Alive Check
Watchdog Reaction	HOST Watchdog 超时时的 PPAR 反应（reset [重置]/Panic process [崩溃处理]/disable [禁用] (none)）
Break Signal	启用/禁用 Break 信号 (STOP-A)
Autoboot (Guest Domain)	PPAR 启动时启用/禁用来宾域自动启动
Elastic Mode	启用/禁用 CPU 和内存的低功耗运行
IOreconfigure	当开启 PPAR 的电源或重新启动时，根据总线配置启用/禁用 I/O 总线重新配置功能（enable（启用）/disable（禁用）/enable only at next start and automatically disable afterward（只在下次启动时启用，以后自动禁用）
PPAR DR (Current)	PPAR DR 功能的当前设置（如果 PPAR 电源未开启，则为连字号 "-"）(*2)

表 3-10 showpparmode 命令显示项目 (续)

显示项目	含义
PPAR DR (Next)	下次启动时 PPAR DR 功能的设置 (*2)
Ethernet Address	物理分区的以太网 (MAC) 地址 (*1)

*1 自动将不同的主机-ID 和以太网地址值分配给各个物理分区。
*2 默认情况下, XCP 2210 及更高版本启用 PPAR DR 功能。进行物理分区动态重新配置之前, 请使用 showpparmode 命令确认是否已启用 PPAR DR 功能。如果此功能禁用, 则对将要被动态重新配置的物理分区执行指定 -m ppar_dr=on 的 setpparmode 命令来启用 PPAR DR 功能。

3.1.10 显示物理分区的逻辑域配置信息

使用 showdomainconfig 命令显示逻辑域配置信息。

```
XSCF> showdomainconfig -p ppar_id
```

- -p ppar_id 选项
指定物理分区。

表 3-11 列出了 showdomainconfig 命令的显示项目。

表 3-11 showdomainconfig 命令的显示项目

显示项目	含义
Index	XSCF 上的配置信息控制编号
PPAR-ID	物理分区编号
Booting config (Current)	当前运行物理分区所使用的配置名称
Booting config (Next)	下次启动时物理分区所使用的配置名称
config_name	配置名称
domains	逻辑域配置中包括的逻辑域的数量 (*1)
date_created	配置信息创建的日期和时间

*1 使用 ldm add-spconfig 命令将逻辑域配置信息保存到 XSCF 时, 将显示绑定或活动状态下的逻辑域的数量。

下面的示例显示物理分区 0 的设置配置信息。

```
XSCF> showdomainconfig -p 0
PPAR-ID :0
Booting config
(Current) :ldm-set1
(Next) :ldm-set2
-----
Index :1
config_name :factory-default
domains :1
```



```

date_created:-
-----
Index :2
config_name :ldm-set1
domains :8
date_created:'2012-08-08 11:34:56'
-----
Index :3
config_name :ldm-set2
domains :20
date_created:'2012-08-09 12:43:56'

```

3.1.11 设置内存镜像

使用 `setupfru` 命令来设置内存镜像。

注 - 有 `platadm` 或 `fieldeng` 权限的用户帐户可执行 `setupfru` 命令。

- 对于 SPARC M12

指定 `-c mirror=yes` 可在 SPARC M12 中为内存创建镜像配置。

```
XSCF> setupfru [[-q] -{y|n}] -c function=mode device location
```

- 对于 SPARC M10

指定 `-m y` 可在 SPARC M10 中为内存创建镜像配置。不能使用 `-c mirror` 选项。

```
XSCF> setupfru [-m {y|n}] device location
```

- **-c function=mode**
指定是否将内存设置为镜像模式。若需设置为镜像模式，请指定 `-c mirror=yes`。否则，请指定 `-c mirror=no`。仅能在 SPARC M12 中使用。
- **-m {y|n}**
指定是否将内存设置为镜像模式。若需设置为镜像模式，请指定为 `y`。否则指定为 `n`。如果省略 `-m` 选项，则继承以前设置。
- **device**
指定要设置为镜像模式的范围。如果指定 `sb`，则该命令将设置反映到指定 PSB 下的所有 CPU。如果指定 `cpu`，则该命令将设置只反映到指定的 CPU。
- **location**
如果设备名称是 `sb`，请指定 PSB 编号。格式为 `BB-ID-0`。如果设备名称是 `cpu`，请指定 CPU。格式为 `BB-ID-0-CPU`。其中 `BB-ID` 是 0 到 15 之间的整数，`CPU` 是 0 到 3 之间的整数。

以下示例将 SPARC M12 中的 BB#01 (PSB#01) 下的所有 CPU 设置为内存镜像模式。

```
XSCF> setupfru -c mirror=yes sb 01-0

Notice:
- Logical domain config_name will be set to "factory-default".

Memory mirror mode setting will be changed, Continue? [y|n] :y
```

以下示例将 SPARC M12 中的 BB#02 下的所有 CPU1 设置为内存镜像模式。

```
XSCF> setupfru -c mirror=yes cpu 02-0-1

Notice:
- Logical domain config_name will be set to "factory-default".

Memory mirror mode setting will be changed, Continue? [y|n] :y
```

注 - 如果使用 SPARC M10, 请指定以下内容:

```
XSCF> setupfru -m y device location
```

3.1.12 设置物理分区配置信息

使用 `setpcl` 命令在物理分区配置信息 (PPAR 配置信息) 中登录系统板, 并设置配置策略、内存无效选项和 I/O 无效选项。

注 - 有 `platadm` 权限的用户帐户可执行 `setpcl` 命令。

```
XSCF> setpcl -p ppar_id -a lsb=psb [lsb=psb...]
XSCF> setpcl -p ppar_id -s policy=value
XSCF> setpcl -p ppar_id -s no-mem=value lsb [lsb...]
XSCF> setpcl -p ppar_id -s no-io=value lsb [lsb...]
```

- `-p ppar_id` 选项
指定要分配给物理分区的编号。此处指定的编号以后用作物理分区编号的参考。

注 - 给物理分区编号分配与系统上所装机箱的 BB-ID 编号相同的号码。如果将未安装的机箱的 BB-ID 指定为物理分区编号, 则该物理分区的上电将失败。

- `-a lsb=psb [lsb=psb...]` 选项
此选项将物理系统板 (PSB) 映射到逻辑系统板 (LSB)。对于 `lsb`, 指定逻辑系统板 (LSB) 编号, 对于 `psb`, 指定物理系统板 (PSB) 编号。
- `-s policy=value` 选项
此选项设置整个物理分区的配置策略。对于 `value`, 指定以下任意一个作为降级单元: `fru` (各部件)、`psb` (系统板单元 (BB 单元)) 或 `system` (整个物理分区)。

注 - 在相关物理分区电源关闭时设置配置策略选项。

- **-s {no-io|no-mem}=value**
指定 **true** 或 **false**。如果指定 **true**，将设置为无效，内存和 I/O 不可用。
- **lsb**
指定逻辑系统板编号（LSB 编号）。

注 -当相关 PSB (BB) 处于系统板池的状态，或当物理分区电源关闭时，设置此内存无效选项和 I/O 无效选项。系统板池状态是系统板不属于任何物理分区的状态。

注 -SPARC M12-2S/M10-4S 可设置内存无效选项和 I/O 无效选项。

在下面的示例中，物理系统板 (PSB) 00-0 和 01-0 分别映射到物理分区 0 的逻辑系统板 (LSB) 0 和 1。

```
XSCF> setpctl -p 0 -a 0=00-0 1=01-0
```

在下面的示例中，将配置策略设置为物理分区 0 和 1 的 "所有物理分区"。

```
XSCF> setpctl -p 0 -s policy=system
```

在下面的示例中，设置内存和 I/O 设备，以便物理分区 0 上的逻辑系统板 (LSB) 1 不使用这些内存和 I/O 设备。

```
XSCF> setpctl -p 0 -s no-mem=true 1
XSCF> setpctl -p 0 -s no-io=true 1
```

3.1.13 添加 CPU 激活密钥

使用 **addcodactivation** 命令添加 CPU 激活密钥。

注 - 有 **platadm** 权限的用户帐户可执行 **addcodactivation** 命令。

```
XSCF> addcodactivation {key-signature | -F url}
```

- **key-signature**
通过用双引号括起来 (") 指定 CPU 激活密钥。复制并粘贴包含 CPU 激活密钥文本文件 (XXXXXX_XX_XXX.TXT) 的内容。
- **-F url**
指定 CPU 激活密钥保存目标的 URL。

在下面的示例中，添加两个 CPU 内核的 CPU 激活密钥。

```
XSCF> addcodactivation "Product: SPARC M10-1
SequenceNumber: 10005
Cpu: noExpiration 2
Text-Signature-SHA256-RSA2048:
PSSrElBrse/r69AVSVFd38sT6AZm2bxeUDdPQHKbtxgvZPsrtYguqiNUieB+mTDC
:
:
b1GCKFx1RH27FdVHiB2H0A=="
Above Key will be added, Continue?[y|n]:y
```

在将 CPU 激活密钥添加到系统中时，CPU 内核资源尚未对在 Oracle Solaris 上使用准备就绪。

如需使 CPU 内核资源准备就绪，需参见 ["3.1.14 分配物理分区中使用的 CPU 内核资源"](#) 将其分配给物理分区。

3.1.14 分配物理分区中使用的 CPU 内核资源

根据注册在系统内的 CPU 激活，交互式执行 setcod 命令，将 CPU 内核资源分配给物理分区。

注 - 有 platadm 权限的用户帐户可执行 setcod 命令。

```
XSCF> setcod [-p ppar_id] -s cpu
PROC Permits installed: XX cores
PROC Permits assigned for PPAR 0 (X MAX)
[Permanent Xcores]
Permanent [X]: permits
PROC Permits assigned for PPAR 1 (X MAX)
[Permanent Xcores]
Permanent [X]: permits
: 省略
```

如果已应用 XCP 2260 或更高版本的 XSCF 固件，也可执行下面的命令实现此目的。

```
XSCF> setcod [[-q] -{y|n}] -p ppar_id -s cpu -c {set|add|del} permits
```

- -p ppar_id 选项
指定分配 CPU 内核资源的物理分区 ID。如果您未指定 permits 操作数，将启动交互式会话来分配 CPU 内核资源。
- -s cpu
指定 CPU。
- permits
指定 CPU 激活的数量。可以逐个内核地分配 CPU 激活。设置数量不能超过可用 CPU 激活的数量。如果您未指定 permits 操作数，将启动交互式会话来分配 CPU 内核资源。

- `-c {set|add|del}`
指定 `-c set` 可将 CPU 激活分配给物理分区，指定 `-c add` 可增加所分配的 CPU 激活的数量，指定 `-c del` 可删除一个所分配的 CPU 激活。

注 - 我们不建议使用以下规格的 `setcod` 命令：

XSCF> `setcod -p ppar_id -s cpu permits`

如需使用 XCP 2260 或更高版本的 XCP 固件执行 `setcod` 命令，需指定 `-c` 选项或使用交互式模式。对于 XCP 2250 或更早版本，需使用交互式模式。

原因如下：

- 当执行该命令时，不输出询问是否使用设置的信息进行变更的确认消息 (`[y|n]`)。
 - 当减少分配给正在运行的物理分区的 CPU 激活的数量时，不输出警告消息。
如果由于不正确的 `permits` 规格或一些其他错误没有足够的 CPU 激活，这可能导致系统关闭。
-

上限是由 `addcodactivation` 命令所添加的 CPU 激活数量。

注 - 若需使用 `-c set` 或不使用 `-c` 选项来指定 CPU 激活的数量，请在许可操作数中指定当前设置的数量加上添加的 CPU 激活数量或当前设置的数目减去已删除的 CPU 激活数量。请勿指定需要单独添加或删除的 CPU 激活数量。如果由于意外指定了需要单独添加或删除的 CPU 激活数量，CPU 激活的数量将降低，这可能导致系统停止。

以下是将四个 CPU 内核资源分配到物理分区 1 的示例。

```
XSCF> setcod -p 1 -s cpu -c set 4
PROC Permits assigned for PPAR 1 : 0 -> 4
PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y
Completed.
```

以下是将 CPU 内核资源交互式分配到物理分区的示例。

```
XSCF> setcod -s cpu
PROC Permits installed: 5 cores
PROC Permits assigned for PPAR 0 (5 MAX) [Permanent 2cores]
Permanent [2]:1
PROC Permits assigned for PPAR 1 (4 MAX) [Permanent 0cores]
Permanent [0]:4
PROC Permits assigned for PPAR 2 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 3 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 4 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 5 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 6 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 7 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 8 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 9 (0 MAX) [Permanent 0cores]
```

```
Permanent [0]:
PROC Permits assigned for PPAR 10 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 11 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 12 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 13 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 14 (0 MAX) [Permanent 0cores]
Permanent [0]:
PROC Permits assigned for PPAR 15 (0 MAX) [Permanent 0cores]
Permanent [0]:
```

以下示例将两个 CPU 内核资源添加到物理分区 0。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 10
XSCF> setcod -p 0 -s cpu -c add 2
PROC Permits assigned for PPAR 0 : 10 -> 12
PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y
Completed.
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 12
```

注 - 指定 `setcod -p 0 -s cpu -c set 12` 时，该命令得到相同的结果。

注 - XCP 2250 和更早版本的 XSCF 固件不支持 `-c add`、`-c delete` 和 `-c set` 选项。按如下所示指定 `setcod` 命令的选项可交互式添加和删除 CPU 内核资源。

XSCF> **setcod -s cpu**

3.1.15 添加系统板到模块化架构配置

使用 `addboard` 命令添加代表模块化架构模块 (BB) 的系统板 (PSB<BB>)，或预约一个系统板添加到物理分区。在执行此命令前，请使用相应的命令检查物理分区状态 (`showpparstatus` 命令) 和 PSB (BB) 状态 (`showboards` 命令)。

注 - 有 `platadm` 或 `pparadm` 权限的用户帐户可执行 `addboard` 命令。

注 - 要添加的 PSB (BB) 必须提前在相关物理分区的 PPAR 配置信息中登录。

```
XSCF> addboard [-c configure] [-m bind={resource|none}]
               [-m diag={off|min}] -p ppar-id psb [psb...]
XSCF> addboard -c assign -p ppar_id psb [psb...]
XSCF> addboard -c reserve -p ppar_id psb [psb...]
```

- **-c configure** 选项
此选项将 PSB (BB) 整合到指定物理分区。如果物理分区电源关闭或 Oracle Solaris 未在控制域中运行，将不整合系统板，并将出现错误。如果省略 **-c assign** 或 **-c reserve** 选项，则假定指定了 **-c configure** 选项。
- **-c assign** 选项
此选项将 PSB (BB) 分配给指定物理分区。分配的 PSB (BB) 为此指定的物理分区预约，不能从另一个物理分区中被分配。完成分配后，重新启动物理分区或执行指定 **-c configure** 选项的 **addboard** 命令来将 PSB (BB) 整合到物理分区。
- **-c reserve** 选项
此选项预约 PSB (BB) 到指定物理分区的分配。此操作与 "assign" 相同。
- **-m bind={resource|none}** 选项
此选项设置需要整合的 PSB (BB) 中添加的资源的自动分配功能。如果指定 **resource**，将启用添加的资源的自动分配功能。如果指定 **none**，将禁用添加的资源的自动分配功能，并且添加的 PSB (BB) 中的资源成为物理分区内的可用资源。默认值是 **resource**。

如果执行 **addboard** 命令之前，使用 **deleteboard** 命令已经删除了资源，您可以在执行 **deleteboard** 命令之前，启用自动分配功能来恢复已经存在的资源分配。但是，如果您在执行 **addboard** 命令之前更改了逻辑域配置信息，资源的分配将遵循逻辑域配置信息。

如果物理分区的逻辑域配置信息是出厂默认值，无论将此选项设置为哪个值，添加的资源都将被分配给控制域。

- **-m diag={off|min}** 选项
此选项设置当将 PSB (BB) 整合到物理分区时所执行的硬件诊断级别。如果指定 **off**，不执行硬件诊断。如果指定 **min**，则作为硬件诊断级别的标准设置。默认值是 **min**。
- **-p ppar_id** 选项
指定物理分区编号。
- **psb**
指定 PSB (BB) 编号。

注 - 使用 **-c configure** 选项在 PSB (BB) 整合中，完成 PSB 的硬件诊断后，目标 PSB (BB) 被整合到物理分区。因此，需要一些时间才可完成 **addboard** 命令。

注 - 当使用指定 **-c configure** 选项的 **addboard** 命令将 PSB (BB) 整合到物理分区时，您需要通过使用 **setpparmode** 命令启用目标物理分区的 PPAR DR 功能并重新启动该物理分区。指定 **-c configure** 选项的命令无法将 PSB (BB) 整合到禁用 PPAR DR 功能的物理分区。

注 - 如果在一个物理分区中发生 CPU 激活冲突，则无法执行 **addboard** 命令进行动态重新配置。请消除 CPU 激活冲突，然后执行 **addboard** 命令来执行动态重新配置。

注 - 如果由 **addboard** 命令执行的动态重新配置操作失败，目标 PSB (BB) 将无法恢复到此操作之前的状态。从 **addboard** 命令引起的错误信息输出和 Oracle Solaris 信息中查明原因，并采取适当措施。但是，依据所发生的错误，您可能需要关闭或重新启动物理分区。

注 -当您执行 `addboard` 命令进行 PSB (BB) 更换时, 如果省略 `-m` 选项或将 `-m` 选项指定为 `bind=resource`, 资源分配可能无法恢复到其执行 `deleteboard` 命令之前的状态。如果更换前 PSB (BB) 资源 (如 CPU 和内存) 的量与更换后的量不同, 将无法将资源分配恢复到其初始状态。无法恢复到其初始状态的资源将成为可用资源。此种情况下, 使用 `ldm` 命令来再次将资源分配给逻辑域。

在下面的示例中, 物理系统板 (PSB) 00-0、01-0、02-0 和 03-0 被分配到物理分区 0。-y 选项用于对命令的所有确认问题回答 "y"。

```
XSCF> addboard -y -c assign -p 0 00-0 01-0 02-0 03-0
```

3.1.16 删除模块化架构配置中的系统板

使用 `deleteboard` 命令删除表示模块化架构 (BB) 的系统板 (PSB<BB>), 或预约一个系统板从物理分区的删除。删除的 PSB (BB) 被分配给不属于任何物理分区的系统板池。

在执行此命令前, 请使用相应的命令检查物理分区状态 (`showpparstatus` 命令)、PSB (BB) 状态 (`showboards` 命令) 和 PSB (BB) 上的设备使用状况 (`showfru` 命令)。

注 - 有 `platadm` 或 `pparadm` 权限的用户帐户可执行 `deleteboard` 命令。

```
XSCF> deleteboard [-f] [-c disconnect]
                        [-m unbind={none|resource|shutdown}]
                        psb [psb...]
XSCF> deleteboard [-f] -c unassign psb [psb...]
XSCF> deleteboard [-f] -c reserve psb [psb...]
```

- **-c disconnect** 选项
此选项将正在物理分区中运行的 PSB (BB) 从物理分区配置中释放, 并使其进入已分配的状态。若需将 PSB (BB) 再次整合到物理分区, 在系统板仍然分配给物理分区配置时, 请重新启动物理分区或执行 `addboard` 命令。如果省略 `-c unassign` 或 `-c reserve` 选项, 则假定指定了 `-c disconnect` 选项。
- **-c unassign** 选项
此选项从物理分区中取消 PSB (BB) 分配并使其进入系统板池状态。可以将处于系统板池状态的 PSB (BB) 整合到或分配给另一个物理分区。
- **-c reserve** 选项
此选项预约物理分区中 PSB (BB) 的取消分配。在取消分配被预约之后, 基于 PPAR 配置信息, 在关闭指定物理分区的电源时, PSB (BB) 被取消分配, 并进入系统板池状态。
- **-m unbind={none|resource|shutdown}** 选项
此选项设置当使用需要释放的 PSB (BB) 上的资源的逻辑域被移动到未释放的 PSB (BB) 并且目标没有足够资源时所采取的措施。

设置	说明
none	目标无法确保资源。如果目标没有足够的资源，执行 <code>deleteboard</code> 命令将失败。 如果物理分区的逻辑域配置信息是出厂默认值，将无法指定 <code>-m unbind=none</code> 选项。 如果出现错误，则必须使用 Oracle VM Server for SPARC 的虚拟 DR 功能，从逻辑域中移除 CPU 内核或内存。
resource	通过减少需要移动资源的逻辑域的资源，或减少另一个逻辑域的资源来确保目标的资源。但是，此设置不能关闭逻辑域来确保目标的资源。此外，在 Oracle VM Server for SPARC 3.2 或更高版本的 XCP 2240 或更高版本上，将自动删除物理 I/O。
shutdown	通过减少需要移动资源的逻辑域的资源，或减少另一个逻辑域的资源来确保目标的资源。另外，此设置关闭逻辑域来确保目标的资源。

- `psb`
指定系统板编号。
- `-f` 选项
此选项强制释放 PSB (BB)。

注 -如果 `deleteboard` 命令具有指定的 `-f` 选项来强制从物理分区中删除 PSB (BB)，在 CPU 绑定的过程中或访问设备的过程中可能发生严重问题。因此，我们建议在正常运行时请勿使用 `-f` 选项。如果指定了 `-f` 选项，请确保在执行 `deleteboard` 命令之前和之后，检查物理分区和任务处理的状态。

注 -在使用指定有 `-c disconnect` 或 `-c unassign` 选项的 `deleteboard` 命令以便在物理分区正在运行时删除 PSB (BB) 之前，请执行以下操作：使用 `setpparmode` 命令为从中删除 PSB (BB) 的物理分区启用 PPAR DR 功能，并重新启动物理分区。您无法在物理分区正在运行时，通过指定 `-c disconnect` 或 `-c unassign` 选项，从禁用 PPAR DR 功能的物理分区中删除 PSB (BB)。

注 -如果在一个物理分区中发生 CPU 激活冲突，则无法执行 `deleteboard` 命令进行动态重新配置。请消除 CPU 激活冲突，然后执行 `deleteboard` 命令来执行动态重新配置。

注 -如果由 `deleteboard` 命令执行的动态重新配置操作失败，目标 PSB (BB) 将无法恢复到此操作之前的状态。从 `deleteboard` 命令引起的错误信息输出和 Oracle Solaris 信息中查明原因，并采取适当措施。但是，依据所发生的错误，您可能需要关闭或重新启动物理分区。

在下面的示例中，将物理系统板 (PSB) 00-0、01-0、02-0 和 03-0 设置为处于系统板池状态。

```
XSCF> deleteboard -c unassign 00-0 01-0 02-0 03-0
```

在下面的示例中，预约 PSB 00-0、01-0、02-0 和 03-0 的释放。

```
XSCF> deleteboard -c reserve 00-0 01-0 02-0 03-0
```

3.1.17 设置物理分区运行模式

使用 `setpparmode` 命令设置物理分区运行模式。

注 - 若需设置诊断级别、信息级别和来宾域的自动启动功能，必须由具有 `fieldeng` 权限的用户帐户执行 `setpparmode` 命令。
对于 `Active Check`、`Break` 信号、`HOST Watchdog` 超时时的反应和来宾域的自动启动功能、省电功能、`IO` 总线重新配置功能以及 `PPAR DR` 功能，必须由具有 `platadm` 或 `pparadm` 权限的用户帐户执行此命令。

```
XSCF> setpparmode -p ppar_id -m function=mode
```

- `-p ppar_id` 选项
指定物理分区。
- `-m function=mode` 选项
指定功能和运行模式。有关详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

表 3-12 列出了可设置的运行模式。

表 3-12 在 PPAR 可被设置的模式

功能名称	解释	可设置模式	反应
诊断级别	设置开机时自诊断测试 (POST) 的诊断级别。	None（无）/Normal（标准）（默认值）/Maximum（最大）	立即 (*1)
信息级别	设置 POST 诊断控制台信息的详细级别。	None（无）/Extract（摘要）/Normal（标准）（默认值）/Maximum（最大）/Debug（调试）	立即 (*1)
Alive Check	设置 Alive Check 的启用/禁用。	Enable（启用）（默认值）/Disable（禁用）	在下次启动时 (*2)
HOST Watchdog 超时时的反应	设置监视程序超时各逻辑域如何反应。	None（无）/Panic process（崩溃处理）/PPAR reset（PPAR 重置）（默认值）	立即 (*1)
Break 信号 (STOP-A) 限制	设置 Break 信号限制的启用/禁用。	Enable（启用）（默认值）/Disable（禁用）	立即 (*1)
来宾域的自动启动功能	设置在物理分区开启电源时，来宾域自动启动的启用/禁用。	Enable（启用）（默认值）/Disable（禁用）	下次启动时 (*2)
省电功能	设置 CPU 和内存低功耗运行的启用/禁用。	Enable（启用）/Disable（禁用）（默认值）	立即 (*1)

表 3-12 在 PPAR 可被设置的模式 (续)

功能名称	解释	可设置模式	反应
I/O 总线重新配置 (ioreconfigure) 功能	物理分区关闭电源然后再次开启电源（重置）时，根据总线配置选择 I/O 总线重新配置功能的操作模式。	Enable（启用）/Disable（禁用）（默认值）/Enable only at the next start and automatically disabled for the subsequent starts（只在下次启动时启用，以后自动禁用）	立即 (*1)
PPAR DR 功能	设置在当前正在运行的物理分区的配置中整合和释放 (PSB<BB>) 的功能。	Enable（启用）/Disable（禁用）(*3)	下次启动时 (*2)

*1 该设置将立即反映。
*2 如需反应此设置，您需要开启物理分区电源，或将其关闭后再重新开启。
*3 在早于 XCP 2210 的版本中，默认禁用 PPAR DR 功能，在 XCP 2210 及更高版本中，默认启用该功能。

注 - SPARC M12-1/M10-1 不支持 I/O 总线重新配置功能。

注 - SPARC M12-1/M12-2/M10-1/M10-4 不支持 PPAR DR 功能。

注 - 使用 `ldm set-variable` 命令，设置是否自动与来宾域一起启动 Oracle Solaris。默认情况下，Oracle Solaris 自动与来宾域一起启动。如需自动启动一个域，则需要进行域激活并保存逻辑域配置信息。有关详情，请参见您所使用版本的《Oracle VM Server for SPARC Administration Guide》。

模式开关操作

根据操作面板上模式开关的设置，物理分区操作如下。

- Locked
按照 `setpparmode` 命令指定的操作模式操作。
- Service
表 3-13 列出了模式开关设置为 Service 时的操作。

表 3-13 模式开关设置为 Service 时的操作

功能名称	操作模式
诊断级别	按照 <code>setpparmode</code> 命令指定的操作模式操作。
信息级别	按照 <code>setpparmode</code> 命令指定的操作模式操作。
Alive Check	无论操作模式处于哪种状态，都将禁用。
HOST Watchdog 超时时的反应	按照 <code>setpparmode</code> 命令指定的操作模式操作。
Break 信号 (STOP-A) 限制	无论操作模式处于哪种状态，都传输 Break 信号。
来宾域的自动启动功能	按照 <code>setpparmode</code> 命令指定的操作模式操作。

表 3-13 模式开关设置为 Service 时的操作 (续)

功能名称	操作模式
省电功能	按照 <code>setpparmode</code> 命令指定的操作模式操作。
I/O 总线重新配置功能	按照 <code>setpparmode</code> 命令指定的操作模式操作。
PPAR DR 功能	按照 <code>setpparmode</code> 命令指定的操作模式操作。

3.1.18 指定物理分区的逻辑域配置信息

使用 `setdomainconfig` 命令指定物理分区配置信息。

```
XSCF> setdomainconfig -p ppar_id -i index
```

- `-p ppar_id` 选项
指定物理分区。
- `-i index` 选项
指定配置信息控制编号。您可以使用 `showdomainconfig` 命令检查控制号。您可以指定 1 到 8 之间的整数。

注 - 只是通过使用 Oracle Solaris 的 `shutdown` 命令重新启动控制域，在 XSCF 中保存的逻辑域配置不会生效。务必使用 XSCF 的 `poweroff` 和 `poweron` 命令重新启动控制域。

3.1.19 启动物理分区

使用 `poweron` 命令来开启物理分区的电源。

注 - 有下列任何权限的用户帐户可执行 `poweron` 命令：
`platadm`, `fieldeng`, `pparadm`, `pparmgr`

```
XSCF> poweron -p ppar_id
```

- `-p ppar_id` 选项
指定物理分区编号。

3.1.20 停止物理分区

使用 `poweroff` 命令来停止物理分区。

```
XSCF> poweroff -p ppar_id
```

- -p ppar_id 选项
指定物理分区。

3.1.21 连接到控制域控制台

使用 console 命令来连接到物理分区的控制域控制台。

注 - 有下列任何权限的用户帐户可执行 console 命令：
platadm, platop, fieldeng, pparadm, pparmgr, pparop

```
XSCF> console -p ppar_id
```

- -p ppar_id 选项
指定物理分区编号。

3.2 与逻辑域配置有关的操作和命令

配置物理分区后，需配置各逻辑域。使用 Oracle VM Server for SPARC 的管理软件：Logical Domains Manager（逻辑域管理器）的 ldm 命令来配置逻辑域。本节通过使用示例介绍逻辑域配置所使用的与显示相关的命令和与操作相关的命令。有关各命令的选项、操作数、使用等的详细信息，请参见 Oracle VM Server for SPARC 和所使用版本的 Oracle Solaris 的手册。

有两种用于配置逻辑域的命令：与显示相关的命令和与操作相关的命令。

表 3-14 与显示相关的命令

命令的名称	功能概述
svcs	检查 Logical Domains Manager（逻辑域管理器）是否已经启动。
ldm list-services	显示服务。
ldm list-permits	显示可被分配的虚拟 CPU 的数量。
ldm list-bindings	显示资源的分配状况。
ldm list-devices	显示资源的使用状况。
ldm list-rsrc-group	
ldm list-socket	
ldm list-domain	显示逻辑域状态。
ldm list-spconfig	显示逻辑域配置信息。
ldm list-io	显示 I/O 设备的使用状况。

表 3-15 与操作相关的命令

命令的名称	功能概述
ldm start-reconf	启动延迟重新配置模式。
ldm add-vconscon	设置默认服务。
ldm add-vdiskserver	
ldm add-vswitch	
ldm add-vcpu	配置虚拟 CPU。
ldm set-vcpu	
ldm remove-vcpu	
ldm add-core	
ldm set-core	
ldm remove-core	
ldm grow-socket	
ldm shrink-socket	
ldm add-memory	配置虚拟内存。
ldm set-memory	
ldm remove-memory	
ldm grow-socket	
ldm shrink-socket	
ldm add-spconfig	设置逻辑域配置信息。
ldm set-spconfig	
ldm remove-spconfig	
ldm add-domain	创建逻辑域。
ldm add-io	配置 I/O 设备。
ldm remove-io	
ldm create-vf	创建或销毁 SR-IOV 的虚拟功能。
ldm destroy-vf	
ldm add-vnet	配置虚拟网络设备。
ldm remove-vnet	
ldm add-vdiskserverdevice	配置虚拟磁盘服务器。
ldm remove-vdiskserverdevice	
ldm add-vdisk	配置虚拟磁盘。
ldm remove-vdisk	
ldm set-socket	设置 CPU 插槽约束条件。
ldm set-vconsole	配置虚拟控制台。
ldm set-variable	配置启动设备。
ldm bind-domain	绑定资源。
ldm start-domain	启动来宾域。
ldm set-domain	配置关闭组。
devfsadm	重新配置设备。
svccfg	配置恢复模式。

3.2.1 确认逻辑域管理器正在运行

使用 Oracle Solaris 的 `svcs` 命令来确认 Logical Domains Manager（逻辑域管理器）是否正在运行。

```
# svcs service
```

- `service`
指定服务名称。

下面的示例检查 Logical Domains Manager（逻辑域管理器）是否正在运行。Logical Domains Manager（逻辑域管理器）的服务名称是 `ldmd`。如果 `[STATE]` 栏显示 `"online"`，则是正在运行。

```
# svcs ldmd
STATE STIME FMRI
online 16:25:31 svc:/ldoms/ldmd:default
```

注 - Logical Domains Manager 是 Oracle VM Server for SPARC 中包括的软件。有关 `svcs` (1) 命令的详情，请参见 Oracle Solaris 的参考手册。

注 - 如果 Logical Domains Manager（逻辑域管理器）未启动，请执行 `svcadm` 命令可将其启动。有关 `svcadm` 命令的详情，请参见 Oracle Solaris 的参考手册。

3.2.2 检查服务

使用 `ldm list-services` 命令来确认所设置的服务。

```
# ldm list-services ldom
```

- `ldom`
指定逻辑域名称。控制域名称是 `"primary"`。

在下面的示例中，`"primary-vds0"` 显示为虚拟磁盘服务器，`"primary-vcc0"` 显示为虚拟控制台终端收集 and 分配设备服务，`"primary-vsw0"` 显示为虚拟交换机服务。

```
# ldm list-services primary
VDS
  NAME          VOLUME          OPTIONS          DEVICE
  primary-vds0
VCC
  NAME          PORT-RANGE
  primary-vcc0  5000-5100
VSW
  NAME          MAC          NET-DEV  DEVICE  MODE
```

3.2.3 基于 CPU 激活，检查可分配的虚拟 CPU 的数量

使用 `ldm list-permits` 命令检查控制域中可以分配的虚拟 CPU 内核数量。

```
# ldm list-permits
```

3.2.4 检查资源的分配状况

使用 `ldm list-bindings` 命令检查分配状况。

```
# ldm list-bindings ldom
```

- `ldom`
指定逻辑域。如果是控制域，"primary" 是固定设置。

3.2.5 检查资源的使用状况

使用 `ldm list-devices` 命令检查物理分区中资源的使用状况。

```
# ldm list-devices [-S] [-a] [cpu]
```

- `-S` 选项
此选项显示设备的状态。
- `-a` 选项
此选项显示已绑定或未绑定的所有资源的使用状况。
- `cpu`
CPU 线程和物理 CPU 资源在列表中显示。

表 3-16 列出了此命令的显示项目的含义。

表 3-16 ldm list-devices 命令的显示项目

显示项目	说明	含义
PID	0 或更大的整数	物理 CPU 编号
%FREE	0 到 100 间的数字值	未使用百分比
PM	yes	管理 CPU 的电源
	no	CPU 电源开启
	---	未分配给逻辑域的 CPU
STATUS (*1)	ok	CPU 状态正常
	fail	CPU 发生故障

*1 指定 -S 选项时显示 STATUS。

下面的示例显示八个虚拟 CPU 的状态。

```
# ldm list-devices -S -a cpu
VCPU
  PID %FREE PM STATUS
  0   0   no  ok
  1   0   yes ok
  2   0   yes ok
  3   0   yes ok
  4  100   --- ok
  5  100   --- ok
  6  100   --- ok
  7  100   --- ok
```

在下面的示例中，指定 -a 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。

```
# ldm list-devices -a
CORE
  ID      %FREE  CPUSSET
  0        0      (0, 1)
  4        0      (8, 9)
  8        0     (16, 17)
(省略)
VCPU
  PID      %FREE  PM
  0        0      no
  1        0      no
  8        0      no
  9        0      no
(省略)
MEMORY
  PA              SIZE      BOUND
  0x7c0000000000  8G      guest2
  0x7c0200000000  24G
  0x7e0000800000  1G      _sys_
  0x7e0040800000  384M   _sys_
(省略)
IO
```

DEVICE	PSEUDONYM	BOUND
OPTIONS		
pci@8000	PCIE0	yes
pci@8100	PCIE1	yes
pci@8200	PCIE2	yes
pci@8300	PCIE3	yes
PCIEDEV		
pci@8000/pci@4/pci@0/pci@9	/BB0/CMUL/NET0	yes
pci@8000/pci@4/pci@0/pci@a	/BB0/CMUL/NET2	yes
pci@8000/pci@4/pci@0/pci@0	/BB0/CMUL/SASHBA	yes
pci@8100/pci@4/pci@0/pci@0	/BB0/PCIO	yes
(省略)		

3.2.6 检查各资源组的使用状况

使用 `ldm list-rsrc-group` 命令检查物理分区中各资源组的资源使用状况。资源组是基于处理器内核、内存以及 I/O 总线之间潜在的物理关系而配置。对于 Oracle VM Server for SPARC 3.3 和更高版本，SPARC M12/M10 系统支持此子命令，并显示机箱或 BB-ID 作为资源组。
有关资源组的详细信息，请参见《Oracle VM Server for SPARC Administration Guide》中的“Managing Resources”。

```
# ldm list-rsrc-group [-a] [-d ldom] [-l] [-o core|memory|io] [-p] [resource-group]
```

- `-a`
此选项显示各资源组的所有资源。也包括未绑定到任何逻辑域的资源。
- `-d`
此选项仅显示指定逻辑域的信息。
- `-l`
此选项显示各资源组的详细信息。
- `-o`
此选项仅显示指定资源的信息。
- `-p`
此选项以可解析的格式输出结果。
- `resource-group`
指定需要列表的资源组。

下面的示例列出了有两个系统板的物理分区中各资源组的所有资源。

# ldm list-rsrc-group			
NAME	CORE	MEMORY	IO
/BB0	64	262136M	8
/BB1	64	262136M	8

3.2.7 检查各 CPU 插槽的资源使用状况

使用 `ldm list-socket` 命令检查物理分区中各 CPU 插槽的资源使用状况。Oracle VM Server for SPARC 3.2 和更高版本支持此子命令。

```
# ldm list-socket [-l] [-p] [--free] [-o format] [socket_id=id[,id[,...]]]
[ldom...]
```

- `-l`
此选项显示物理 CPU 集和内核 ID。
- `-p`
此选项以可解析的格式输出结果。
- `--free`
此选项仅显示可用资源。
- `-o`
此选项将结果输出限制到一个或多个子集。有关子集的详情，请参见您所使用版本的《Oracle VM Server for SPARC Reference Manual》。
- `socket_id`
指定需要列出其信息的一个或多个 CPU 插槽。
- `ldom`
指定需要列出其信息的一个或多个逻辑域。

下面的示例列出了各 CPU 插槽的所有资源的使用状况。

```
# ldm list-socket
CONSTRAINTS
SOCKET
  TENANT      VCPUS    CORES    SOCKET_ID  GROUP
  primary     32       16       0          /BB0
  primary     32       16       1          /BB0
  primary     32       16       2          /BB0
  primary     32       16       3          /BB0
  primary     32       16       4          /BB1
  primary     32       16       5          /BB1
  primary     32       16       6          /BB1
  primary     32       16       7          /BB1
MEMORY
  PA          SIZE      SOCKET_ID  BOUND
  0x700000000000 64G      7          primary
  0x720000000000 64G      6          primary
  0x740000000000 64G      5          primary
  0x760050000000 31488M   4          primary
  0x780000000000 64G      3          primary
  0x7a0000000000 64G      2          primary
  0x7c0000000000 64G      1          primary
  0x7e0080000000 62G      0          primary
IO
  NAME      TYPE      BUS      SOCKET_ID  BOUND
```

PCIE0	BUS	PCIE0	0	primary
PCIE1	BUS	PCIE1	0	primary
PCIE2	BUS	PCIE2	1	primary
PCIE3	BUS	PCIE3	1	primary
PCIE4	BUS	PCIE4	2	primary
PCIE5	BUS	PCIE5	2	primary
PCIE6	BUS	PCIE6	3	primary
PCIE7	BUS	PCIE7	3	primary
PCIE8	BUS	PCIE8	4	primary
PCIE9	BUS	PCIE9	4	primary
PCIE10	BUS	PCIE10	5	primary
PCIE11	BUS	PCIE11	5	primary
PCIE12	BUS	PCIE12	6	primary
PCIE13	BUS	PCIE13	6	primary
PCIE14	BUS	PCIE14	7	primary
PCIE15	BUS	PCIE15	7	primary

3.2.8 检查逻辑域状态

使用 `ldm list-domain` 命令来检查物理分区中正在运行的逻辑域的状态和关闭组设置。

```
# ldm list-domain [-o format] [ldom ...]
```

- **format**
指定命令的格式。指定 **domain** 来检查关闭组设置。
- **ldom**
指定目标逻辑域的名称。

下面的示例表明控制域和三个来宾域正在运行。

```
# ldm list-domain
NAME                STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary             active    -n-cv-   UART    14     8G      64%   2h 54m
guest0              active    -n----   5000    16     8G      42%   2h 54m
guest1              active    -n----   5001    16     8G      11%   2h 54m
guest2              active    -n----   5002    16     8G      7.3%  2h 54m
```

下面的示例设置逻辑域 `ldom1` 的关闭组 7，并检查设置。最后一行显示关闭组为 7。

```
# ldm set-domain shutdown-group=7 ldom1
# ldm list-domain -o domain ldom1
NAME                STATE      FLAGS    UTIL
ldom1               active     -n----   0.1%

CONTROL
  failure-policy=ignore
  extended-mapin-space=on
  shutdown-group=7
```

关闭组设置包括在逻辑域配置信息中。因此，当变更关闭组设置时，您需要通过使用 `ldm add-spconfig` 命令将其内容保存到 XSCF 中。

3.2.9 显示逻辑域配置信息

从控制域中使用 `ldm list-spconfig` 命令来显示保存到 XSCF 中的逻辑域配置信息。

```
# ldm list-spconfig
```

下面的示例从控制域中执行了 `ldm list-spconfig` 命令来显示逻辑域配置信息。

```
# ldm list-spconfig
ldm-set1 [current]
ldm-set2
factory-default
```

3.2.10 检查 I/O 设备的使用状况

使用 `ldm list-io` 命令检查在物理分区内逻辑域中配置的 I/O 设备。

```
# ldm list-io [-l] [-p] [bus | device | pf-name]
# ldm list-io -d pf-name
```

- `-l`
显示根合成体和 PCIe 直接 I/O 设备之间的关系等详细信息。
- `-p`
以可解析的格式，如以脚本的形式，显示信息。
- `-d pf-name`
显示关于指定的物理功能信息的列表。
- `bus`、`device` 和 `pf-name`
这些分别是 PCIe 总线、直接 I/O 可分配设备和 PCIe SR-IOV 物理功能。

下面的示例显示逻辑域中 I/O 设备的使用状况。
对于支持 SR-IOV 功能的 PCIe 设备，[TYPE]（类型）栏显示“PF”。
对于创建或使用 SR-IOV 的虚拟功能的设备，[TYPE]（类型）栏显示“VF”。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	primary	IOV

PCIE6	BUS	PCIE6	primary	IOV
PCIE7	BUS	PCIE7	primary	IOV
(省略)				
/BB0/PCI6	PCIE	PCIE6	primary	EMP
/BB0/PCI9	PCIE	PCIE7	primary	EMP
/BB0/PCI10	PCIE	PCIE7	primary	EMP
/BB0/PCI0/IOVNET.PF0	PF	PCIE1	primary	
/BB0/PCI0/IOVNET.PF1	PF	PCIE1	primary	
/BB0/CMUL/NET0/IOVNET.PF0	PF	PCIE0	primary	
/BB0/CMUL/NET0/IOVNET.PF1	PF	PCIE0	primary	
/BB0/CMUL/NET2/IOVNET.PF1	PF	PCIE4	primary	
/BB0/CMUL/NET2/IOVNET.PF0	PF	PCIE4	primary	
/BB0/PCI0/IOVNET.PF0.VF0	VF	PCIE1	iodom00	
/BB0/PCI0/IOVNET.PF0.VF1	VF	PCIE1	iodom01	
/BB0/PCI0/IOVNET.PF0.VF2	VF	PCIE1	iodom02	

如需确认根合成体和 PCIe 直接 I/O 设备之间关系等具体内容，请执行指定 -l 选项的该命令。

下面的示例显示逻辑域中 I/O 设备的使用状况。

- 分配 PCIe 卡的逻辑域
在维护目标 PCIe 卡插槽（其 [NAME]（名称）栏显示“/BBX/PCIX”的行）的 [DOMAIN]（域）栏中显示的名称表示 PCIe 卡被分配的逻辑域。
- 分配 PCIe 卡所属的 PCIe 根合成体的逻辑域
在上例已确认的 [BUS]（总线）栏中显示的“PCIEx”，是 PCIe 卡所属 PCIe 根合成体的名称。[DOMAIN]（域）栏显示同一 PCIe 根合成体（在其 [NAME]（名称）栏显示“PCIEx”的行上）的名称。此名称是一个逻辑域名。该 PCIe 卡所属 PCIe 根合成体被分配给逻辑域。

# ldm list-io -l				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
[pci@8000]				
PCIE1	BUS	PCIE1	primary	IOV
[pci@8100]				
PCIE2	BUS	PCIE2	primary	IOV
[pci@8200]				
PCIE3	BUS	PCIE3	rootdom	IOV
[pci@8300]				
PCIE4	BUS	PCIE4	primary	IOV
[pci@8400]				
PCIE5	BUS	PCIE5	primary	IOV
[pci@8500]				
PCIE6	BUS	PCIE6	rootdom	IOV
[pci@8600]				
PCIE7	BUS	PCIE7	primary	IOV
[pci@8700]				
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
[pci@8000/pci@4/pci@0/pci@9]				
network@0				
network@0,1				
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC

```
[pci@8000/pci@4/pci@0/pci@0]
scsi@0/iport@f/disk@w50000393a82368b2,0
scsi@0/iport@f/smp@w500000e0e06d03bf
scsi@0/iport@f/enclosure@w500000e0e06d03bd,0
scsi@0/iport@v0
/BB0/PCI0
[pci@8100/pci@4/pci@0/pci@0]
/BB0/PCI3
[pci@8200/pci@4/pci@0/pci@0]
/BB0/PCI4
[pci@8200/pci@4/pci@0/pci@8]
/BB0/PCI7
[pci@8300/pci@4/pci@0/pci@0]
/BB0/PCI8
[pci@8300/pci@4/pci@0/pci@8]
/BB0/CMUL/NET2
[pci@8400/pci@4/pci@0/pci@a]
network@0
network@0,1
```

3.2.11 启动延迟重新配置模式

执行重新配置已分配给控制域或根域的硬件资源操作之前，请切换到延迟重新配置模式。这是因为在采用动态重新配置进行重新配置期间，虚拟内存的重新配置可能需要一定时间。使用 `ldm start-reconf` 命令启动延迟重新配置模式。

```
# ldm start-reconf ld0m
```

下面的示例将 `primary`（控制域）切换到延迟重新配置模式。

```
# ldm start-reconf primary
```

使用 `ldm list-binding` 命令检查此模式是否是延迟重新配置模式。下面的示例显示 `ldm list-bindings` 命令的执行结果。在结果中，从 [FLAGS] 栏 "-ndcv-" 开始的第三个字符为 "d"，表示延迟重新配置模式。

```
# ldm list-bindings primary
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-ndcv-	SP	8	4G	0.1%	2h 16m

切换到延迟重新配置模式后，您将无法使用 `ldm` 命令来操作来宾域。下面的示例显示造成来宾域停止的故障。

```
# ldm stop-domain -a
-----
Notice: LDom primary is in the process of a delayed
reconfiguration.
Any changes made to primary will only take effect after it
reboots.
```

3.2.12 设置默认服务

使用以下命令来设置逻辑域的默认服务。

表 3-17 设置默认服务的命令

命令	功能概述
<code>ldm add-vconscon</code>	添加虚拟控制台终端收集和分配服务。
<code>ldm add-vdiskserver</code>	添加虚拟磁盘服务器。
<code>ldm add-vswitch</code>	添加虚拟交换机服务。

ldm add-vconscon 命令

使用 `ldm add-vconscon` 命令将虚拟控制台终端收集和分配设备服务添加到控制域。

```
# ldm add-vconscon port-range=x-y vcc-name ldom
```

- `x`
指定虚拟控制台终端收集和分配设备端口号的下限值。
- `y`
指定虚拟控制台终端收集和分配设备端口号的上限值。
- `vcc-name`
指定虚拟控制台终端收集和分配设备的名称。
- `ldom`
指定逻辑域名称。控制域名称是 "primary"。如果指定的逻辑域不是控制域，逻辑域将成为控制台输出目标。
如果指定的逻辑域不是控制域，执行 `ldm add-vcc` 命令，然后使用 `svcadm enable` 命令启用逻辑域的虚拟网络终端服务器守护程序 (`vntsd`)。

```
# svcadm enable svc:/ldoms/vntsd
```

ldm add-vdiskserver 命令

使用 `ldm add-vdiskserver` 命令将虚拟磁盘服务器添加到控制域。

```
# ldm add-vdiskserver service-name ldom
```

- `service-name`
指定虚拟磁盘服务器的名称。此名称在系统内的所有虚拟磁盘服务器名称中必须是

唯一的。

- **ldom**
指定逻辑域名称。控制域名称是 "primary"。

ldm add-vswitch 命令

使用 **ldm add-vswitch** 命令将虚拟交换服务添加到控制域。

```
# ldm add-vswitch net-dev=device vswitch-name ldom
```

- **net-dev=device**
指定网络设备的名称。
- **vswitch-name**
指定虚拟交换机服务的名称。
- **ldom**
指定逻辑域名称。控制域名称是 "primary"。

3.2.13 配置虚拟 CPU

使用以下命令来配置虚拟 CPU。

表 3-18 配置虚拟 CPU 的命令

命令	功能概述
ldm add-vcpu	以线程为单位添加虚拟 CPU。
ldm set-vcpu	以线程为单位设置虚拟 CPU。
ldm remove-vcpu	以线程为单位移除虚拟 CPU。
ldm add-core	以内核为单位添加虚拟 CPU。
ldm set-core	以内核为单位设置虚拟 CPU。
ldm remove-core	以内核为单位移除虚拟 CPU。
ldm grow-socket	添加与特定 CPU 插槽相关联的虚拟 CPU。
ldm shrink-socket	移除与特定 CPU 插槽相关联的虚拟 CPU。

注 - 您不能同时以内核为单元和以线程为单元将 CPU 分配给逻辑域。请为各逻辑域选择其中一种分配单元。

ldm add-vcpu 命令

使用 **ldm add-vcpu** 命令将虚拟 CPU 添加到逻辑域。

```
# ldm add-vcpu number ldom
```

- **number**
指定要添加的虚拟 CPU 数（线程）。
 - **ldom**
指定逻辑域。如果是控制域，"primary" 是固定设置。
- 下面的示例将一个虚拟 CPU 添加到控制域。

```
# ldm add-vcpu 1 primary
-----
Notice: The primary domain is in the process of a delayed reconf
iguration.
Any changes made to the primary domain will only take effect aft
er it reboots.
-----
```

ldm set-vcpu 命令

使用 **ldm set-vcpu** 命令将虚拟 CPU 分配给逻辑域。

```
# ldm set-vcpu number ldom
```

- **number**
指定要分配的虚拟 CPU 数（线程）。
- **ldom**
指定逻辑域。如果是控制域，"primary" 是固定设置。

ldm remove-vcpu 命令

使用 **ldm remove-vcpu** 命令从逻辑域中移除虚拟 CPU。

```
# ldm remove-vcpu number ldom
```

- **number**
指定要移除的虚拟 CPU 数（线程）。
- **ldom**
指定逻辑域。如果是控制域，"primary" 是固定设置。

下面的示例从控制域中删除一个虚拟 CPU。

```
# ldm remove-vcpu 1 primary
-----
Notice: The primary domain is in the process of a delayed reconf
iguration.
Any changes made to the primary domain will only take effect aft
er it reboots.
-----
```

ldm add-core 命令

使用 ldm add-core 命令将特定内核添加到逻辑域。

```
# ldm add-core num ldom
# ldm add-core cid=core-ID [,core-ID [...]] ldom
```

- num
指定要添加的 CPU 内核数量。
- ldom
指定逻辑域。如果是控制域, "primary" 是固定设置。
- cid=core-ID [,core-ID [...]]
指定要添加的各 CPU 内核的 ID。

注 - 如果您通过指定 CID 添加 CPU 内核, 则不能通过动态重新配置更改配置。

注 - 有关 CPU 内核 ID (cid) 和 CPU 物理位置之间的关系, 请参见 ["2.4.1 逻辑域配置注意事项"](#) 中的 ["CPU 内核 ID 指定"](#)。

下面的示例将 1 个内核的 CPU 添加到控制域。

```
# ldm add-core 1 primary
```

ldm set-core 命令

使用 ldm set-core 命令在逻辑域上设置特定内核。

```
# ldm set-core num ldom
# ldm set-core cid=core-ID [,core-ID [...]] ldom
```

- num
指定要设置的 CPU 内核数量。
- ldom
指定逻辑域。如果是控制域, "primary" 是固定设置。
- cid=core-ID [,core-ID [...]]
指定所设置 CPU 内核的 ID。

注 - 如果您通过指定 CID 设置 CPU 内核, 则不能通过动态重新配置更改配置。

注 - 有关 CPU 内核 ID (cid) 和 CPU 物理位置之间的关系, 请参见 ["2.4 逻辑域的条件和设置"](#) 中的 ["CPU 内核 ID 指定"](#)。

ldm remove-core 命令

使用 ldm remove-core 命令从逻辑域中移除特定内核。

```
# ldm remove-core num ldom
# ldm remove-core cid=core-ID [,core-ID [...]] ldom
# ldm remove-core -g resource-group [-n num] ldom
```

- num
指定要设置的 CPU 内核数量。
- ldom
指定逻辑域。如果是控制域, "primary" 是固定设置。
- cid=core-ID [,core-ID [...]]
指定所设置 CPU 内核的 ID。
- -g resource-group
此选项通过指定一个资源组来移除 CPU 内核。对于 Oracle VM Server for SPARC 3.3 和更高版本, SPARC M12/M10 系统支持此选项。

在下面的示例中, 一个 CPU 内核从控制域被删除。

```
# ldm remove-core 1 primary
```

ldm grow-socket 命令

使用 ldm grow-socket 命令可将与 CPU 插槽相关联的虚拟 CPU 添加到逻辑域。Oracle VM for SPARC 3.2 和更高版本支持此子命令。

```
# ldm grow-socket vcpus=num socket_id=id ldom
# ldm grow-socket cores=num socket_id=id ldom
```

- num
指定要添加的 CPU 线程或 CPU 内核的数量。
- id
指定要添加的 CPU 插槽的 ID。
- ldom
指定逻辑域。

下面的示例将 CPU 内核添加到 CPU 插槽#2 中的控制域。

```
# ldm grow-socket cores=1 socket_id=2 primary
```

ldm shrink-socket 命令

使用 ldm shrink-socket 命令可移除与 CPU 插槽相关联的虚拟 CPU。Oracle VM for SPARC 3.2 和更高版本支持此子命令。

```
# ldm shrink-socket vcpus=num socket_id=id ldom
# ldm shrink-socket cores=num socket_id=id ldom
```

- num
指定要移除的 CPU 线程或 CPU 内核的数量。
- id
指定要移除的 CPU 插槽的 ID。
- ldom
指定逻辑域。

下面的示例移除与控制域中 CPU 插槽#2 相关联的 CPU 内核。

```
# ldm shrink-socket cores=1 socket_id=2 primary
```

3.2.14 配置虚拟内存

使用以下命令来配置虚拟内存。

表 3-19 配置虚拟内存的命令

命令	功能概述
ldm add-memory	添加虚拟内存。
ldm set-memory	设置虚拟内存。
ldm remove-memory	移除虚拟内存。
ldm grow-socket	添加与 CPU 插槽相关联的虚拟内存。
ldm shrink-socket	移除与 CPU 插槽相关联的虚拟内存。

ldm add-memory 命令

使用 ldm add-memory 命令将虚拟内存添加到逻辑域。

```
# ldm add-memory [--auto-adj] size[unit] ldom
```

- --auto-adj 选项
此选项将内存容量调整到 256 MB 范围。
- size
指定要添加的内存容量。最小分配单位是 256 MB。
- unit
指定内存容量的单位。您可以指定以下单位：G（千兆字节）、M（兆字节）和 K（千字节）。不区分大小写。
- ldom
指定逻辑域。如果是控制域，"primary" 是固定设置。

下面的示例添加 1 GB 虚拟内存。

```
# ldm add-memory 1g primary
-----
Notice: The primary domain is in the process of a delayed
reconfiguration.
Any changes made to the primary domain will only take effect
after it reboots.
-----
```

ldm set-memory 命令

使用 ldm set-memory 命令将虚拟内存分配给逻辑域。

```
# ldm set-memory [--auto-adj] size[unit] ldom
```

- **--auto-adj** 选项
此选项将内存容量调整到 256 MB 范围。
- **size**
指定要分配的内存容量。最小分配单位为 256 MB。
- **unit**
指定内存容量的单位。您可以指定以下单位：G（千兆字节）、M（兆字节）和 K（千字节）。不区分大小写。
- **ldom**
指定逻辑域。如果是控制域，"primary" 是固定设置。

ldm remove-memory 命令

使用 ldm remove-memory 命令从逻辑域中移除虚拟内存。

```
# ldm remove-memory [--auto-adj] size[unit] ldom
# ldm remove-memory -g resource-group [-s size[unit]] ldom
```

- **--auto-adj** 选项
此选项将内存容量调整到 256 MB 范围。
- **size**
指定内存容量。
- **unit**
指定内存容量的单位。您可以指定以下单位：G（千兆字节）、M（兆字节）和 K（千字节）。不区分大小写。
- **-g resource-group**
此选项通过指定一个资源组来移除内存。对于 Oracle VM Server for SPARC 3.3 和更高版本，SPARC M12/M10 系统支持此选项。
- **ldom**
指定逻辑域名称。

下面的示例删除 1 GB 的虚拟内存。

```
# ldm remove-memory 1g primary
-----
Notice: The primary domain is in the process of a delayed
reconfiguration.
Any changes made to the primary domain will only take effect
after it reboots.
-----
```

ldm grow-socket 命令

使用 ldm grow-socket 命令可将与 CPU 插槽相关联的虚拟内存添加到逻辑域。Oracle VM for SPARC 3.2 和更高版本支持此子命令。

```
# ldm grow-socket memory=size[unit] socket_id=id ldom
```

- size
指定要添加的内存容量。最小单位是 256 MB。
- unit
指定内存容量的单位。您可以指定 G（千兆字节）、M（兆字节）或 K（千字节）。不区分大小写。
- id
指定 CPU 插槽的 ID。
- ldom
指定逻辑域。

下面的示例将 1 GB 的虚拟内存添加到 CPU 插槽#2 中的控制域。

```
# ldm grow-socket memory=1G socket_id=2 primary
```

ldm shrink-socket 命令

使用 ldm shrink-socket 命令可将与 CPU 插槽相关联的虚拟内存从逻辑域中移除。Oracle VM for SPARC 3.2 和更高版本支持此子命令。

```
# ldm shrink-socket memory=size[unit] socket_id=id ldom
```

- size
指定要移除的内存容量。最小单位为 256 MB。
- unit
指定内存容量的单位。您可以指定 G（千兆字节）、M（兆字节）或 K（千字节）。不区分大小写。
- id
指定要移除的 CPU 插槽的 ID。
- ldom

指定逻辑域。

下面的示例移除 1 GB 的与控制域中 CPU 插槽#2 相关联的虚拟内存。

```
# ldm shrink-socket memory=1G socket_id=2 primary
```

3.2.15 设置 CPU 插槽限制条件

使用 `ldm set-socket` 命令可设置 CPU 插槽的约束条件。CPU 插槽约束条件可用于仅从指定的 CPU 插槽分配虚拟 CPU、内核和内存。Oracle VM for SPARC 3.3 和更高版本支持此子命令。

```
# ldm set-socket [-f] [--remap] socket_id=[id[,id[,...]]] ldom
```

- `-f`
此选项可清除非活动逻辑域的所有现存虚拟 CPU、内核和内存的约束条件。对于绑定的或活动的域，如果指定的 CPU 插槽的资源少于通过约束条件定义的资源，则可修改资源的约束条件和/或降级的 CPU 插槽。
- `--remap`
此选项可将分配给活动的物理资源的虚拟资源移至含有虚拟资源的域正在运行时的其他物理资源。
- `id`
指定 CPU 插槽的 ID。

下面的示例在逻辑域 `ldom1` 上设置 CPU 插槽#2 的约束条件。

```
# ldm set-socket socket_id=2 ldom1
```

3.2.16 设置逻辑域配置信息

使用以下命令设置逻辑域配置信息。

表 3-20 设置逻辑域配置信息的命令

命令	功能概述
<code>ldm add-spconfig</code>	保存逻辑域配置信息。
<code>ldm set-spconfig</code>	切换逻辑域配置信息。
<code>ldm remove-spconfig</code>	删除逻辑域配置信息。

ldm add-spconfig 命令

使用 `ldm add-spconfig` 命令保存逻辑域配置信息。


```
# ldm add-spconfig config-name
```

- **config-name**
指定配置名称。

不包括出厂默认值（配置名称：**factory-default**），在 XSCF 上您可以保存七组配置信息。最新保存的配置信息将成为默认值，将在下次及以后的启动中使用。

下面的示例使用名称 **config1** 保存配置信息。

```
# ldm add-spconfig config1
```

ldm set-spconfig 命令

使用 **ldm set-spconfig** 命令切换逻辑域配置信息。

```
# ldm set-spconfig config-name
```

- **config-name**
指定配置名称。下次启动时，逻辑域根据相应逻辑域配置信息启动。

ldm remove-spconfig 命令

使用 **ldm remove-spconfig** 命令删除 XSCF 中逻辑域配置信息。

```
# ldm remove-spconfig config-name
```

- **config-name**
指定配置名称。相应逻辑域配置信息被删除。

3.2.17 创建逻辑域

使用 **ldm add-domain** 命令创建来宾域。

```
# ldm add-domain ldom
```

- **ldom**
指定逻辑域名称。

3.2.18 配置 I/O 设备

使用以下命令配置 I/O 设备。

表 3-21 配置 I/O 设备的命令

命令	功能概述
ldm add-io	添加 I/O 设备。
ldm remove-io	删除 I/O 设备。

请注意：取决于 Oracle VM Server for SPARC 的版本，可将 I/O 设备动态重新配置。有关详情，请参见 表 1-2、所使用服务器最新版本的《产品说明》、所使用版本的《Oracle VM Server for SPARC Release Notes》以及《Oracle VM Server for SPARC Administration Guide》。

ldm add-io 命令

使用 ldm add-io 命令将 I/O 设备添加到逻辑域。

```
# ldm add-io (bus | device | vf-name) ldom
```

- bus
指定 PCIe 总线（根合成体）。
- device
指定直接 I/O 可分配设备的名称（PCIe 端点设备名）。
- vf-name
指定 PCIe SR-IOV 的虚拟功能名。
- ldom
指定添加总线、设备或 SR-IOV 虚拟功能的逻辑域。

ldm remove-io 命令

使用 ldm remove-io 命令从逻辑域删除 I/O 设备。

```
# ldm remove-io [-n] (bus | device | vf-name) ldom
```

- -n
提前确认本命令是否成功执行。
- bus
指定 PCIe 总线（根合成体）。
- device
指定直接 I/O 可分配设备的名称（PCIe 端点设备名）。
- vf-name
指定 PCIe SR-IOV 的虚拟功能名。
- ldom
指定移除总线、设备或 SR-IOV 虚拟功能的逻辑域。

下面是从控制域 (primary) 删除 PCIe 总线 (PCIE1) 并将其分配给 I/O 域的步骤的示例。另外，对于 PCIe 端点，此步骤相同。

注 - 出厂默认值 (factory-default) 情况下，所有的 I/O 设备被分配给主域。

1. 执行 **Oracle Solaris** 的 **fmadm faulty** 命令和 **XSCF** 固件的 **showstatus** 命令，确认设备已经无错误信息。

```
# fmadm faulty
```

```
XSCF> showstatus
```

执行硬件维护作业后，如果存在错误信息，请执行 **fmadm repair** 命令清除该错误信息。

```
# fmadm repaired fmri | lable
```

2. 将控制域设置为延迟重新配置模式。
如果 I/O 设备被动态重新配置，则您可以省略此步骤。

```
# ldm start-reconf primary
```

3. 从控制域释放已分配的 **PCIe** 端点 (**PCIE1**) 。

```
# ldm remove-io PCIE1 primary
```

4. 重新启动控制域。
如果 I/O 设备被动态重新配置，则您可以省略此步骤。

```
# shutdown -y -g0 -i6
```

5. 如果 **Oracle Solaris** 在分配目的地 **I/O** 域中正在运行，则将其停止。
如果 I/O 设备被动态重新配置，则您可以省略此步骤。

```
# ldm stop-domain ldom1
```

6. 将 **PCIe** 端点 (**PCIE1**) 的分配添加到 **I/O** 域。

```
# ldm add-io PCIE1 ldom1
```

7. 启动分配目的地 **I/O** 域的 **Oracle Solaris** 。
- 如果 I/O 设备被动态重新配置，则您可以省略此步骤。

```
# ldm start-domain ldom1
```

3.2.19 创建或销毁 SR-IOV 虚拟功能

使用下述命令可创建或销毁 SR-IOV 虚拟功能。

表 3-22 创建或销毁 SR-IOV 虚拟功能的命令

命令	功能概述
ldm create-vf	创建虚拟功能 (VF)。
ldm destroy-vf	销毁虚拟功能 (VF)。

请注意：取决于 Oracle VM Server for SPARC 的版本，可将 I/O 设备动态重新配置。有关详情，请参见 表 1-2、所使用服务器的《产品说明》、所使用版本的《Oracle VM Server for SPARC Release Notes》以及《Oracle VM Server for SPARC Administration Guide》。

创建 SR-IOV 虚拟功能

使用 ldm create-vf 命令创建虚拟功能 (VF)。

ldm create-vf 命令

当在批处理中创建多个虚拟功能 (VFs) 时
ldm create-vf -n *number* | max *pf-name*

当一次创建一个虚拟功能 (VF) 时
ldm create-vf *pf-name*

- -n
指定虚拟功能的数量 (VFs)。
- pf-name
支持 SR-IOV 功能的 PCIe 设备名

下面是创建 SR-IOV 虚拟功能 (VF) 和将虚拟设备添加到 I/O 域的步骤示例。

1. 查看支持 **SR-IOV** 功能的 **PCIe** 设备。
在下面的执行示例中，来自所安装 PCIe 卡的 PCIe 设备名 "/BB0/PCI0/IOVNET.PF0" 和 "/BB0/PCI0/IOVNET.PF1" 都适用并分配给控制域。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	primary	IOV

PCIE6	BUS	PCIE6	primary	IOV
PCIE7	BUS	PCIE7	primary	IOV
(省略)				
/BB0/PCI6	PCIE	PCIE6	primary	EMP
/BB0/PCI9	PCIE	PCIE7	primary	EMP
/BB0/PCI10	PCIE	PCIE7	primary	EMP
/BB0/PCI0/IOVNET.PF0	PF	PCIE1	primary	
/BB0/PCI0/IOVNET.PF1	PF	PCIE1	primary	
/BB0/CMUL/NET0/IOVNET.PF0	PF	PCIE0	primary	
/BB0/CMUL/NET0/IOVNET.PF1	PF	PCIE0	primary	
/BB0/CMUL/NET2/IOVNET.PF1	PF	PCIE4	primary	
/BB0/CMUL/NET2/IOVNET.PF0	PF	PCIE4	primary	

- 将控制域设置为延迟重新配置模式。
如果 I/O 设备被动态重新配置，没有必要进行此操作。转到步骤 3。

```
# ldm start-reconf primary
```

- 创建虚拟功能 (VF)。

```
# ldm create-vf /BB0/PCI0/IOVNET.PF0
# ldm create-vf /BB0/PCI0/IOVNET.PF1
```

- 重新启动控制域。
如果 I/O 设备被动态重新配置，没有必要进行此操作。转到步骤 6。

```
# shutdown -y -g0 -i6
```

- 停止所有在分配目的地 I/O 域 (iodom00 和 iodom01) 中运行的 Oracle Solaris。
如果 I/O 设备被动态重新配置，没有必要进行此操作。转到步骤 6。

```
# ldm stop-domain ldom00
# ldm stop-domain ldom01
```

- 将虚拟功能 (/BB0/PCI0/IOVNET.PF0.VF0 和 /BB0/PCI0/IOVNET.PF1.VF1) 添加到 I/O 域。

```
# ldm add-io /BB0/PCI0/IOVNET.PF0.VF0 ldom00
# ldm add-io /BB0/PCI0/IOVNET.PF1.VF0 ldom01
```

- 启动 I/O 域的 Oracle Solaris 。
如果 I/O 设备被动态重新配置，没有必要进行此操作。

```
# ldm start-domain ldom00
# ldm start-domain ldom01
```

销毁 SR-IOV 虚拟功能

使用 `ldm destroy-vf` 命令销毁虚拟功能 (VF)。

ldm destroy-vf 命令

- 一次销毁多个虚拟功能 (VFs)

```
# ldm destroy-vf -n number | max pf-name
```

- 销毁指定的虚拟功能 (VF)

```
# ldm destroy-vf vf-name
```

- `-n`
指定虚拟功能的数量 (VFs)。
- `pf-name`
支持 SR-IOV 功能的 PCIe 设备名
- `vf-name`
虚拟功能 (VF) 的名称

下面是从 I/O 域断开虚拟设备并销毁 SR-IOV 虚拟功能 (VF) 的步骤示例。

1. 查看逻辑域中 I/O 设备的使用状况。
在下面的执行示例中，虚拟设备 `"/BB0/PCI0/IOVNET.PF0.VF0"` 和 `"/BB0/PCI0/IOVNET.PF0.VF1"` 被分配给 I/O 域 (`iodom00` 和 `iodom01`)。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	primary	IOV
PCIE6	BUS	PCIE6	primary	IOV
PCIE7	BUS	PCIE7	primary	IOV
(省略)				
/BB0/PCI6	PCIE	PCIE6	primary	EMP
/BB0/PCI9	PCIE	PCIE7	primary	EMP
/BB0/PCI10	PCIE	PCIE7	primary	EMP
/BB0/PCI0/IOVNET.PF0	PF	PCIE1	primary	
/BB0/PCI0/IOVNET.PF1	PF	PCIE1	primary	
/BB0/CMUL/NET0/IOVNET.PF0	PF	PCIE0	primary	
/BB0/CMUL/NET0/IOVNET.PF1	PF	PCIE0	primary	
/BB0/CMUL/NET2/IOVNET.PF1	PF	PCIE4	primary	
/BB0/CMUL/NET2/IOVNET.PF0	PF	PCIE4	primary	
/BB0/PCI0/IOVNET.PF0.VF0	VF	PCIE1	iodom00	
/BB0/PCI0/IOVNET.PF0.VF1	VF	PCIE1	iodom01	

2. 停止分配虚拟功能的 I/O 域中所有已启动的 **Oracle Solaris** 。
如果 I/O 设备被动态重新配置，没有必要进行此操作。转到步骤 3。

```
# ldm stop-domain iodom00
# ldm stop-domain iodom01
```

3. 从 I/O 域，释放虚拟功能 (/BB0/PCI0/IOVNET.PF0.VF0 和 /BB0/PCI0/IOVNET.PF0.VF1) 的分配。

```
# ldm remove-io /BB0/PCI0/IOVNET.PF0.VF0 iodom00
# ldm remove-io /BB0/PCI0/IOVNET.PF0.VF1 iodom01
```

4. 启动 I/O 域的 **Oracle Solaris** 。
如果 I/O 设备被动态重新配置，没有必要进行此操作。转到步骤 6。

```
# ldm start-domain ldom00
# ldm start-domain ldom01
```

5. 将控制域设置为延迟重新配置模式。
如果 I/O 设备被动态重新配置，没有必要进行此操作。转到步骤 6。

```
# ldm start-reconf primary
```

6. 销毁虚拟功能 (VF)。

```
# ldm destroy-vf /BB0/PCI0/IOVNET.PF0
# ldm destroy-vf /BB0/PCI0/IOVNET.PF1
```

7. 重新启动控制域。
如果 I/O 设备被动态重新配置，没有必要进行此操作。

```
# shutdown -y -g0 -i6
```

3.2.20 配置虚拟网络设备

使用以下命令配置虚拟网络设备。

表 3-23 配置虚拟网络设备的命令

命令	功能概述
ldm add-vnet	添加虚拟网络设备。
ldm remove-vnet	移除虚拟网络设备。

ldm add-vnet 命令

使用 ldm add-vnet 命令将虚拟网络设备添加到逻辑域。

```
# ldm add-vnet if-name vswitch-name ldom
```

- if-name
指定分配给虚拟网络设备实例的接口的名称。
- vswitch-name
指定虚拟交换机服务的名称。
- ldom
指定逻辑域名称。

ldm remove-vnet 命令

使用 ldm remove-vnet 命令移除虚拟网络设备。

```
# ldm remove-vnet if-name ldom
```

- if-name
指定接口名称。
- ldom
指定逻辑域名称。

3.2.21 配置虚拟磁盘服务器

使用以下命令配置虚拟磁盘服务器。

表 3-24 配置虚拟磁盘服务器的命令

命令	功能概述
ldm add-vdiskserverdevice	将设备卷添加到虚拟磁盘服务器。
ldm remove-vdiskserverdevice	将设备卷从虚拟磁盘服务器移除。

ldm add-vdiskserverdevice 命令

使用 ldm add-vdiskserverdevice 命令将卷添加到虚拟磁盘服务器。

```
# ldm add-vdiskserverdevice backend volume-name@service-name
```

- backend
指定物理磁盘路径。除文件外，您还可以指定磁盘、磁盘切片等的路径。

- **volume-name**
指定在虚拟磁盘服务器内唯一的卷名。
- **service-name**
指定虚拟磁盘服务的名称。

下面的示例创建和添加 100GB 的文件，作为磁盘服务器 **primary-vds0** 的卷。

```
# mkfile 100g /ldoms/domain/test/fdisk1
# ldm add-vdiskserverdevice /ldoms/domain/test/fdisk1 vol2@primary-vds0
```

ldm remove-vdiskserverdevice 命令

使用 **ldm remove-vdiskserverdevice** 命令从虚拟磁盘服务器中删除卷。

```
# ldm remove-vdiskserverdevice volume-name@service-name
```

- **volume-name**
指定卷的名称。
- **service-name**
指定虚拟磁盘服务器的名称。

下面的示例从虚拟磁盘服务器 **primary-vds0** 中删除卷 **vol2**。

```
# ldm remove-vdiskserverdevice vol2@primary-vds0
```

3.2.22 配置虚拟磁盘

使用以下命令配置虚拟磁盘。

表 3-25 配置虚拟磁盘的命令

命令	功能概述
ldm add-vdisk	添加虚拟磁盘。
ldm remove-vdisk	移除虚拟磁盘。

ldm add-vdisk 命令

使用 **ldm add-vdisk** 命令将虚拟设备作为虚拟磁盘添加到来宾域。

```
# ldm add-vdisk disk-name volume-name@service-name ldom
```

- **disk-name**
指定虚拟磁盘的名称。
- **volume-name**

指定将要连接到现有虚拟磁盘服务器的设备的名称。

- **service-name**
指定将要连接到现有虚拟磁盘服务器的服务名称。
- **ldom**
指定逻辑域。

下面的示例将已添加的虚拟磁盘服务器 **primary-vds0** 的设备 **vol2** 作为虚拟磁盘 **vdisk1** 添加来宾域 **ldom1**。

```
# ldm add-vdisk vdisk1 vol2@primary-vds0 ldom1
```

ldm remove-vdisk 命令

使用 **ldm remove-vdisk** 命令从来宾域移除虚拟磁盘。

```
# ldm remove-vdisk disk-name ldom
```

- **disk-name**
指定虚拟磁盘的名称。
- **ldom**
指定逻辑域名称。

下面的示例从来宾域 **ldom1** 中移除虚拟磁盘 **vdisk2**。

```
# ldm remove-vdisk vdisk2 ldom1
```

3.2.23 配置虚拟控制台

使用 **ldm set-vconsole** 命令将虚拟控制台分配给来宾域，并且设置来宾域控制台日志输出为文件。

```
# ldm set-vconsole port=port-num ldom  
# ldm set-vconsole [service=vcc-server] log=on ldom
```

- **port-num**
指定虚拟控制台的端口号。
- **ldom**
指定逻辑域名称。
- **vcc-server**
如果为逻辑域指定虚拟控制台终端收集和分配设备服务，而不是为控制域，请指定该逻辑域的名称。因此，将指定为 *ldom* 的来宾域的控制台输出和日志输出到指定逻辑域。

注 - 当 Oracle Solaris 11 提供虚拟控制台终端分配设备服务时，将支持逻辑域的指定 **service=vcc-server**。

注 - 日志内容以名称为 `/var/log/vntsd/`（来宾域名称）`/console-log` 的文件被输出在指定为 `ldm add-vconscon` 的服务域中。

3.2.24 配置启动设备

使用 `ldm set-variable` 命令设置来宾域的启动设备。也使用 `ldm set-variable` 命令设置在来宾域启动时，是否同时自动启动 Oracle Solaris。默认情况下，Oracle Solaris 自动与来宾域一起启动。

```
# ldm set-variable boot-device=disk-name ldom
# ldm set-variable auto-boot?={true|false} ldom
```

- `disk-name`
指定虚拟磁盘的名称。
- `ldom`
指定逻辑域名称。
- `true|false`
对自动启动指定 "true"。对不自动启动指定 "false"。

注 - 大写字符不能用于 OpenBoot PROM 的设备别名。如果由 `ldm add-vdisk` 命令添加的虚拟磁盘的名称包含大写字符，需将虚拟磁盘名的所有字母改成小写字母。然后，以 OpenBoot PROM 的环境变量 `boot-device` 进行设置。

3.2.25 绑定资源

使用 `ldm bind-domain` 命令将资源绑定到来宾域。

```
# ldm bind-domain ldom
```

- `ldom`
指定逻辑域名称。

3.2.26 启动来宾域

使用 `ldm start-domain` 命令启动来宾域。

```
# ldm start-domain ldom
```

- `ldom`
指定逻辑域名称。

3.2.27 指定关闭组

使用 `ldm set-domain` 命令指定关闭组。

```
# ldm set-domain shutdown-group=num ldom
```

- `num`
指定关闭组。您可以指定 1 到 15 间的一个数字值。
- `ldom`
指定目标逻辑域的名称。您不能指定 "primary"（控制域）。

注 - 您可以仅在来宾域处于 `inactive` 或 `bound` 状态时更改关闭组。

3.2.28 重新配置设备

当重新配置设备时，使用 `devfsadm` 命令。登录到分配虚拟磁盘的来宾域，并执行 `devfsadm` 命令。

```
guest# devfsadm
```

`devfsadm` 命令将所有驱动程序加载到系统中，以便连接到尽可能多的设备。然后，此命令在 `/devices` 目录创建设备文件，在 `/dev` 目录创建逻辑链接。有关 `devfsadm` 命令的详情，请参见 Oracle Solaris 的参考手册。

3.2.29 启用恢复模式

如需启用恢复模式，请执行 `svccfg` 命令将 `ldmd/recovery_mode` 属性设置为 `auto`。然后，执行 `svcadm` 命令启用设置值。
请注意，Oracle VM Server for SPARC 3.3 和更高版本默认启用恢复模式。

```
# svccfg -s ldmd setprop ldmd/recovery_mode = astring: auto
# svcadm refresh ldmd
```

从 XCP 2230 开始，支持 Oracle VM Server for SPARC 3.1 所提供的恢复模式。有关恢复模式的详情，请参见所使用版本的《Oracle VM Server for SPARC Administration Guide》中的 "Handling Hardware Errors"。此外，关于恢复模式所需的修正，请参见所使用版本的《Oracle VM Server for SPARC Release Notes》。

物理分区配置示例

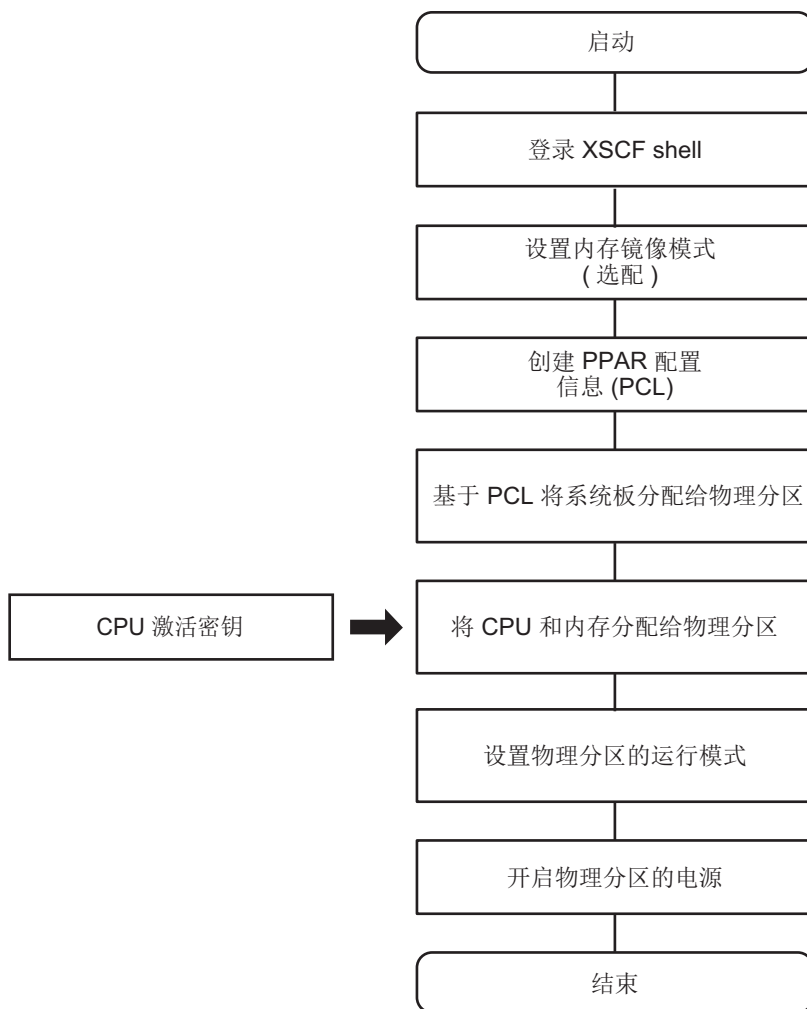
本章介绍配置物理分区的示例。

- [物理分区配置流程](#)
- [物理分区配置操作示例](#)

4.1 物理分区配置流程

如需配置域，请首先配置各个物理分区。使用 XSCF 固件来配置物理分区。[图 4-1](#) 显示了模块化架构配置中的物理分区配置流程。

图 4-1 配置物理分区



4.2 物理分区配置操作示例

本节介绍在具有四个 SPARC M10-4S 单元的配置中配置两个物理分区的示例。

注 -SPARC M12-2S 也使用相同的操作。

- 确定物理分区编号 (PPAR-ID)
确定识别物理分区的编号 (PPAR-ID)。您可以将 0 至 15 的数字指定为物理分区编号。然而，根据系统配置的不同，可能有不同限制。当配置一个物理分区时，需要从 SPARC M12/M10 系统上所安装机箱的 BB-ID 中选择一个来确定物理分区编号。如

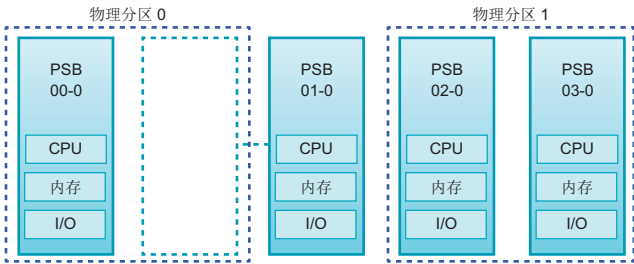
果将 SPARC M12/M10 系统中未安装的机箱的 BB-ID 指定为物理分区编号，则该物理分区的上电将失败。

例如，在配置物理分区时，当系统上安装的 SPARC M10-4S 单元的 BB-ID 是 BB#0、BB#1、BB#2 和 BB#3 时，可以指定 0、1、2 或 3 为物理分区编号。本节我们配置物理分区编号为 0 和 1 的两个物理分区。

- PSB (BB) 的分配
将系统板 (PSB 00-0 和 PSB 01-0) 分配到物理分区 0，将系统板 (PSB 02-0 和 PSB 03-0) 分配到物理分区 1。但是，使 PSB 01-0 保持可分配状态，而不是实际分配它。
- 镜像模式设置（选配）
仅在 PSB 00-0 上设置内存镜像。
- 配置策略/内存无效/IO 无效
此示例将物理分区 0 和 1 的配置策略设置为 "system"，不使用内存无效和 I/O 无效。
- 物理分区运行模式
将诊断信息的信息级别设置为 "normal"（标准）。

图 4-2 是已配置的物理分区的概念图。

图 4-2 已配置的物理分区的概念图



1. 用具有 **platadm** 权限的用户帐户登录 **XSCF shell**。
2. 执行 **setupfru** 命令来设置内存镜像。

```
XSCF> setupfru -m y sb 00-0
```

3. 根据需要执行 **showfru** 命令，查看内存镜像的状态。
检查系统板 (PSB 00-0) 上内存镜像的状态。如果 [Memory Mirror Mode]（内存镜像模式）全部显示 "yes"，则镜像模式已正确设置。

```
XSCF> showfru sb 00-0
Device      Location      Memory Mirror Mode
sb          00-0
  cpu       00-0-0      yes
  cpu       00-0-1      yes
  cpu       00-0-2      yes
  cpu       00-0-3      yes
XSCF>
```

4. 执行 **setpcl** 命令来创建物理分区配置信息（PPAR 配置信息）。

分别将物理系统板 PSB 00-0 和 PSB 01-0 映射到物理分区 0 的逻辑系统板 LSB 0 和 LSB 1。分别将物理系统板 PSB 02-0 和 PSB 03-0 映射到物理分区 0 的逻辑系统板 LSB 1 和 LSB 1。此示例将配置策略设置为 "system"，不使用内存无效和 I/O 无效。

```
XSCF> setpcl -p 0 -a 0=00-0 1=01-0
XSCF> setpcl -p 0 -s policy=system
XSCF> setpcl -p 0 -s no-mem=false
XSCF> setpcl -p 0 -s no-io=false
XSCF> setpcl -p 1 -a 0=02-0 1=03-0
XSCF> setpcl -p 1 -s policy=system
XSCF> setpcl -p 1 -s no-mem=false
XSCF> setpcl -p 1 -s no-io=false
```

5. 执行 **showpcl** 命令检查 **PPAR** 配置信息的内容。
[PPAR-ID] 显示物理分区编号，[LSB] 显示逻辑系统板编号，[PSB] 显示 PSB 编号。

```
XSCF> showpcl -v -a
PPAR-ID  LSB  PSB  Status  No-Mem  No-IO  Cfg-policy
00
          00    00-0      Powered Off      False  False      System
          01    01-0      False  False
01
          00    02-0      Powered Off      False  False      System
          01    03-0      False  False
```

6. 执行 **showboards** 命令检查 **PSB**。
[PPAR-ID] 下显示为 "SP"，[Assignment] 下显示为 "Available" 的 PSB 可供分配。
"SP" 表示系统板池状态，其含义是该系统板不属于任何物理分区。

```
XSCF> showboards -a
PSB  PPAR-ID(LSB)  Assignment  Pwr  Conn  Conf  Test  Fault
-----
00-0 SP           Available  n    n    n    Passed  Normal
01-0 SP           Available  n    n    n    Passed  Normal
02-0 SP           Available  n    n    n    Passed  Normal
03-0 SP           Available  n    n    n    Passed  Normal
```

7. 执行 **addboard** 命令分配 **PSB**。

```
XSCF> addboard -c assign -p 0 00-0
XSCF> addboard -c assign -p 1 02-0 03-0
```

8. 执行 **showboards** 命令检查所分配 **PSB** 的状态。
当已分配的 PSB 的 [R] 栏显示 "*"（星号），已经设置物理分区和逻辑系统板 (LSB)，如 [PPAR-ID] 栏的设置所显示，则 PSB 已经正确分配。

XSCF> showboards -v -a								
PSB	R	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	----	-----	-----	----	----	----	-----	-----
00-0	*	00(00)	Assigned	n	n	n	Passed	Normal
01-0		SP	Available	n	n	n	Passed	Normal
02-0	*	01(00)	Assigned	n	n	n	Passed	Normal
03-0	*	01(01)	Assigned	n	n	n	Passed	Normal

9. 执行 **showcodactivation** 命令来检查是否 **CPU** 激活密钥在该系统登录。
 如果仅显示标题，则 CPU 激活密钥未在 XSCF 中登录。若需添加 CPU 激活密钥，请执行从步骤 10 开始的流程。如果仅显示索引号，则 CPU 激活密钥未在 XSCF 中登录。如需继续此流程，请转到步骤 13。

XSCF> showcodactivation		
Index	Description	Count
-----	-----	-----

10. 执行 **addcodactivation** 命令以登录 **CPU** 激活密钥。

XSCF> addcodactivation "Product: SPARC M10-4S	
SequenceNumber: 10005	
Cpu: noExpiration 2	
Text-Signature-SHA256-RSA2048:	
PSSrElBrse/r69AVSVFd38sT6AZm2bxUDdPQHKbtxgvgZPsrtYguqiNUieB+mTDC	
:	
:	
b1GCKFx1RH27FdVHiB2H0A=="	
Above Key will be added, Continue?[y n]: y	

11. 执行 **showcodactivation** 命令，检查已添加 **CPU** 激活密钥的内容。
 如果执行该命令显示已购买资源的 CPU 激活号，则 CPU 激活密钥已经正确注册。

XSCF> showcodactivation		
Index	Description	Count
-----	-----	-----
0	PROC	2
1	PROC	2
2	PROC	2
3	PROC	2

12. 执行 **setcod** 命令将 **CPU** 内核资源分配给物理分区。

XSCF> setcod -p 0 -s cpu -c set 4	
PROC Permits assigned for PPAR 0 : 0 -> 4	
PROC Permits assigned for PPAR will be changed.	
Continue? [y n] : y	
Completed.	

```
XSCF> setcod -p 1 -s cpu -c set 4
PROC Permits assigned for PPAR 1 : 0 -> 4

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 -XCP 2250 和更早版本的 XSCF 固件不支持 -c add、-c delete 和 -c set 选项。按如下所示指定 setcod 命令的选项可交互式添加和删除 CPU 内核资源。

```
XSCF> setcod -s cpu
```

13. 执行 **resetdateoffset** 命令重置 XSCF 管理的时间和物理分区管理的时间之间的差异。

```
XSCF> resetdateoffset -p 0
XSCF> resetdateoffset -p 1
```

14. 执行 **setpparmode** 命令并将诊断信息的详细级别设置为 "normal" (标准)。

```
XSCF> setpparmode -p 0 -m message=normal
XSCF> setpparmode -p 1 -m message=normal
```

15. 执行 **poweron** 命令来开启物理分区的电源。
下面的示例开启物理分区 0 和 1 的电源。

```
XSCF> poweron -p 0
PPAR-IDs to power on:00
Continue? [y|n] :y
00 :Powering on

*Note*
This command only issues the instruction to power-on.
The result of the instruction can be checked by the "showlogs power".

XSCF> poweron -p 1
PPAR-IDs to power on:01
Continue? [y|n] :y
01 :Powering on

*Note*
This command only issues the instruction to power-on.
The result of the instruction can be checked by the "showlogs power".
```

16. 执行 **showboards** 命令检查 PSB 状态。
如果在所有的 [Pwr]、[Conn] 和 [Conf] 栏显示 "y"，则物理分区正在正确运行。

```
XSCF> showboards -a
```

PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal
02-0	01(00)	Assigned	y	y	y	Passed	Normal
03-0	01(01)	Assigned	y	y	y	Passed	Normal

逻辑域配置示例

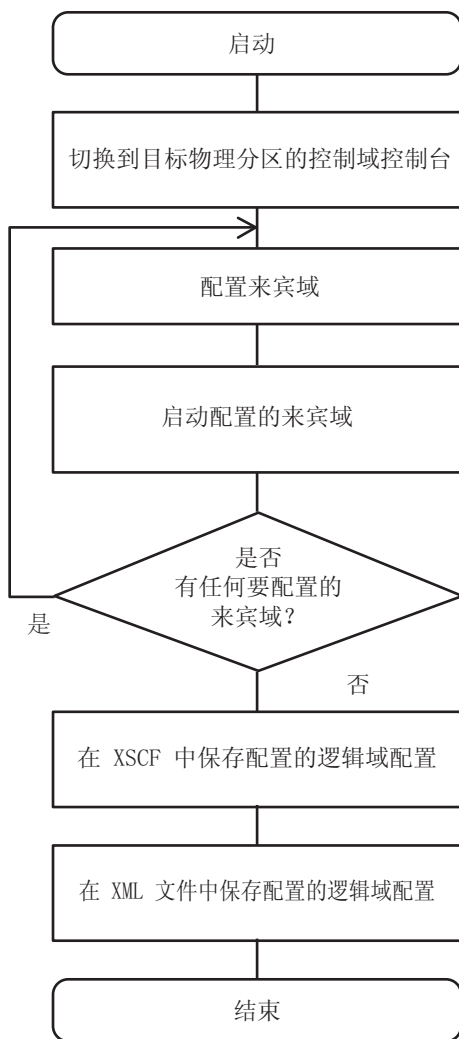
本章介绍配置逻辑域的示例。

- [逻辑域配置流程](#)
- [逻辑域配置操作示例](#)

5.1 逻辑域配置流程

配置物理分区和开启物理分区的电源后，配置各逻辑域。使用 Oracle VM Server for SPARC 的管理软件：Logical Domains Manager（逻辑域管理器）的 `ldm` 命令来配置逻辑域。[图 5-1](#) 显示了逻辑域配置的流程。

图 5-1 配置逻辑域



物理分区配置期间，系统板上的硬件资源被分配。紧随逻辑域配置后，将所有资源都分配给控制域。因此，在逻辑域配置后，根据需要，将硬件资源重新分配给来宾域或不同的服务域。

完成逻辑域配置后，将逻辑域配置信息保存到 XSCF。开启物理分区的电源后，物理分区中的所有逻辑域按照此配置信息启动。

5.2 逻辑域配置操作示例

本节介绍一个具体逻辑域配置示例。

若需配置逻辑域，您需要定义不同名称和指定不同资源以及其编号。在开始工作之前，准备和组织该信息。

下表显示本节中作为示例使用的信息。[表 5-1](#) 列出了名称定义的示例。[表 5-2](#) 列出了分配给控制域的资源示例。[表 5-3](#) 列出了分配给来宾域的资源示例。

表 5-1 名称定义示例

项目	名称
虚拟控制台终端收集和分配设备的名称	primary-vcc0
虚拟磁盘服务器的名称	primary-vds0
虚拟交换机服务的名称	primary-vsw0
虚拟网络设备接口的名称	vnet1
虚拟设备的名称	vol1
虚拟磁盘的名称	vdisk1
来宾域的名称	ldom1
逻辑域配置信息的名称	config1

表 5-2 分配给控制域的资源示例

项目	值
虚拟控制台终端端口号范围	5000 至 5100
用于虚拟交换机服务的设备	net0
用于虚拟磁盘服务的设备	/ldoms/domain/test/fdisk0
已分配的 CPU 的数量	4 个内核
已分配的内存的容量	8 GB

表 5-3 分配给来宾域的资源示例

项目	值
已分配的虚拟控制台终端的端口号	5000
已分配的 CPU 的数量	2 个内核
已分配的内存的容量	4 GB

5.2.1 登录控制域

本节中的说明假定目标的物理分区编号为 0。

1. 用具有 **platadm** 权限的用户帐户登录 **XSCF shell**。
2. 执行 **poweron** 命令来开启物理分区 **0** 的电源。

```
XSCF> poweron -p 0
PPAR-IDs to power on:00
Continue? [y|n] :y
00 :Powering on
*Note*
This command only issues the instruction to power-on.
The result of the instruction can be checked by the "showlogs power".
XSCF>
```

3. 执行 **console** 命令以切换到物理分区 0 的控制域控制台。

```
XSCF> console -p 0
Console contents may be logged. Connect to PPAR-ID 0?[y|n] :y
```

注 - 如果 Oracle Solaris 未被设置为自动启动，请执行启动操作并等到启动完成。

4. 以 **root**（根）用户身份登录控制域。
5. 执行 **svcs ldmd** 命令，确认 **Logical Domains Manager**（逻辑域管理器）是否正在运行。
Logical Domains Manager（逻辑域管理器）的服务名称是 **ldmd**。如果 [STATE] 栏显示 "online"，则是正在运行。

```
# svcs ldmd
STATE      STIME      FMRI
online 16:25:31 svc:/ldoms/ldmd:default
```

注 - 如果 Logical Domains Manager（逻辑域管理器）未启动，请执行 **svcadm** 命令可将其启动。有关 **svcadm** 命令的详情，请参见 Oracle Solaris 的参考手册。

5.2.2 设置默认服务

1. 执行 **ldm add-vconscon** 命令将虚拟控制台终端收集和分配设备服务添加到控制域。
下面示例将 5000 至 5100 端口号，名称为 **primary-vcc0** 的虚拟控制台终端收集和分配设备添加到控制域。

```
# ldm add-vconscon port-range=5000-5100 primary-vcc0 primary
```

2. 执行 **ldm add-vdiskserver** 命令将虚拟磁盘服务器添加到控制域。
下面示例将名称为 **primary-vds0** 的虚拟磁盘服务器添加到控制域。

```
# ldm add-vdiskserver primary-vds0 primary
```


3. 执行 **ldm add-vswitch** 命令将虚拟交换服务添加到控制域。
下面示例将名称为 `primary-vsw0` 的虚拟交换机服务添加到控制域，并且 `net0` 用作一个设备。

```
# ldm add-vswitch net-dev=net0 primary-vsw0 primary
```

4. 执行 **ldm list-services** 命令来确认所设置的服务。
下面示例显示控制域的服务。

```
# ldm list-services primary
```

5.2.3 进行控制域的初始设置

1. 执行 **ldm start-reconf** 命令启动延迟重新配置模式。
控制域名称固定为 "primary"。

```
# ldm start-reconf primary
```

2. 执行 **ldm list-permits** 命令检查可以分配的 CPU 内核数量。

```
# ldm list-permits
```

3. 执行 **ldm set-core** 命令将虚拟 CPU 分配给控制域。
下面示例将 4 个内核的 CPU 分配给控制域。

```
# ldm set-core 4 primary
```

4. 执行 **ldm set-memory** 命令分配虚拟内存。
下面示例将 8 GB 的内存分配给控制域。

```
# ldm set-memory 8g primary
```

5. 执行 **ldm list-bindings** 命令检查资源的分配状况。

```
# ldm list-bindings primary
```

6. 执行 **ldm add-spconfig** 命令将逻辑域配置保存到 XSCF。
下面示例使用名称 "config1" 保存逻辑域配置信息。

```
# ldm add-spconfig config1
```

7. 执行 **XSCF** 固件的 **poweroff** 和 **poweron** 命令以关闭物理分区的电源，然后重新开启电源。

```
XSCF> poweroff -p 0
XSCF> poweron -p 0
```

8. 重新启动控制域后，执行 **list-bindings** 命令检查资源的分配状态。
如果以下命令显示与步骤 4 中说明的分配状态相同，则设置正确。

```
# ldm list-bindings primary
```

5.2.4 配置来宾域

1. 执行 **ldm list-devices** 命令检查可分配给来宾域的资源。
此命令显示任何域都未使用的资源。

```
# ldm list-devices
```

2. 执行 **ldm add-domain** 命令创建来宾域 **ldom1**。

```
# ldm add-domain ldom1
```

3. 执行 **ldm list-permits** 命令检查可以分配的 **CPU** 内核数量。

```
# ldm list-permits
```

4. 执行 **ldm set-core** 命令添加两核虚拟 **CPU**。

```
# ldm set-core 2 ldom1
```

5. 执行 **ldm add-memory** 命令添加 **4 GB** 虚拟内存。

```
# ldm add-memory 4g ldom1
```

6. 执行 **ldm add-vnet** 命令添加虚拟网络设备。

```
# ldm add-vnet vnet1 primary-vsw0 ldom1
```

7. 执行 **ldm add-vdiskserverdevice** 命令将设备添加到虚拟磁盘服务器。
下面示例使用 **mkfile** 命令来创建一个 100 GB 的普通文件用作控制域中的数据存储空间。此文件作为设备被添加。

```
# mkfile 100g /ldoms/domain/test/fdisk0
# ldm add-vdiskserverdevice /ldoms/domain/test/fdisk0 vol1@primary-vds0
```

8. 执行 **ldm add-vdisk** 命令添加虚拟磁盘。
将步骤 6 中添加的虚拟磁盘添加到来宾域。

```
# ldm add-vdisk vdisk1 vol1@primary-vds0 ldom1
```

9. 执行 **ldm set-vconsole** 命令分配 **5000** 端口号的虚拟控制台。

```
# ldm set-vconsole port=5000 ldom1
```

10. 执行 **ldm set-variable** 命令设置来宾域的启动设备。

```
# ldm set-variable boot-device=vdisk1 ldom1
```

注 - 大写字母不能用于 OpenBoot PROM 的设备别名。如果由 **ldm add-vdisk** 命令添加的虚拟磁盘的名称包含大写字母，需将虚拟磁盘名的所有字母改成小写字母。然后，以 OpenBoot PROM 的环境变量 **boot-device** 进行设置。

11. 执行 **ldm list-bindings** 命令检查来宾域的配置。

```
# ldm list-bindings ldom1
```

12. 执行 **ldm bind-domain** 和 **ldm start-domain** 命令启动来宾域。

```
# ldm bind-domain ldom1
# ldm start-domain ldom1
```

5.2.5 配置恢复模式

使用 **svccfg** 命令和 **svcadm** 命令配置恢复模式。

```
# svccfg -s ldmd setprop ldmd/recovery_mode = astring: auto
# svcadm refresh ldmd
```

从 XCP 2230 开始，支持 Oracle VM Server for SPARC 3.1 所提供的恢复模式。有关恢复模式的详情，请参见所使用版本的《Oracle VM Server for SPARC Administration Guide》中的 "Handling Hardware Errors"。此外，关于恢复模式所需的修正，请参见所使用版本的《Oracle VM Server for SPARC Release Notes》。

5.2.6 保存逻辑域配置信息

完成所有来宾域配置后，保存逻辑域配置信息。

1. 执行 **ldm add-spconfig** 命令将逻辑域配置添加到 **XSCF**。
下面示例使用名称 "config2" 保存逻辑域配置信息。

```
# ldm add-spconfig config2
```

2. 执行 **ldm list-constraints** 命令将逻辑域配置信息保存到 **XML** 文件。
下面示例将逻辑域配置信息保存到 "config2.xml" 文件。

```
# ldm list-constraints -x > config2.xml
```

物理分区重新配置示例

本章介绍重新配置模块化架构配置中的物理分区的示例。

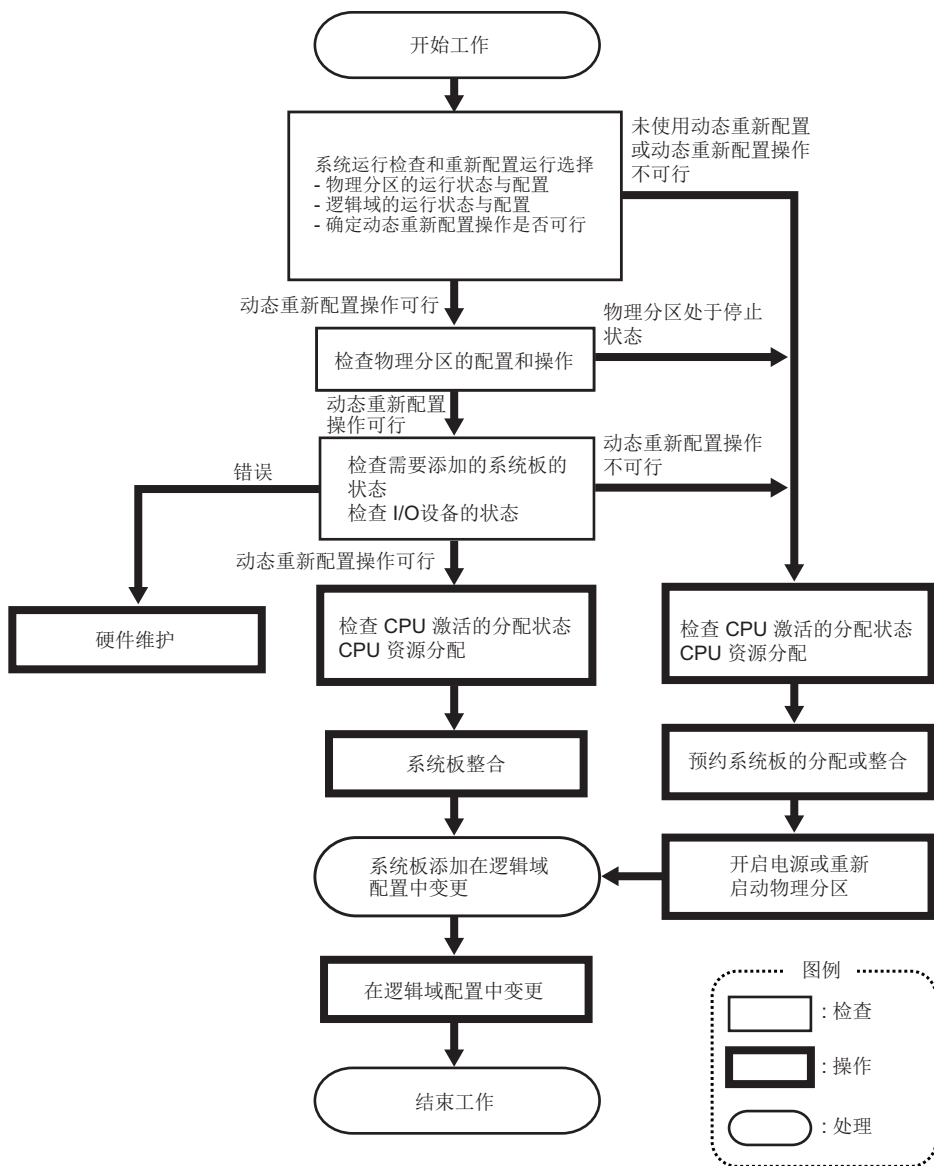
- [物理分区重新配置流程](#)
- [系统板添加示例](#)
- [删除系统板的操作示例](#)
- [移动系统板的操作示例](#)
- [更换系统板的操作示例](#)
- [变更物理分区编号注意事项](#)

6.1 物理分区重新配置流程

本节以流程图形式介绍物理分区重新配置操作，如系统板 (PSB) 添加、删除、移动和更换的流程。

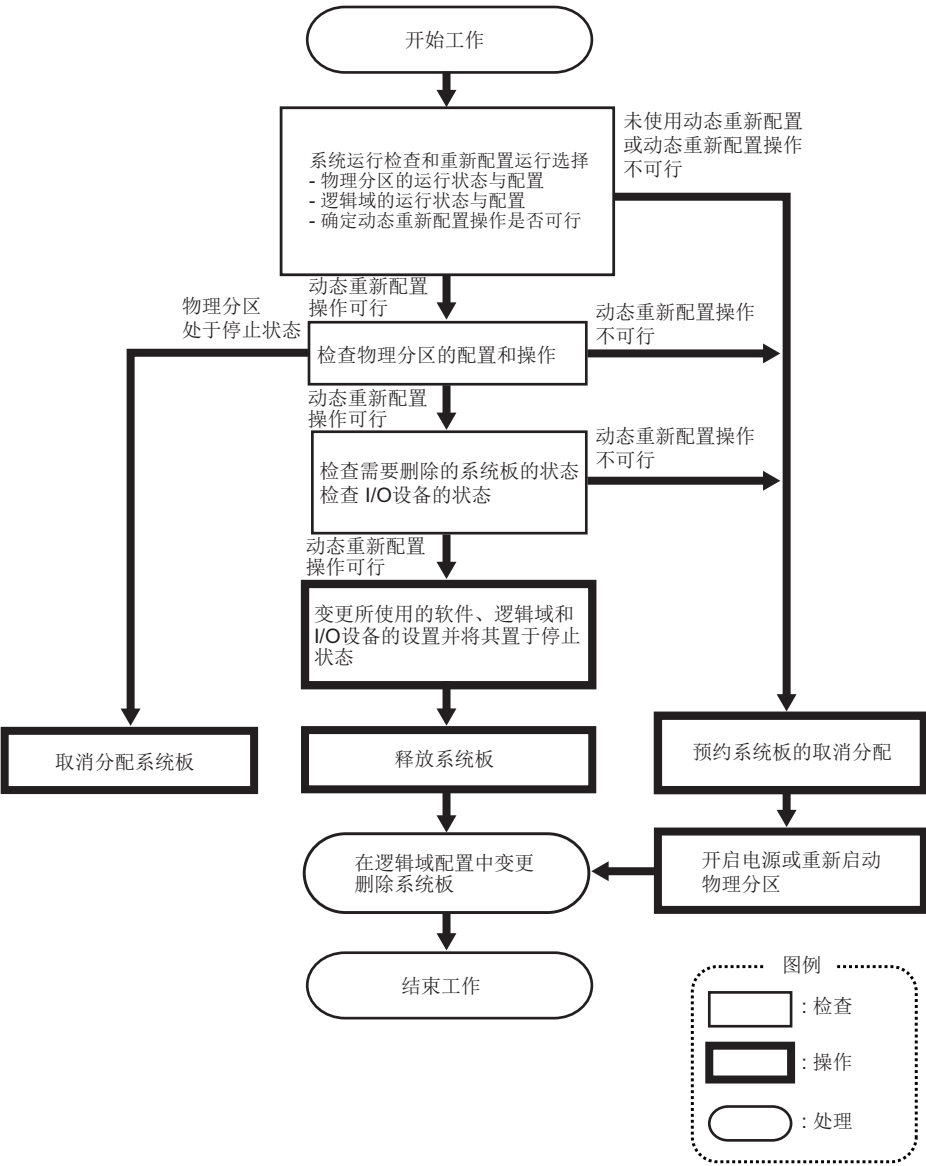
6.1.1 系统板添加流程

图 6-1 系统板添加流程



6.1.2 系统板删除流程

图 6-2 系统板删除流程



6.1.3 系统板移动流程

图 6-3 系统板移动流程 (1)

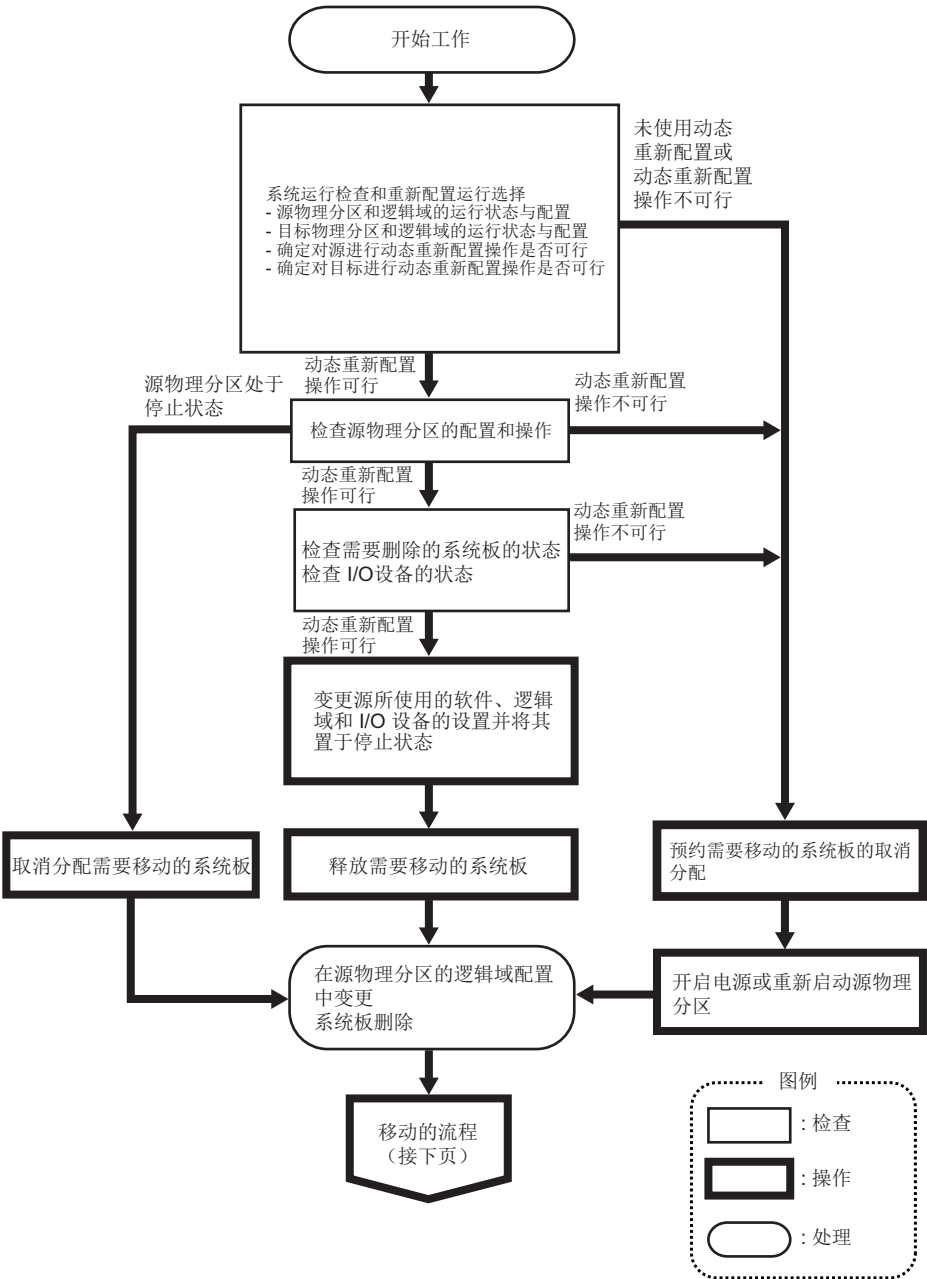
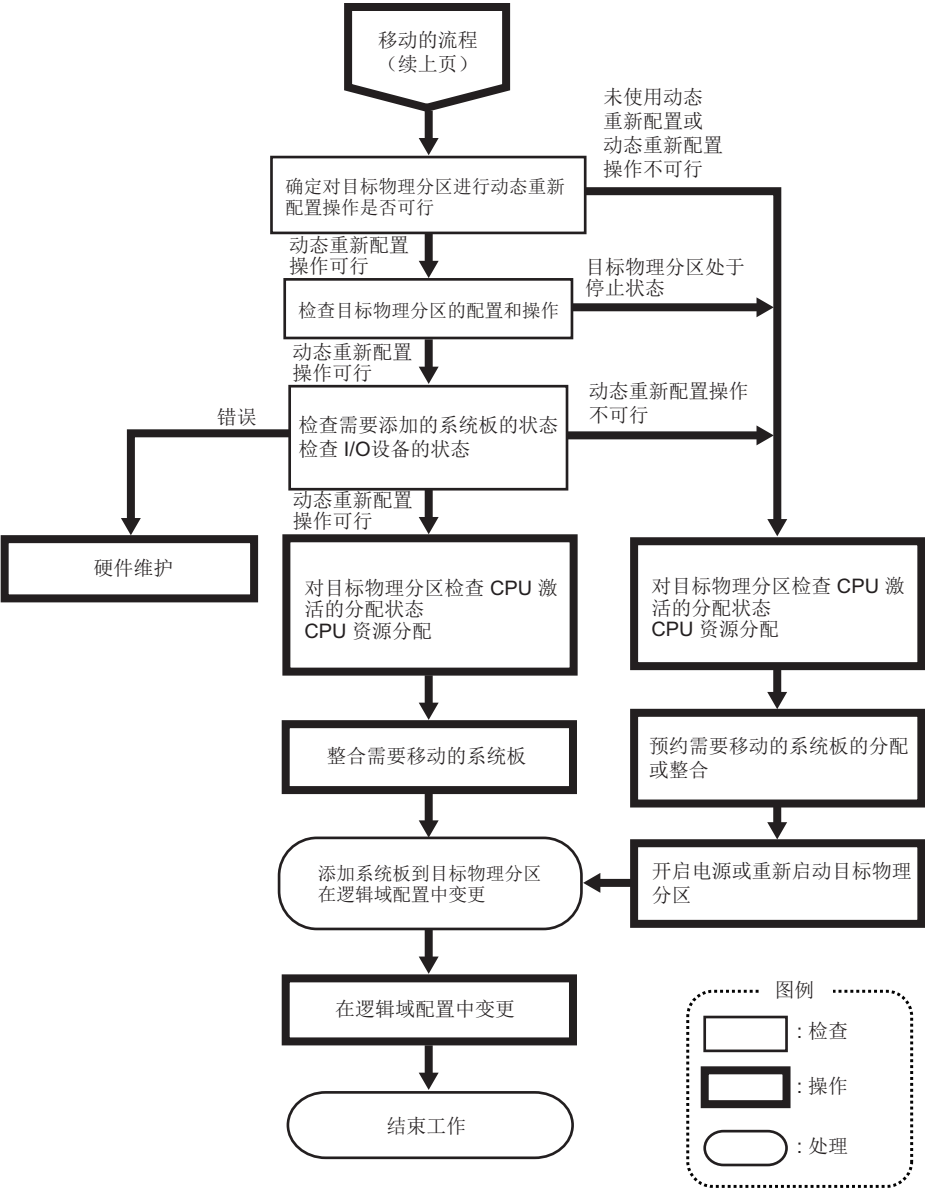


图 6-4 系统板移动流程 (2)



6.1.4 系统板更换流程

图 6-5 系统板更换流程 (1)

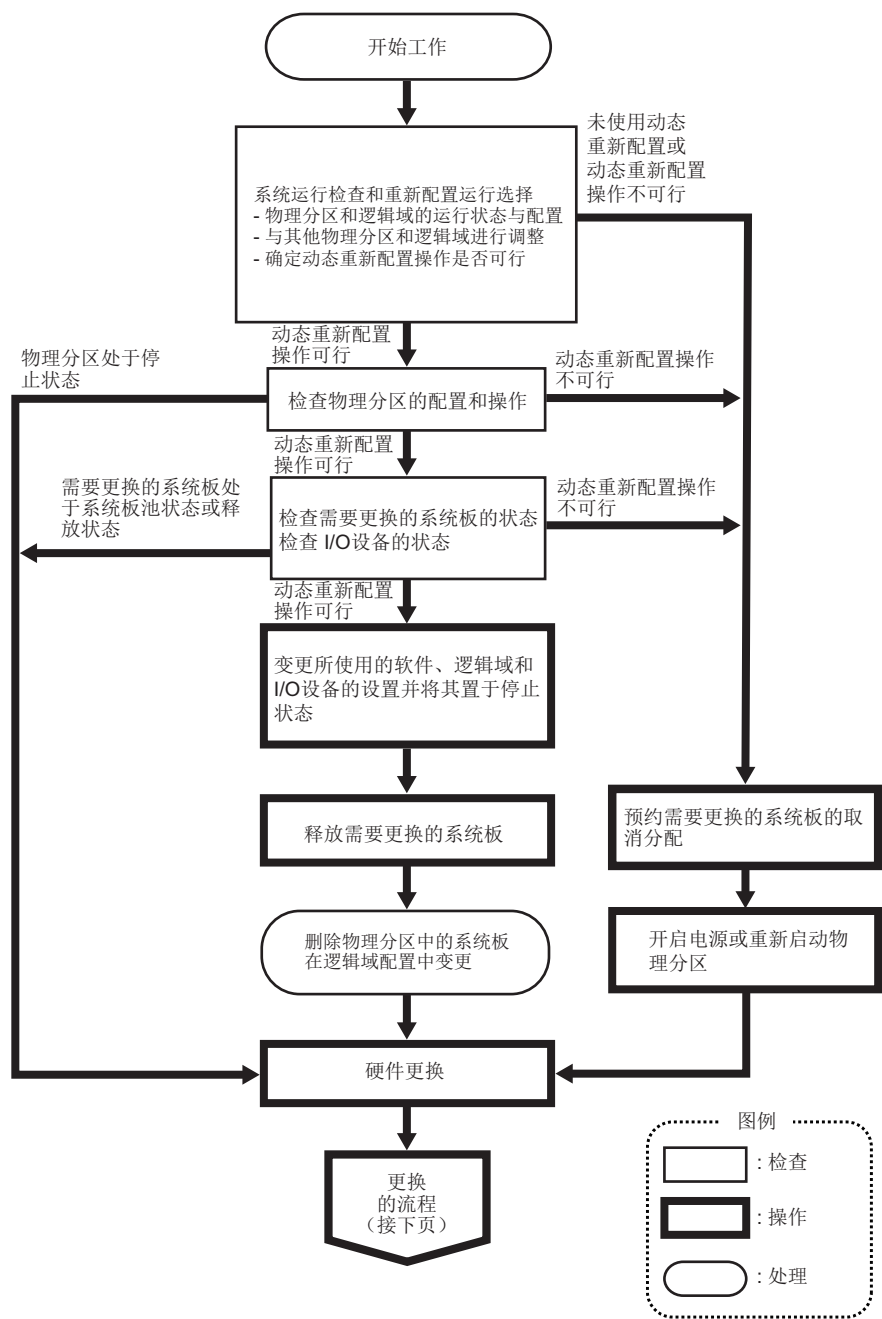
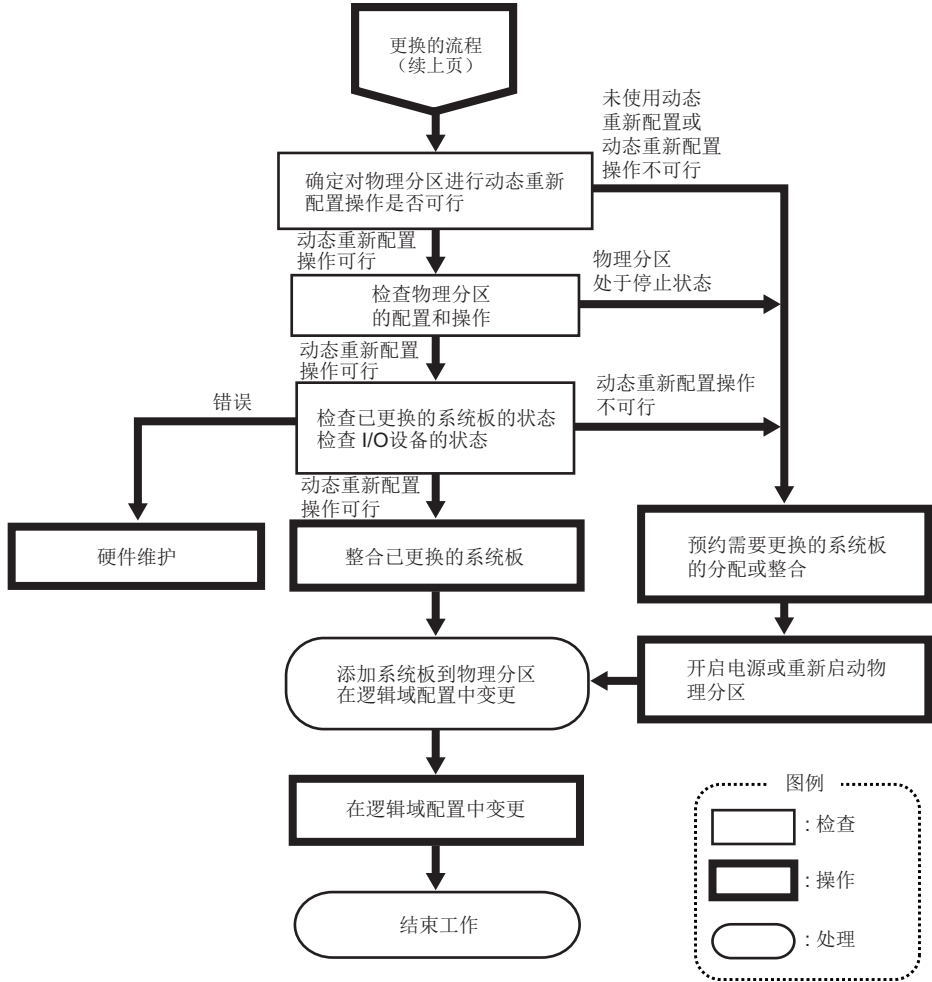


图 6-6 系统板更换流程 (2)

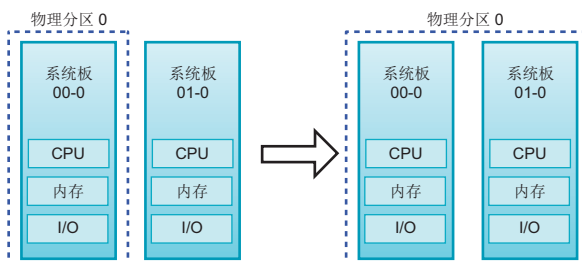


6.2

系统板添加示例

本节将介绍添加系统板 (PSB) 的操作示例，如 图 6-7 所示。

图 6-7 系统板添加示例



6.2.1 系统板分配示例

如果添加系统板 (PSB) 的物理分区处于停止状态, 使用指定 **-c assign** 选项的 **addboard** 命令可将 PSB 分配给物理分区。分配 PSB 后, 启动物理分区则将 PSB 整合到该物理分区。

这里介绍的 SPARC M10-4S 的操作示例使用 XSCF shell 命令。SPARC M12-2S 也使用相同的操作。

1. 登录 **XSCF**。
2. 添加 **PSB** 进行系统扩展时, 使用 **addfru** 命令将 **SPARC M10-4S** 整合到系统。

注- 有关系统扩展的详细信息, 请参见《SPARC M12-2S 安装指南》中的 "第8章 扩展模块化架构配置的系统" 或《SPARC M10-4S 安装指南》中的 "第9章 安装模块化架构配置的系统"。

3. 执行 **showpcl** 命令来检查当前物理分区配置信息 (**PPAR** 配置信息)。

```
XSCF> showpcl -p 0
PPAR-ID  LSB  PSB  Status
00                Powered Off
           00    00-0
```

4. 执行 **setpcl** 命令在 **PPAR** 配置信息中登录 **PSB**。
下面的示例中, 将 PSB 01-0 映射到物理分区 0 的逻辑系统板 (LSB 1)。

```
XSCF> setpcl -p 0 -a 1=01-0
```

5. 执行 **showpcl** 命令检查是否已经设置 **PPAR** 配置信息。

```
XSCF> showpcl -p 0
PPAR-ID  LSB  PSB  Status
00                Powered Off
           00    00-0
           01    01-0
```

6. 执行 **showcodusage** 命令显示 CPU 内核资源信息。

下面是显示 CPU 内核资源信息的示例。

如此处所示，该系统已安装 128 个 CPU 内核资源，已注册 128 个 CPU 激活，无 CPU 内核资源正在使用中，当前未在使用的 CPU 激活的数量是 128。

```
XSCF> showcodusage -p resource
Resource In Use Installed CoD Permitted Status
-----
PROC      0      128      128 OK: 128 cores
available

Note:
  Please confirm the value of the "In Use" by the ldm command of
  Oracle VM Server for SPARC.

  The XSCF may take up to 20 minutes to reflect the "In Use" of
  logical domains.
```

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。有关如何添加 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的“第5章 CPU 激活”。

7. 执行 **setcod** 命令将 CPU 内核资源分配给物理分区。

以下是将 128 个 CPU 内核资源分配到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c set 128
PROC Permits assigned for PPAR 0 : 0 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 - XCP 2250 和更早版本的 XSCF 固件不支持 -c add、-c delete 和 -c set 选项。按如下所示指定 setcod 命令的选项可交互式添加和删除 CPU 内核资源。

XSCF> **setcod -s cpu**

8. 执行 **showboards** 命令检查 PSB 状态。

在下面示例中，PSB 01-0 处于系统板池状态。

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test  Fault
-----
00-0 00(00)      Assigned    n    n    n    Passed Normal
01-0 SP          Available  n    n    n    Passed Normal
```

9. 执行指定 **-c assign** 选项的 **addboard** 命令来添加 PSB。

```
XSCF> addboard -c assign -p 0 01-0
```

10. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **addboard** 命令正常结束。

```
XSCF> showresult
0
```

11. 执行 **showboards** 命令检查 **PSB** 状态。

```
XSCF> showboards -p 0
PSB   PPAR-ID (LSB) Assignment   Pwr   Conn Conf Test      Fault
-----
00-0  00 (00)      Assigned   n     n    n    Passed Normal
01-0  00 (01)      Assigned   n     n    n    Passed Normal
```

12. 开启系统电源，启动物理分区。
13. 通过使用 **showpparstatus** 命令检查物理分区运行状态。
下一个示例在 [PPAR Status] 栏显示 "Running"，表明物理分区正在正确运行。

```
XSCF> showpparstatus -p 0
PPAR-ID PPAR Status
00      Running
```

14. 使用 **showboards** 命令检查 **PSB** 状态。
下一个示例在 PSB 01-0 的 [Conn] 和 [Conf] 栏显示 "y"，表明已经正确添加 PSB。

```
XSCF> showboards -p 0
PSB   PPAR-ID (LSB) Assignment   Pwr   Conn Conf Test      Fault
-----
00-0  00 (00)      Assigned   y     y    y    Passed Normal
01-0  00 (01)      Assigned   y     y    y    Passed Normal
```

15. 在逻辑域中分配物理分区资源来重新配置逻辑域。

6.2.2 系统板整合示例

如果添加系统板 (PSB) 的物理分区处于运行状态，使用指定 **-c configure** 选项的 **addboard** 命令可将 PSB 整合到物理分区。

执行 **addboard** 命令时，登录在物理分区配置信息（PPAR 配置信息）中的 PSB 或通过 **-c assign** 选项分配给物理分区的 PSB 被整合到物理分区。此时，需要开启物理分区的电源，并且控制域需要正在运行。

当您 将 PSB 整合到物理分区时，您可以指定是否启用将要整合的 PSB 添加资源的自动

分配功能。

- 不指定 **-m** 选项或指定 **-m bind=resource** 时

执行 `addboard` 命令之前，如果使用 `deleteboard` 命令已经将资源删除，则可恢复执行 `deleteboard` 命令之前存在的资源分配。然而，如果逻辑域配置在执行 `addboard` 命令之前发生变更，则最终的资源分配是按照逻辑域配置信息而获得。假设物理分区的逻辑域配置信息是出厂默认设置。则 PSB 所添加的资源将被分配给控制域。

- 指定 **-m bind=none** 时

PSB 添加的资源将是物理分区的可用资源。

注 - 指定 `-c configure` 选项的命令无法将 PSB 整合到禁用 PPAR DR 功能的物理分区。

注 - 使用 `addboard` 命令在 PSB 整合中，在完成 PSB 的硬件诊断后，目标 PSB 被整合到物理分区。因此，可能需要一些时间才可完成该命令。

这里介绍的 SPARC M10-4S 的操作示例使用 XSCF shell 命令和 Oracle VM Server for SPARC 的管理软件：Logical Domains Manager（逻辑域管理器）的 `ldm` 命令。SPARC M12-2S 也使用相同的操作。

1. 登录 **XSCF**。
2. 添加 **PSB** 进行系统扩展时，使用 `addfru` 命令将 **SPARC M10-4S** 整合到系统。

注 - 有关系统扩展的详细信息，请参见您的服务器的《安装指南》中的 "第9章 安装模块化架构配置的系统"。

3. 执行 `showpcl` 命令来检查当前物理分区配置信息（PPAR 配置信息）。

```
XSCF> showpcl -p 0
PPAR-ID   LSB   PSB   Status
00                               Running
          00   00-0
```

4. 执行 `setpcl` 命令在 PPAR 配置信息中登录 PSB。
下面的示例中，将 PSB 01-0 映射到物理分区 0 的逻辑系统板 (LSB 1)。

```
XSCF> setpcl -p 0 -a 1=01-0
```

5. 执行 `showpcl` 命令检查是否已经设置 PPAR 配置信息。

```
XSCF> showpcl -p 0
PPAR-ID   LSB   PSB   Status
00                               Running
          00   00-0
          01   01-0
```

6. 连接到物理分区的控制域控制台。

7. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。
下面的示例表明控制域和三个来宾域正在运行。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	14	8G	64%	2h 54m
guest0	active	-n----	5000	16	8G	42%	2h 54m
guest1	active	-n----	5001	16	8G	11%	2h 54m
guest2	active	-n----	5002	16	8G	7.3%	2h 54m

8. 返回 **XSCF shell**，然后执行 **showcodusage** 命令来显示 **CPU** 内核资源的信息。
下面是显示 **CPU** 内核资源信息的示例。
如此处所示，该系统已安装 128 个 **CPU** 内核资源，已注册 128 个 **CPU** 激活，64 个 **CPU** 内核资源正在使用中，当前未在使用的 **CPU** 激活的数量是 64。

```
XSCF> showcodusage -p resource
Resource In Use Installed CoD Permitted Status
-----
PROC          64          128          128 OK: 64 cores available

Note:
  Please confirm the value of the "In Use" by the ldm command of
  Oracle VM Server for SPARC.

  The XSCF may take up to 20 minutes to reflect the "In Use" of
  logical domains.
```

注 - 如果注册的 **CPU** 激活的数量对于将要使用的 **CPU** 的数量而言不足，则需购买 **CPU** 激活并添加 **CPU** 激活密钥。有关如何添加 **CPU** 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的“第5章 **CPU** 激活”。

9. 执行 **setcod** 命令将 **CPU** 内核资源分配给物理分区。
以下是将 64 个 **CPU** 内核资源添加到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c add 64
PROC Permits assigned for PPAR 0 :64 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 - XCP 2250 和更早版本的 XSCF 固件不支持 **-c add**、**-c delete** 和 **-c set** 选项。按如下所示指定 **setcod** 命令的选项可交互式添加和删除 **CPU** 内核资源。
XSCF> **setcod -s cpu**

10. 执行 **showboards** 命令检查 **PSB** 状态。

在下面示例中，PSB 01-0 处于系统板池状态。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal

11. 执行指定 **-c configure** 选项的 **addboard** 命令将 **PSB** 整合到物理分区。
在下面示例中，将 PSB 01-0 整合到物理分区 0。

```
XSCF> addboard -c configure -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
810.....840.....870.....900.....930.....960.....end
Connected PSB to PPAR.
Start configuring PSB to Logical Domains (LDoms) Manager. [1800sec]
0.....end
Configured PSB to Logical Domains (LDoms) Manager.
Operation has completed.
```

12. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **addboard** 命令正常结束。

```
XSCF> showresult
0
```

13. 执行 **showboards** 命令检查 **PSB** 状态。
下一个示例在 PSB 01-0 的 [Conn] 和 [Conf] 栏显示 "y"，表明已经正确添加 PSB。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

14. 在物理分区的控制域控制台上执行 **ldm list-domain** 命令，以便确认添加 **PSB** 后，逻辑域的运行状态是否变化。
15. 在逻辑域中分配物理分区资源来重新配置逻辑域。

6.2.3 预约系统板整合的操作示例

如果您无法进行与逻辑域的运行状态或配置、物理分区的运行状态或配置、或物理分区

的动态重新配置有关的操作，则预约系统板 (PSB) 到物理分区的整合。在系统运行过程中的适当时间重新启动物理分区将把 PSB 整合到物理分区。

这里介绍的 SPARC M10-4S 的操作示例使用 XSCF shell 命令。SPARC M12-2S 也使用相同的操作。

1. 登录 **XSCF**。
2. 添加 **PSB** 进行系统扩展时，使用 **addfru** 命令将 **SPARC M10-4S** 整合到系统。

注- 有关系统扩展的详细信息，请参见《SPARC M12-2S 安装指南》中的“第8章 扩展模块化架构配置的系统”或《SPARC M10-4S 安装指南》中的“第9章 安装模块化架构配置的系统”。

3. 执行 **showpctl** 命令来检查当前物理分区配置信息 (**PPAR** 配置信息)。

```
XSCF> showpctl -p 0
PPAR-ID   LSB   PSB   Status
00                               Running
          00   00-0
```

4. 执行 **setpctl** 命令在 **PPAR** 配置信息中登录 **PSB**。
下面的示例中，将 PSB 01-0 映射到物理分区 0 的逻辑系统板 (LSB 1)。

```
XSCF> setpctl -p 0 -a 1=01-0
```

5. 执行 **showpctl** 命令检查是否已经设置 **PPAR** 配置信息。

```
XSCF> showpctl -p 0
PPAR-ID LSB PSB Status
00                               Running
          00   00-0
          01   01-0
```

6. 执行 **showcodusage** 命令显示 **CPU** 内核资源信息。

下面是显示 CPU 内核资源信息的示例。

如此处所示，该系统已安装 128 个 CPU 内核资源，已注册 128 个 CPU 激活，64 个 CPU 内核资源正在使用中，当前未在使用的 CPU 激活的数量是 64。

```
XSCF> showcodusage -p resource
Resource In Use Installed CoD Permitted Status
-----
PROC          64        128          128 OK: 64 cores available
Note:
  Please confirm the value of the "In Use" by the ldm command of
  Oracle VM Server for SPARC.
  The XSCF may take up to 20 minutes to reflect the "In Use" of
  logical domains.
```

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。有关如何添加 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "第5章 CPU 激活"。

7. 执行 **setcod** 命令将 **CPU** 内核资源分配给物理分区。
以下是将 64 个 CPU 内核资源添加到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c add 64
PROC Permits assigned for PPAR 0 :64 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 - XCP 2250 和更早版本的 XSCF 固件不支持 -c add、-c delete 和 -c set 选项。按如下所示指定 setcod 命令的选项可交互式添加和删除 CPU 内核资源。
XSCF> **setcod -s cpu**

8. 执行 **showboards** 命令检查 **PSB** 状态。
在下面示例中，PSB 01-0 处于系统板池状态。

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test      Fault
-----
00-0 00(00)      Assigned    y    y    y    Passed Normal
01-0 SP          Available   n    n    n    Passed Normal
```

9. 执行指定 **-c reserve** 选项的 **addboard** 命令来预约 **PSB** 的添加。

```
XSCF> addboard -c reserve -p 0 01-0
```

10. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 addboard 命令正常结束。

```
XSCF> showresult
0
```

11. 执行 **showboards** 命令并指定 **-v** 选项检查 **PSB** 状态，然后确认 **PSB** 的添加是否预约。
由于下面示例在 [R] 栏显示 "*"，表明预约状态，PSB 正确预约了添加操作。

```
XSCF> showboards -v 01-0
PSB  R PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test      Fault
-----
01-0 * 00(01)      Assigned    n    n    n    Passed Normal
```

12. 重新启动物理分区。
13. 通过使用 **showpparstatus** 命令检查物理分区运行状态。
下一个示例在 [PPAR Status] 栏显示 "Running"，表明物理分区正在正确运行。

```
XSCF> showpparstatus -p 0
PPAR-ID PPAR Status
00      Running
```

14. 使用 **showboards** 命令检查 PSB 状态。
下一个示例在 PSB 01-0 的 [Conn] 和 [Conf] 栏显示 "y"，表明已经正确添加 PSB。

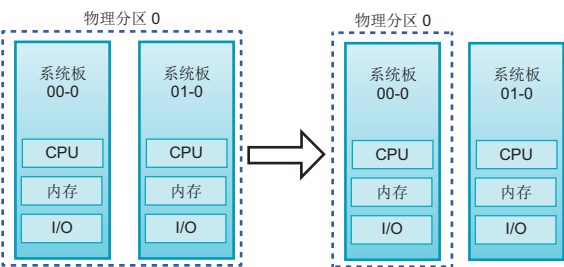
```
XSCF> showboards -p 0
PSB  PPAR-ID (LSB) Assignment  Pwr  Conn Conf Test      Fault
-----
00-0 00 (00)      Assigned  y    y    y    Passed  Normal
01-0 00 (01)      Assigned  y    y    y    Passed  Normal
```

15. 在逻辑域中分配物理分区资源来重新配置逻辑域。

6.3 删除系统板的操作示例

本节将介绍删除系统板 (PSB) 的操作示例，如 图 6-8 所示。

图 6-8 删除系统板的操作示例



6.3.1 删除系统板分配的操作示例

如果将要删除系统板 (PSB) 的物理分区处于停止状态，使用指定 **-c unassign** 选项的 **deleteboard** 命令将 PSB 从物理分区取消分配。PSB 从物理分区被取消分配，并进入系统板池状态。

这里介绍的操作示例使用 XSCF shell 命令。

1. 登录 **XSCF**。

2. 执行 **showpparstatus** 命令确认物理分区电源是否关闭。

```
XSCF> showpparstatus -p 0
PPAR-ID PPAR Status
00      Powered Off
```

3. 执行 **showboards** 命令检查需要删除的 **PSB** 的状态。

```
XSCF> showboards 01-0
PSB   PPAR-ID(LSB)  Assignment Pwr   Conn Conf Test      Fault
-----
01-0  00(01)         Assigned   n     n     n     Passed Normal
```

4. 执行指定 **-c unassign** 选项的 **deleteboard** 命令来删除 **PSB**。

```
XSCF> deleteboard -c unassign 01-0
```

5. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **deleteboard** 命令正常结束。

```
XSCF> showresult
0
```

6. 执行 **showboards** 命令检查 **PSB** 状态。
在下面示例中，PSB 01-0 处于系统板池状态。

```
XSCF> showboards 01-0
PSB   PPAR-ID(LSB)  Assignment Pwr   Conn Conf Test      Fault
-----
01-0  SP            Available n     n     n     Passed Normal
```

6.3.2 释放系统板的操作示例

如果将要释放系统板 (PSB) 的物理分区处于运行状态，使用指定 **-c disconnect** 或 **-c unassign** 选项的 **deleteboard** 命令将 PSB 从物理分区释放。

如果指定 **-c disconnect** 选项，PSB 将从物理分区释放，但仍然被分配给物理分区。因此，重新启动物理分区或执行 **addboard** 命令将已释放的 PSB 再次整合到物理分区。

如果通过指定 **-c unassign** 选项，从物理分区释放 PSB，PSB 将进入系统板池状态并可以被整合到和分配给其他物理分区。对于通过指定 **-c disconnect** 选项已经释放的 PSB，执行指定 **-c unassign** 选项的 **deleteboard** 命令，PSB 的状态将从已分配的状态变更到系统板池状态。

当将 PSB 从物理分区释放后，物理分区上的可用资源将用于移动使用将要释放的 PSB

资源的逻辑域。可以指定如何根据自由资源的状态在移动逻辑域时获取资源。若需采取上述操作，您可以使用 `deleteboard` 命令的 `-m` 选项。

- 如果未指定 `-m` 选项，或指定 `-m unbind=none` 选项

如果无法获取可用资源，或如果物理 I/O 分配未被提前释放，则使用 `deleteboard` 命令进行释放将失败。

- 如果指定 `-m unbind=resource`

使用 Oracle VM Server for SPARC 3.4 或更高版本，可以通过 `ldmd` 服务的 `ldmd/fj_ppar_dr_policy` 属性中设置的 PPAR DR 策略选择解除绑定资源的策略。Oracle VM Server for SPARC 的版本早于 Oracle VM Server for SPARC 3.4 时，此策略始终是 "targeted"。

有关策略的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统操作和管理指南》中的 "8.15 设置物理分区动态重新配置策略"。

`ldmd/fj_ppar_dr_policy = auto`

此 PPAR DR 策略设置指定自动使用最新的策略。对于使用 Oracle VM Server for SPARC 3.4 版本的系统，"auto" 设置与 "ratio" 设置的工作方式相同。"auto" 设置是默认策略。

`ldmd/fj_ppar_dr_policy = ratio`

如果物理分区中的所有逻辑域均运行 Oracle Solaris 11.3 或更高版本并且系统使用 XCP 2271 或更高版本，则此策略有效。否则，系统以与设置为 "targeted" 时的方式相同的方式操作。

如果将策略设置为 "ratio" 并且剩余的 PSB 没有足够的可用资源来移动将要释放的 PSB 中的资源，则该资源将被通过减少现有域中的资源而自动释放。在这种情况下，资源被尽可能成比例地从所有现有域中自动减少。由于资源忙，或由于其他原因，并不总是知晓确切的减少比例。基于最大努力进行比例的减少。

`ldmd/fj_ppar_dr_policy = targeted`

如果未能获取可用资源，将从使用目标 PSB 上的 CPU/内存的各逻辑域删除的资源获取可移动资源。策略定义了选择要从中删除资源的逻辑域的顺序。此策略定义了以下顺序：默认逻辑域（未指定任何内容）、指定 master 的逻辑域、I/O 域、根域和控制域。

如果从目标 PSB 未能获取可用资源，使用 `deleteboard` 命令的释放将失败。

- 如果指定 `-m unbind=shutdown`

此操作与指定 `-m unbind=resource` 的操作相同。如果未能获取更多的可用资源，将关闭源中的逻辑域。如需使用此选项，必须将 `ldmd/fj_ppar_dr_policy` 设置为 "targeted"。否则，逻辑域可能不会关闭。

注 - 对于禁用 PPAR DR 功能的物理分区，当物理分区正在运行时，无法通过指定 `-c disconnect` 或 `-c unassign` 选项释放 PSB。

注 - 当使用 `deleteboard` 命令释放 PSB 时，将从 Oracle Solaris 中释放 PSB 上的硬件资源。因此，需要一些时间才可完成该命令。

这里介绍的操作示例使用 XSCF shell 命令和 Oracle VM Server for SPARC 的管理软件：Logical Domains Manager（逻辑域管理器）的 `ldm` 命令。

1. 登录 XSCF。

2. 连接到物理分区的控制域控制台。
3. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。
- 下面的示例表明控制域、两个来宾域和一个根域正在运行。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	14	8G	64%	2h 54m
guest0	active	-n----	5000	16	8G	42%	2h 54m
guest1	active	-n----	5001	16	8G	11%	2h 54m
domain01	active	-n----	5002	16	8G	7.3%	2h 54m

4. 执行指定 **-a** 选项的 **ldm list-devices** 命令来检查资源的使用状况。
- 在下面的示例中，指定 **-a** 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。

# ldm list-devices -a		
CORE		
ID	%FREE	CPUSET
0	0	(0, 1)
4	0	(8, 9)
8	0	(16, 17)
(省略)		
184	0	(368, 369)
188	100	(376, 377)
512	100	(1024, 1025)
516	100	(1032, 1033)
520	100	(1040, 1041)
524	100	(1048, 1049)
(省略)		
VCPU		
PID	%FREE	PM
0	0	no
1	0	no
8	0	no
9	0	no
(省略)		
369	0	no
376	100	---
377	100	---
1024	100	---
1025	100	---
1032	100	---
1033	100	---
(省略)		

5. 执行 **ldm list-io** 命令检查 I/O 设备的使用状况。
- 在下面的示例中，**primary** 是一个控制域，**domain01** 是一个使用 I/O 设备的根域。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----

PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
PCIE8	BUS	PCIE8	domain01	IOV
PCIE9	BUS	PCIE9	domain01	IOV
PCIE10	BUS	PCIE10	domain01	IOV
PCIE11	BUS	PCIE11	domain01	IOV
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
/BB0/CMUL/NET2	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/PCI0	PCIE	PCIE1	primary	EMP
/BB0/PCI1	PCIE	PCIE1	primary	EMP
(省略)				

6. 在由根域提供的服务中，释放需要删除的 **PSB** 上的所有 **I/O** 设备。
 在下面示例中，停止根域 domain01，在 domain01 所拥有的 PCIe 根合成体中，释放与需要删除的 PSB 有关的根合成体。

注 - 动态重新配置时，不要求使用 ldm stop-domain 命令停止根域。请从 I/O 设备释放开始操作。

```
# ldm stop-domain domain01
# ldm remove-io PCIE11 domain01
# ldm remove-io PCIE10 domain01
# ldm remove-io PCIE9 domain01
# ldm remove-io PCIE8 domain01
# ldm unbind-domain domain01
```

7. 再次执行 **ldm list-io** 命令检查 **I/O** 设备的使用状况。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10		
PCIE11	BUS	PCIE11		
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
/BB0/CMUL/NET2	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/PCI0	PCIE	PCIE1	primary	EMP
/BB0/PCI1	PCIE	PCIE1	primary	EMP
(省略)				

8. 执行指定 **-c unassign** 选项的 **deleteboard** 命令，将 **PSB** 从物理分区释放。

在下面示例中，PSB 01-0 从物理分区释放并进入系统板池状态。

```
XSCF> deleteboard -c unassign 01-0
PSB#01-0 will be unassigned from PPAR immediately. Continue?[y|n] :y
Start unconfigure preparation of PSB. [1200sec]
0.end
Unconfigure preparation of PSB has completed.
Start unconfiguring PSB from PPAR. [43200sec]
0..... 30.end
Unconfigured PSB from PPAR.
PSB power off sequence started. [1200sec]
0..... 30..... 60..... 90.....120.....150.end
Operation has completed.
```

9. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **deleteboard** 命令正常结束。

```
XSCF> showresult
0
```

10. 执行 **showboards** 命令检查 **PSB** 状态。
在下面示例中，PSB 01-0 处于系统板池状态。

```
XSCF> showboards -p 0
```

PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal

11. 在物理分区的控制域控制台上执行 **ldm list-domain** 命令，以便确认删除 **PSB** 后，逻辑域的运行状态是否变化。
12. 当您移除模块化架构时，请在主 **XSCF** 上执行 **initbb** 命令，以便从系统释放目标 **SPARC M12-2S/M10-4S**。

注- 有关系统缩减的详细信息，请参见《SPARC M12-2S 安装指南》中的“第9章 缩减模块化架构配置的系统”或《SPARC M10-4S 安装指南》中的“第10章 移除模块化架构配置的系统”。

6.3.3 预约系统板取消分配的操作示例

如果您无法进行与逻辑域的运行状态或配置、物理分区的运行状态或配置、或物理分区的动态重新配置有关的操作，则预约从物理分区中系统板 (PSB) 的取消分配。通过在系统运行过程中的适当时间停止物理分区，PSB 将从物理分区取消分配。PSB 进入系统板池状态。

这里介绍的操作示例使用 XSCF shell 命令。

1. 执行 **showpparstatus** 命令来检查物理分区运行状态。

```
XSCF> showpparstatus -p 0
PPAR-ID PPAR Status
00      Running
```

2. 执行 **showboards** 命令检查需要删除的 **PSB** 的状态。

```
XSCF> showboards 01-0
PSB  PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test      Fault
-----
01-0 00(01)      Assigned    y    y    y    Passed Normal
```

3. 执行指定 **-c reserve** 选项的 **deleteboard** 命令来预约 **PSB** 的删除。

```
XSCF> deleteboard -c reserve 01-0
```

4. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **deleteboard** 命令正常结束。

```
XSCF> showresult
0
```

5. 执行 **showboards** 命令并指定 **-v** 选项检查 **PSB** 状态，然后确认 **PSB** 的删除是否预约。
由于下面示例在 [R] 栏显示 **"*"**，表明预约状态，正确预约了 **PSB** 的删除。

```
XSCF> showboards -v 01-0
PSB  R PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test      Fault
-----
01-0 * 00(01)      Assigned    y    y    y    Passed Normal
```

6. 重新启动物理分区。

7. 通过使用 **showpparstatus** 命令检查物理分区运行状态。
下一个示例在 [PPAR Status] 栏显示 **"Running"**，表明物理分区正在正确运行。

```
XSCF> showpparstatus -p 0
PPAR-ID PPAR Status
00      Running
```

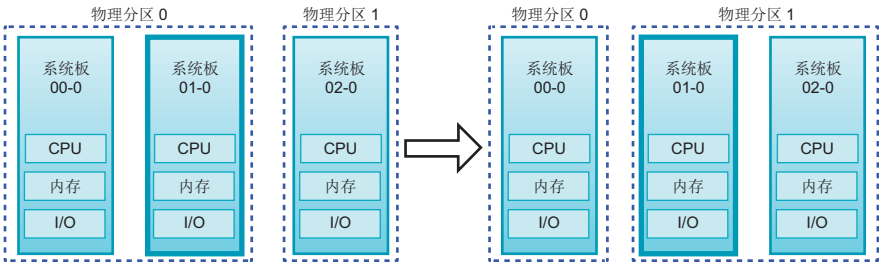
8. 执行 **showboards** 命令检查 **PSB** 状态。
在下面示例中，**PSB 01-0** 处于系统板池状态。

XSCF> showboards -p 0								
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault	
00-0	00(00)	Assigned	y	y	y	Passed	Normal	
01-0	SP	Available	n	n	n	Passed	Normal	

6.4 移动系统板的操作示例

本节将介绍在物理分区之间移动系统板 (PSB) 的操作示例，如 图 6-9所示。

图 6-9 移动系统板的操作示例



物理分区之间移动 PSB 的操作包括 PSB 取消分配和整合的组合操作。

如果动态重新配置操作由于源物理分区操作状态而无法进行，则通过取消分配 PSB 或预约 PSB 的取消分配从源物理分区中删除 PSB。当目标物理分区无法进行动态重新配置操作时，通过分配 PSB 或预约 PSB 到目标物理分区的分配来添加 PSB。

注 - 假设禁用物理分区的 PPAR DR 功能, 并且物理分区正在运行。在这种情况下，您无法通过指定 -c disconnect 或 -c unassign 选项释放 PSB，或通过指定 -c configure 选项整合 PSB。

此处的操作示例假定源物理分区和目标物理分区允许动态重新配置操作。这里介绍的 SPARC M10-4S 的操作示例使用 XSCF shell 命令和 Oracle VM Server for SPARC 的管理软件：Logical Domains Manager（逻辑域管理器）的 ldm 命令。SPARC M12-2S 也使用相同的操作。

- 1. 登录 XSCF。
- 2. 连接到物理分区的控制域控制台。
- 3. 执行 ldm list-domain 命令检查逻辑域的运行状态。
下面的示例表明控制域、两个来宾域和一个根域正在运行。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	14	8G	64%	2h 54m
guest0	active	-n----	5000	16	8G	42%	2h 54m
guest1	active	-n----	5001	16	8G	11%	2h 54m

4. 执行指定 **-a** 选项的 **ldm list-devices** 命令来检查资源的使用状况。
- 在下面的示例中，指定 **-a** 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。

```
# ldm list-devices -a
CORE
  ID      %FREE    CPUSSET
  0        0      (0, 1)
  4        0      (8, 9)
  8        0     (16, 17)
  (省略)
 184       0     (368, 369)
 188      100    (376, 377)
 512      100   (1024, 1025)
 516      100   (1032, 1033)
 520      100   (1040, 1041)
 524      100   (1048, 1049)
  (省略)
VCPU
  PID      %FREE    PM
  0        0      no
  1        0      no
  8        0      no
  9        0      no
  (省略)
 369       0      no
 376      100     ---
 377      100     ---
 1024     100     ---
 1025     100     ---
 1032     100     ---
 1033     100     ---
  (省略)
```

5. 执行 **ldm list-io** 命令检查 I/O 设备的使用状况。
- 在下面的示例中，**primary** 是一个控制域，**domain01** 是一个使用 I/O 设备的根域。

```
# ldm list-io
NAME                                     TYPE    BUS      DOMAIN    STATUS
----                                     -
PCIE0                                  BUS     PCIE0    primary   IOV
PCIE1                                  BUS     PCIE1    primary   IOV
PCIE2                                  BUS     PCIE2    primary   IOV
PCIE3                                  BUS     PCIE3    primary   IOV
PCIE8                                  BUS     PCIE8    domain01  IOV
PCIE9                                  BUS     PCIE9    domain01  IOV
PCIE10                                 BUS     PCIE10   domain01  IOV
PCIE11                                 BUS     PCIE11   domain01  IOV
/BB0/CMUL/NET0                         PCIE    PCIE0    primary   OCC
/BB0/CMUL/NET2                         PCIE    PCIE0    primary   OCC
```

/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/PCIO	PCIE	PCIE1	primary	EMP
/BB0/PCI1	PCIE	PCIE1	primary	EMP
(省略)				

6. 在由根域提供的服务中，释放需要删除的 **PSB** 上的所有 **I/O** 设备。
在下面示例中，停止根域 **domain01**，在 **domain01** 所拥有的 **PCIe** 根合成体中，释放与需要删除的 **PSB** 有关的根合成体。

注 - 动态重新配置时，不要求使用 **ldm stop-domain** 命令停止根域。请从 **I/O** 设备释放开始操作。

```
# ldm stop-domain domain01
# ldm remove-io PCIE11 domain01
# ldm remove-io PCIE10 domain01
# ldm remove-io PCIE9 domain01
# ldm remove-io PCIE8 domain01
# ldm unbind-domain domain01
```

7. 再次执行 **ldm list-io** 命令检查 **I/O** 设备的使用状况。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10		
PCIE11	BUS	PCIE11		
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
/BB0/CMUL/NET2	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/PCIO	PCIE	PCIE1	primary	EMP
/BB0/PCI1	PCIE	PCIE1	primary	EMP
(省略)				

8. 执行指定 **-c unassign** 选项的 **deleteboard** 命令，将 **PSB** 从物理分区释放。
在下面示例中，**PSB 01-0** 从物理分区释放并进入系统板池状态。

```
XSCF> deleteboard -c unassign 01-0
```

9. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。
在下面示例中，返回结束状态 **0**，表明 **deleteboard** 命令正常结束。

```
XSCF> showresult
0
```

10. 执行 **showboards** 命令检查 **PSB** 状态。
在下面示例中，PSB 01-0 处于系统板池状态。

```
XSCF> showboards -a
PSB  PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test  Fault
-----
00-0 00(00)      Assigned   y    y    y    Passed Normal
01-0 SP          Available n    n    n    Passed Normal
02-0 01(00)      Assigned   y    y    y    Passed Normal
```

11. 在源物理分区的控制域控制台上执行 **ldm list-domain** 命令，以便确认删除 **PSB** 后，逻辑域的运行状态是否变化。
12. 执行 **showpctl** 命令来检查目标物理分区的当前物理分区配置信息（**PPAR** 配置信息）。

```
XSCF> showpctl -p 1
PPAR-ID  LSB   PSB   Status
01                Running
          00    00-0
```

13. 执行 **setpctl** 命令在目标物理分区的 **PPAR** 配置信息中登录 **PSB**。
下面的示例中，将 PSB 01-0 映射到物理分区 1 的逻辑系统板 (LSB 1)。

```
XSCF> setpctl -p 1 -a 1=01-0
```

14. 执行 **showpctl** 命令检查是否已经设置 **PPAR** 配置信息。

```
XSCF> showpctl -p 1
PPAR-ID  LSB   PSB   Status
01                Running
          00    02-0
          01    01-0
```

15. 连接到物理分区的控制域控制台。
16. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。
下面的示例表明控制域和三个来宾域正在运行。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    14      8G        64%     2h 54m
guest0        active    -n----   5000    16      8G        42%     2h 54m
guest1        active    -n----   5001    16      8G        11%     2h 54m
guest2        active    -n----   5002    16      8G        7.3%    2h 54m
```

17. 返回 **XSCF shell**，并执行 **showcod** 命令检查分配给物理分区的 **CPU** 内核资源。
下面是显示 CPU 内核资源设置信息的示例。

如示例所示，该系统安装有 192 个 CPU 内核资源，物理分区 0 中注册 128 个 CPU 激活，物理分区 1 中注册 64 个 CPU 激活。

```
XSCF> showcod -v -s cpu
PROC Permits installed : 192 cores
PROC Permits assigned for PPAR 0: 128 [Permanent 128cores]
PROC Permits assigned for PPAR 1: 64 [Permanent 64cores]
PROC Permits assigned for PPAR 2: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 3: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 4: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 5: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 6: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 7: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 8: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 9: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 10: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 11: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 12: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 13: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 14: 0 [Permanent 0cores]
PROC Permits assigned for PPAR 15: 0 [Permanent 0cores]
```

18. 执行 **showcodusage** 命令查看 CPU 内核资源信息。
结果表明安装状态如下。

[物理分区 0]

- CPU 内核资源的数量: 64
- 已注册的 CPU 激活的数量: 128
- 处于使用状态的 CPU 内核资源的数量: 64

[物理分区 1]

- CPU 内核资源的数量: 64
- 已注册的 CPU 激活的数量: 64
- 处于使用状态的 CPU 内核资源的数量: 64

[整个系统]

- 未处于使用状态（未使用）的 CPU 内核资源的数量: 64

```
XSCF> showcodusage -p ppar
PPAR-ID/Resource In Use Installed Assigned
-----
0 - PROC          64          64      128 cores
1 - PROC          64          64       64 cores
Unused - PROC      0           0       64 cores
```

Note:

Please confirm the value of the "In Use" by the ldm command of

Oracle VM Server for SPARC.

The XSCF may take up to 20 minutes to reflect the "In Use" of logical domains.

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。有关如何添加 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的“第5章 CPU 激活”。

19. 执行 **setcod** 命令将 CPU 内核资源分配给目的地物理分区。

下面的示例通过移除 64 个 CPU 激活并将其添加到物理分区 1 从而减少物理分区 0 中分配的 CPU 激活数。

```
XSCF> setcod -p 0 -s cpu -c del 64
PROC Permits assigned for PPAR 0 : 128 -> 64

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.

XSCF> setcod -p 1 -s cpu -c add 64
PROC Permits assigned for PPAR 1 : 64 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 - XCP 2250 和更早版本的 XSCF 固件不支持 **-c add**、**-c delete** 和 **-c set** 选项。按如下所示指定 **setcod** 命令的选项可交互式添加和删除 CPU 内核资源。

XSCF> **setcod -s cpu**

20. 执行 **showboards** 命令检查 PSB 状态。

```
XSCF> showboards -a
```

PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal
02-0	01(00)	Assigned	y	y	y	Passed	Normal

21. 执行指定 **-c configure** 选项的 **addboard** 命令将 PSB 整合到物理分区。

在下面示例中，将 PSB 01-0 整合到物理分区 1。

```
XSCF> addboard -c configure -p 1 01-0
```

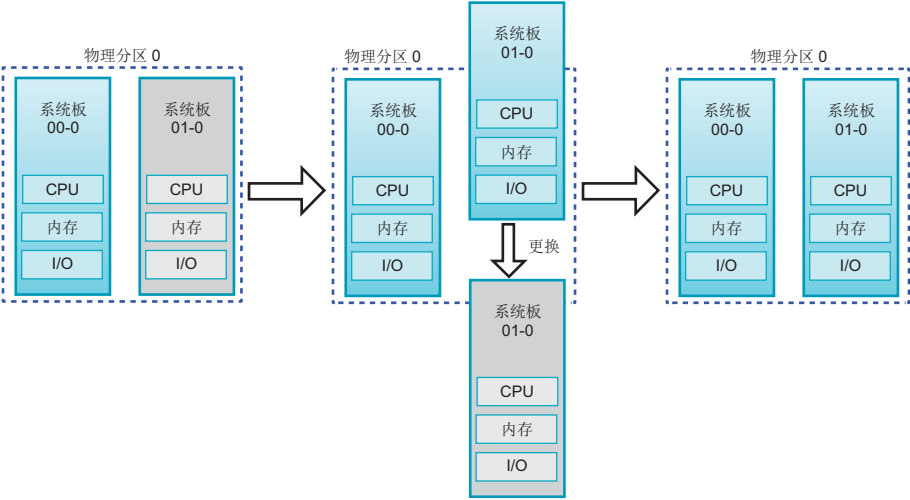

22. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **addboard** 命令正常结束。
- ```
XSCF> showresult
0
```
23. 执行 **showboards** 命令检查 **PSB** 状态。  
下一个示例在 PSB 01-0 的 [Conn] 和 [Conf] 栏显示 "y"，表明已经正确添加 PSB。
- ```
XSCF> showboards -a
```

PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	01(01)	Assigned	y	y	y	Passed	Normal
02-0	01(00)	Assigned	y	y	y	Passed	Normal
24. 在目标物理分区的控制域控制台上执行 **ldm list-domain** 命令，以便确认添加 **PSB** 后，逻辑域的运行状态是否变化。
25. 在逻辑域中分配物理分区资源来重新配置逻辑域。

6.5 更换系统板的操作示例

本节将介绍更换 (PSB) 的操作示例，如 图 6-10所示。

图 6-10 更换系统板的操作示例



更换 PSB 的操作包括 PSB 释放和整合的组合操作。当由于物理分区的运行状态而无法执行动态重新配置操作时，请在关闭物理分区的电源后更换 PSB。

注 - 假设禁用物理分区的 PPAR DR 功能, 并且物理分区正在运行。在这种情况下, 您无法通过指定 `-c disconnect` 或 `-c unassign` 选项释放 PSB, 或通过指定 `-c configure` 选项整合 PSB。

此处的操作示例假定物理分区允许动态重新配置操作。这里介绍的 SPARC M10-4S 的操作示例使用 XSCF shell 命令和 Oracle VM Server for SPARC 的管理软件: Logical Domains Manager (逻辑域管理器) 的 `ldm` 命令。SPARC M12-2S 也使用相同的操作。

- 1. 登录 **XSCF**。
- 2. 执行 **showbbstatus** 命令来确认需要更换的 **PSB** 中的 **XSCF** 不是主 **XSCF**。

注 - 如果需要更换的 PSB 的 XSCF 是主 XSCF, 请执行 `switchscf` 命令将 XSCF 从激活状态切换到待机状态。释放 PSB 前, 请确认 XSCF 是否已经切换并重新启动。

- 3. 连接到物理分区的控制域控制台。
- 4. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。
下面的示例表明控制域、两个来宾域和一个根域正在运行。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	14	8G	64%	2h 54m
guest0	active	-n----	5000	16	8G	42%	2h 54m
guest1	active	-n----	5001	16	8G	11%	2h 54m
domain01	active	-n----	5002	16	8G	7.3%	2h 54m

- 5. 执行指定 **-a** 选项的 **ldm list-devices** 命令来检查资源的使用状况。
在下面的示例中, 指定 **-a** 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。

```
# ldm list-devices -a

CORE
    ID          %FREE    CPUSET
    0           0        (0, 1)
    4           0        (8, 9)
    8           0        (16, 17)
(省略)
    184         0        (368, 369)
    188         100      (376, 377)
    512         100      (1024, 1025)
    516         100      (1032, 1033)
    520         100      (1040, 1041)
    524         100      (1048, 1049)
(省略)

VCPU
    PID         %FREE    PM
    0           0        no
    1           0        no
    8           0        no
    9           0        no
```

```
(省略)
369      0      no
376      100    ---
377      100    ---
1024     100    ---
1025     100    ---
1032     100    ---
1033     100    ---
(省略)
```

6. 执行 `ldm list-io` 命令检查 I/O 设备的使用状况。

在下面的示例中，`primary` 是一个控制域，`domain01` 是一个使用 I/O 设备的根域。

```
# ldm list-io
NAME                                     TYPE      BUS        DOMAIN     STATUS
----                                     -
PCIE0                                  BUS       PCIE0      primary    IOV
PCIE1                                  BUS       PCIE1      primary    IOV
PCIE2                                  BUS       PCIE2      primary    IOV
PCIE3                                  BUS       PCIE3      primary    IOV
PCIE8                                  BUS       PCIE8      domain01   IOV
PCIE9                                  BUS       PCIE9      domain01   IOV
PCIE10                                 BUS       PCIE10     domain01   IOV
PCIE11                                 BUS       PCIE11     domain01   IOV
/BB0/CMUL/NET0                         PCIE      PCIE0      primary    OCC
/BB0/CMUL/NET2                         PCIE      PCIE0      primary    OCC
/BB0/CMUL/SASHBA                       PCIE      PCIE0      primary    OCC
/BB0/PCI0                              PCIE      PCIE1      primary    EMP
/BB0/PCI1                              PCIE      PCIE1      primary    EMP
(省略)
```

7. 在由根域提供的服务中，释放需要删除的 PSB 上的所有 I/O 设备。

在下面示例中，停止根域 `domain01`，在 `domain01` 所拥有的 PCIe 根合成体中，释放与需要删除的 PSB 有关的根合成体。

注 - 动态重新配置时，不要求使用 `ldm stop-domain` 命令停止根域。请从 I/O 设备释放开始操作。

```
# ldm stop-domain domain01
# ldm remove-io PCIE11 domain01
# ldm remove-io PCIE10 domain01
# ldm remove-io PCIE9 domain01
# ldm remove-io PCIE8 domain01
# ldm unbind-domain domain01
```

8. 再次执行 `ldm list-io` 命令检查 I/O 设备的使用状况。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10		
PCIE11	BUS	PCIE11		
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
/BB0/CMUL/NET2	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/PCIO	PCIE	PCIE1	primary	EMP
/BB0/PCI1	PCIE	PCIE1	primary	EMP
(省略)				

9. 执行指定 **-c disconnect** 选项的 **deleteboard** 命令，将 **PSB** 从物理分区释放。
在下面示例中，PSB 01-0 从物理分区释放并分配给物理分区 0。

```
XSCF> deleteboard -c disconnect 01-0
```

10. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **deleteboard** 命令正常结束。

```
XSCF> showresult
0
```

11. 执行 **showboards** 命令检查 **PSB** 状态。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	---	---	---	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

12. 执行 **replacefru** 命令来更换 **PSB**。

```
XSCF> replacefru
```

注- 有关使用 **replacefru** 命令更换 PSB 的详情，请参见您的服务器的《Service Manual》。

13. 执行 **showboards** 命令检查 **PSB** 状态。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

14. 执行指定 **-c configure** 选项的 **addboard** 命令将 **PSB** 整合到物理分区。
在下面示例中，将 PSB 01-0 整合到物理分区 0。

```
XSCF> addboard -c configure -p 0 01-0
```

注 - 当您执行 **addboard** 命令时，如果省略 **-m** 选项或将 **-m** 选项指定为 **bind=resource**，资源分配可能无法恢复到你执行 **deleteboard** 命令之前的状态。
如果更换前 PSB 资源（如 CPU 和内存）的量与更换后的量不同，将无法将资源分配恢复到你初始状态。
无法恢复到其初始状态的资源将成为可用资源。此种情况下，使用 **ldm** 命令来再次将资源分配给逻辑域。

15. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。
在下面示例中，返回结束状态 0，表明 **addboard** 命令正常结束。

```
XSCF> showresult
0
```

16. 执行 **showboards** 命令检查 **PSB** 状态。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

17. 在物理分区的控制域控制台上执行 **ldm list-domain** 命令，以便确认添加 **PSB** 后，逻辑域的运行状态是否变化。
18. 启动根域恢复其服务。



6.6 变更物理分区编号注意事项

您可以将一个已配置的物理分区的物理分区编号变更为其其他编号。变更编号时请考虑以下问题：

- 必须重新设置物理分区配置信息。
如果您已经变更了物理分区编号，则需要设置新物理分区的配置信息。有关设置物

理分区配置信息的详情，请参见 "[3.1.12 设置物理分区配置信息](#)" 和 "[第4章 物理分区配置示例](#)"。

- 必须重新分配 CPU 激活。

变更物理分区编号后，CPU 激活仍分配给原来的物理分区编号。因此，您需要释放 CPU 激活分配，然后通过指定新物理分区编号进行重新分配。

- 主机 ID 和以太网 (MAC) 地址已变更。

当物理分区编号变更时，物理分区特定的主机 ID 和以太网 (MAC) 地址也发生变更。也必须考虑到这样的事实：主机 ID 和以太网 (MAC) 地址可能用于许可证认证和 Oracle Solaris ZFS 文件系统 and 应用程序的设定信息。

迁移来宾域

本章介绍如何使用实时迁移功能将来宾域迁移到不同的物理分区。

- 概述
- 迁移来宾域
- 来宾域迁移示例

7.1 概述

实时迁移功能可以在保持来宾域运行的同时在不同物理分区之间迁移来宾域。在以下情况下使用实时迁移功能。

- 调整物理分区之间的负荷
通过将来宾域从高负荷物理分区迁移到低负荷物理分区，对整个系统的负荷进行调整。
- 在来宾域运行时维护硬件
如需将系统板从物理分区中释放以进行维护，需将物理分区中正在运行的来宾域迁移到不同的物理分区。

执行实时迁移时，源物理分区和目标物理分区的Logical Domains Manager（逻辑域管理器）进行通讯来执行以下处理：

- (1) 在目标物理分区中，配置与源物理分区具有相同配置的来宾域。
- (2) 将信息从源来宾域复制到目标来宾域。源来宾域上的执行继续。
- (3) 复制过程结束后，删除源逻辑域，开始在目标逻辑域上执行。

逻辑域可以在同一个系统及不同系统间迁移。图 7-1 显示了在同一个系统内的迁移。图 7-2 显示了在不同系统间的迁移。

图 7-1 同一个系统内的迁移

系统

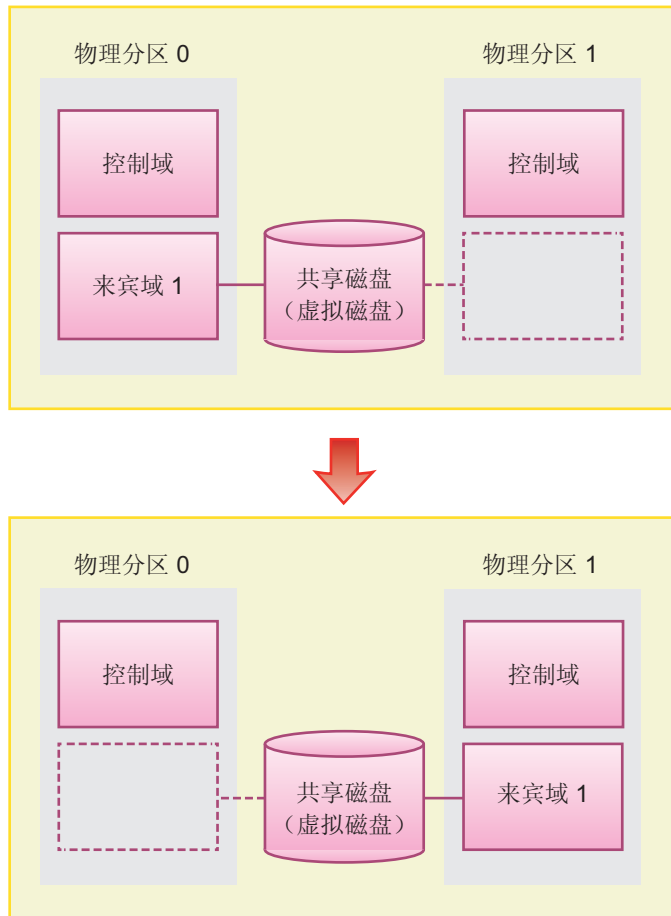
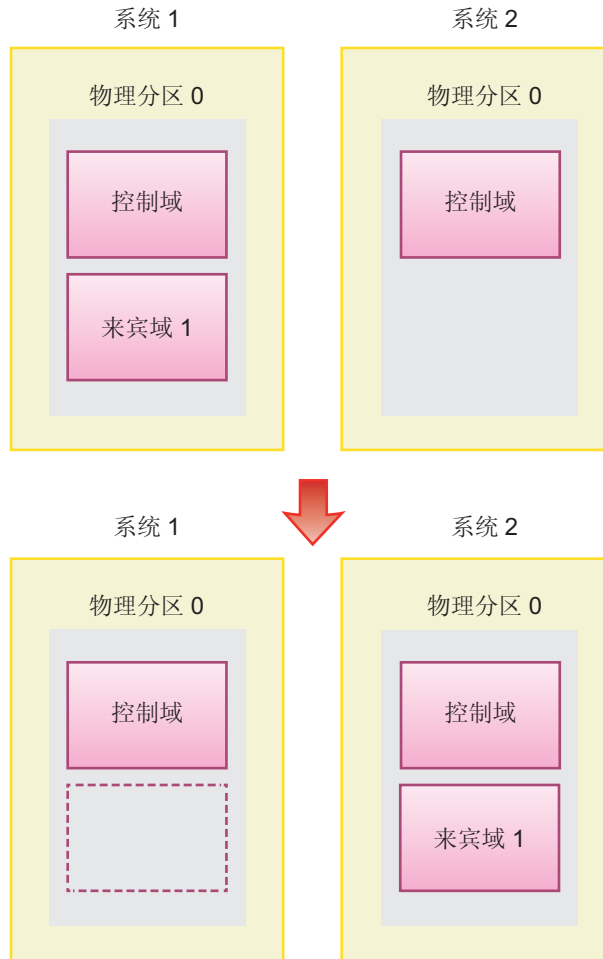


图 7-2 不同系统间的迁移



7.1.1 实时迁移的要求

需要符合以下要求才能使用实时迁移功能。

- Logical Domains Manager（逻辑域管理器）在源和目标中运行
在源和目标中必须安装了 Oracle VM Server for SPARC 2.1 或更高版本，并且 Logical Domains Manager（逻辑域管理器）必须正在运行。
- 目标具有足够多的可用资源
目标物理分区必须有足够可用的适用于源来宾域的硬件资源。

实时迁移除了上述要求之外，还有许多其它要求。有关这些要求的详情，请参见所使用版本的《Oracle VM Server for SPARC Administration Guide》中的“Migrating Domains”，所使用版本的《产品说明》中的“Oracle VM Server for SPARC 的实时迁移注意事项”。您可以通过执行指定 `-n` 选项的 `ldm migrate-domain` 命令，检查您的环境是否符合这些要求。此时，不符合要求的项目将报告为错误。

7.2 迁移来宾域

若需使用实时迁移功能迁移来宾域，请登录到控制域，然后执行 `ldm migrate-domain` 命令。

`ldm migrate-domain` 命令

注 - 此处只显示主要选项。有关详情，请参见您所使用版本的《Oracle VM Server for SPARC Reference Manual》。

```
# ldm migrate-domain [-n] source-ldom target-host[:target-ldom]
```

- `-n` 选项
此选项只对是否可以迁移来宾域执行测试。不执行实际迁移。
- `source-ldom`
指定源来宾域的名称。
- `target-host`
指定属于目标物理分区控制域的主机名。此主机名应是网络中有效的主机名。
- `target-ldom`
指定目标来宾域的名称。如果未指定，将与源来宾域名称相同。

7.3 来宾域迁移示例

本节显示实时迁移的具体示例。在此示例中，名为 `ldom1` 的来宾域迁移到其主机名是 `target1` 的控制域的物理分区。

1. 以 **root**（根）用户身份登录控制域。
2. 执行 **`ldm migrate-domain`** 命令将来宾域 **`ldom1`** 迁移到其主机名是 **`target1`** 的控制域的物理分区。

```
# ldm migrate-domain ldom1 target1
```

要求您输入目标物理分区控制域根用户的密码。

3. 输入密码。

```
Target Password:*****
```

4. 执行迁移来宾域。

名为 `ldom1` 的来宾域在目标物理分区中创建，`ldom1` 被从源中删除。

使用物理分区动态重新配置的环境配置示例和步骤

本附录介绍使用物理分区动态重新配置的环境配置示例和步骤。

使用模块化架构 (BB) 配置与 Oracle VM Server for SPARC 3.2 或更高版本的系统的示例可以在 Technical Park 中找到。有关详细信息，请参见 Technical Park 中的 “Low Cost, High Availability Systems That Can be Implemented Using Building Blocks”。
<http://www.fujitsu.com/jp/products/computing/servers/unix/sparc/technical/document/index.html>

配置以下环境作为示例。

- 将不支持物理分区动态重新配置的系统更新到支持物理分区动态重新配置的系统
- 对于从 XCP 2220 或更高版本的新安装，逻辑域配置无可用硬件资源
- 对于从 XCP 2220 或更高版本的新安装，逻辑域配置有可用硬件资源
- 采用 XCP 2220 或更高版本新安装的系统从 1BB 至 2BB 配置的扩展
- 向仅由 SPARC64 X 处理器配置的物理分区安装由 SPARC64X+ 处理器配置的系统板
- 仅配置控制域的系统板的活动状态更换（适用于 Oracle VM Server for SPARC 3.2 或更高版本）

在上述情况下，使用物理分区动态重新配置的活动状态更换步骤说明如下。

- 对于从 XCP 2220 或更高版本的新安装，逻辑域配置无可用硬件资源
- 对于从 XCP 2220 或更高版本的新安装，逻辑域配置有可用硬件资源

使用物理分区动态重新配置，活动状态添加的实际步骤也说明如下。

- 采用 XCP 2220 或更高版本新安装的系统从 1BB 至 2BB 配置的扩展
- 向仅由 SPARC64 X 处理器配置的物理分区安装由 SPARC64X+ 处理器配置的系统板
- 仅配置控制域的系统板的活动状态更换（适用于 Oracle VM Server for SPARC 3.2 或更高版本）

A.1 将不支持物理分区动态重新配置的系统更新到支持物理分区动态重新配置的系统

本节介绍在 SPARC M10 中将系统更新到可使用物理分区动态重新配置的环境配置示例和步骤。需要更新的系统将是其版本早于不支持物理分区动态重新配置的 XCP 2210 的一个版本。有关物理系统板 (PSB) 和逻辑系统板 (LSB) 的详情，请参见 ["1.3.1 了解物理分区组件"](#)。

A.1.1 配置示例

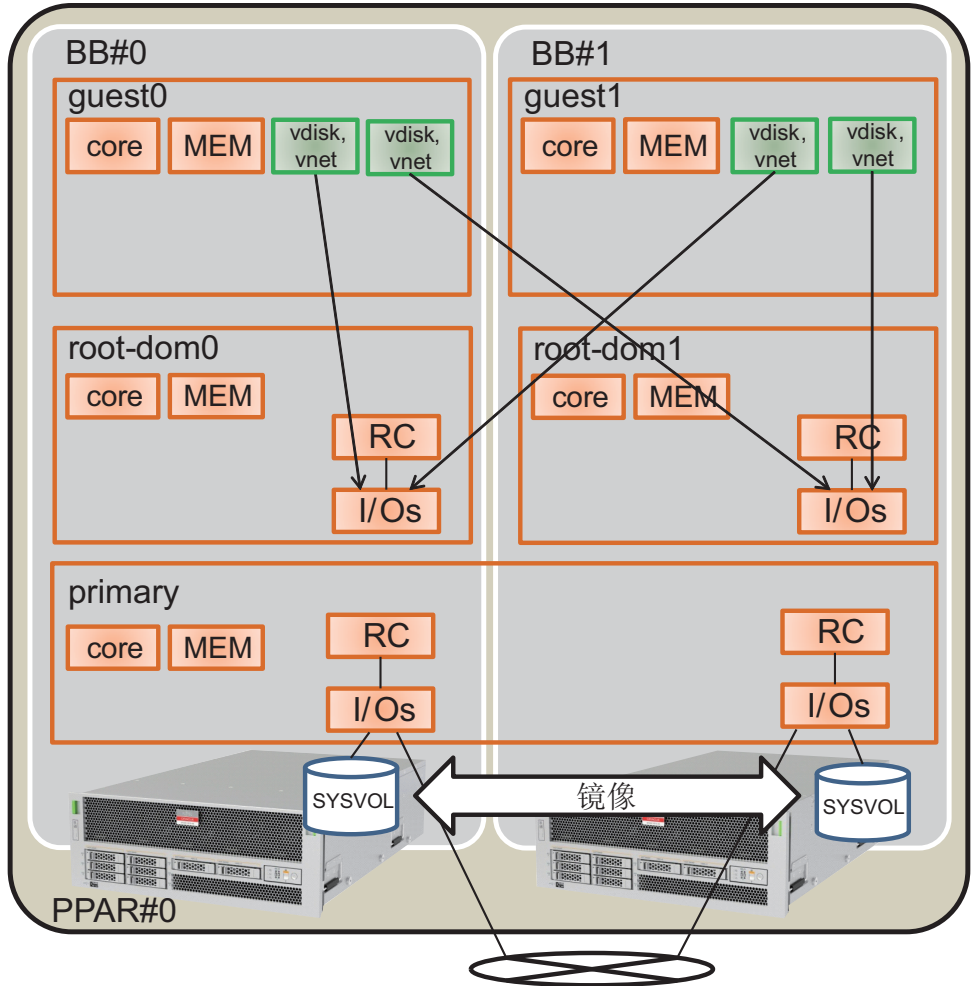
本节介绍由两个 SPARC M10-4S 即两个系统板 (PSB<BB>) (2BB 配置) 组成逻辑域的配置示例。

若需使用物理分区动态重新配置启用 SPARC M10-4S 的活动状态更换，必须满足以下条件。

- 通过将在各 SPARC M10-4S 的根合成体下的 I/O 设备连接到系统卷的 I/O 设备和控制域的网络上，创建一个冗余配置。
这是由于以下原因：即使控制域重新启动时，I/O 设备的配置发生了变更，当移除一个 SPARC M10-4S 时，剩余 SPARC M10-4S 的 I/O 设备可被用于继续该操作。
- 分割各 SPARC M10-4S 的根域。
这是由于以下原因：当移除 SPARC M10-4S 时，必须避免发生无法使用其他 SPARC M10-4S 的物理 I/O 设备的服务的这种状态。
- 使用由虚拟 I/O 设备 (vdisk, vnet) 组成的来宾域来操作任务。
因此，在 SPARC M10-4S 移除时，根域被停止，物理 I/O 设备可以被断开。此外，这些操作将不会影响正在执行任务的来宾域。

下图显示了一个可执行物理分区动态重新配置的配置示例。

图 A-1 可执行物理分区动态重新配置的配置示例



A.1.2 配置步骤

下面的示例将不支持物理分区动态重新配置的系统更新到支持物理分区动态重新配置的系统：

1. 将逻辑域配置信息保存到 XML 文件。

如果 XCP 固件更新后启用 PPAR DR 功能，逻辑域配置将恢复到出厂设置状态 (factory-default)。因此，无法使用保存到 XSCF 的逻辑域配置信息。出于上述原因，将当前逻辑域配置信息保存到 XML 文件，然后更新 XCP 固件。更新后，将逻辑域配置信息从 XML 文件恢复，以便启用简易逻辑域重新配置。

下面介绍将保存在 XSCF 中的逻辑域配置信息保存到 XML 文件的步骤。

但是，有关保存到 XML 文件中的配置信息的详情，请参见《Oracle VM Server for SPARC Administration Guide》。

- a. 登录控制域。
- b. 切换到需要保存的配置信息。

在控制域上执行 `ldm list-spconfig` 命令来显示逻辑域配置信息。[current] 行显示当前应用的配置信息。下面的示例中被应用的是 `ldm-set3`。

```
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3 [current]
```

如果仅显示 [next poweron]，则存储在 XSCF 中的逻辑域配置信息和存储在控制域中的逻辑域配置信息之间存在差异。因此，请执行 `ldm add-spconfig` 命令，以便使用不同的名称保存当前逻辑域配置信息。

```
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3 [next poweron]
# ldm add-spconfig ldm-set4
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3
ldm-set4 [current]
```

当显示在 [current] 的逻辑域配置信息与您需要保存的配置信息相匹配，请执行步骤 1-c。

如果不同，请执行 `ldm set-spconfig` 命令切换到您需要保存的配置信息。在下面的示例中，使用 `ldm set-spconfig` 命令切换配置信息，然后保存 `ldm-set1`。

```
# ldm set-spconfig ldm-set1
# ldm list-spconfig
factory-default
ldm-set1 [next poweron]
ldm-set2
ldm-set3 [current]
```

在 XSCF 上执行 `poweroff` 和 `poweron` 命令，以便关闭物理分区 (PPAR) 的电源，然后再将其开启电源。在这种情况下，将逻辑域状态变更到 Oracle Solaris 正在运行的状态或 inactive 状态后，执行 `poweroff` 命令。

下面的示例关闭 PPAR-ID 0 的电源，然后再将其开启电源。

```
XSCF> poweroff -p 0
XSCF> poweron -p 0
```


执行 `ldm list-sconfig` 命令检查指定的逻辑域配置信息是否已设置。下面的示例显示将 `ldm-set1` 设置为当前配置信息。

```
# ldm list-sconfig
factory-default
ldm-set1 [current]
ldm-set2
ldm-set3
```

- c. 执行 **ldm list-constraints** 命令将当前逻辑域配置信息保存到 **XML** 文件。

执行 `ldm list-constraints` 命令将当前逻辑域配置信息保存到 XML 文件。

下面的示例将当前逻辑域配置信息保存到 `ldm-set1.xml`。如需防止已保存的 XML 文件丢失，请将其备份到其他介质或类似物。

```
# ldm list-constraints -x > /ldm-set1.xml
```

- d. 检查是否将配置信息保存到 **XML** 文件。

执行 Oracle Solaris 的 `more` 命令或相似命令，检查信息是否已经保存到 XML 文件。

```
# more /ldm-set1.xml
<?xml version="1.0"?>
<LDM_interfaceversion="1.3" xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
```

- e. 如需保存其他逻辑域配置信息，请重复步骤 **1.b** 至 **1.e**。

如果存在其他您需要保存的配置信息，请重复步骤 **1.b** 至 **1.e**。

2. 保存 **XSCF** 设置信息。

可将 XSCF 设置信息保存到 USB 设备或通过网络保存到外部服务器。下面说明各种方法。

- 将 **XSCF** 设置信息保存到 **USB** 设备

- a. 登录主 **XSCF**。

执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。如果正在使用备用 XSCF，请再次登录主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

- b. 将设置信息保存到主 **XSCF** 的 **USB** 设备。

i. 将 **USB** 设备连接到主 **XSCF** 的 **XSCF** 单元面板（后面板）上的 **USB** 端口。

ii. 指定 **XSCF** 上本地 **USB** 设备的输出文件名，然后执行 `dumpconfig` 命令。

设置信息以 base64 编码文本格式采用指定的文件名被保存。

```
XSCF> dumpconfig file:///media/usb_msd/backup-file.txt
operation completed
```

当 "operation completed" 信息出现时，数据传输已经正常结束。

iii. 数据传输完成后，从 **USB** 端口拔下 **USB** 设备。

c. 使用编辑器或个人计算机上的类似软件，查看保存在设置文件中的以下信息。

- User-Comments: 在 dumpconfig 命令中指定 -c 选项时所编制的注释
- Created: 保存信息的日期和时间
- Platform: 型号名
- Serial-No: 系统序列号

```
XSCF Configuration File
User-Comments:
Encrypted: No
Created: Mon Jan 27 13:47:38 2014
Platform: M10-4S
Serial-No: 2111234001
Chassis-serial80:
Chassis-serial81:
Chassis-serial82:
Chassis-serial83:
Chassis-serial00:2111234001
Chassis-serial01:2111234003
Chassis-serial02:
Chassis-serial03:
Chassis-serial04:
Chassis-serial05:
Chassis-serial06:
Chassis-serial07:
Chassis-serial08:
Chassis-serial09:
Chassis-serial10:
Chassis-serial11:
Chassis-serial12:
Chassis-serial13:
Chassis-serial14:
Chassis-serial15:
Version: 0001
begin-base64_common
U1VOVyxTUEFSQy1FbnRlcnByaXNlAAAAAAAFmZ6gAAPrfADhbdAAAAIyMTExMjM0MDAzAAAA
...
```

- 通过网络将设置信息保存到外部服务器

a. 登录主 **XSCF**。

执行 showbbstatus 命令来检查您已经登录的 XSCF 是否是主 XSCF。如果正在使用备用 XSCF，请再次登录主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

b. 通过网络指定目标目录，以便保存 **XSCF** 设置信息。

指定目标目录和输出文件名，然后执行 **dumpconfig** 命令。设置信息以 base64 编码文本格式采用指定的文件名被保存。

```
XSCF> dumpconfig -u "user-name" "ftp://server/backup/backup-sca-
ff2-16.txt"
operation completed
```

当 "operation completed" 信息出现时，数据传输已经正常结束。

c. 使用编辑器或个人计算机上的类似软件，检查所保存的设置文件信息。

检查以下信息。

- User-Comments: 在 **dumpconfig** 命令中指定 -c 选项时所编制的注释
- Created: 保存信息的日期和时间
- Platform: 型号名
- Serial-No: 系统序列号

```
XSCF Configuration File
User-Comments:
Encrypted: No
Created: Mon Jan 27 13:47:38 2014
Platform: M10-4S
Serial-No: 2111234001
Chassis-serial80:
Chassis-serial81:
Chassis-serial82:
Chassis-serial83:
Chassis-serial100:2111234001
Chassis-serial101:2111234003
Chassis-serial102:
Chassis-serial103:
Chassis-serial104:
Chassis-serial105:
Chassis-serial106:
Chassis-serial107:
Chassis-serial108:
Chassis-serial109:
Chassis-serial110:
Chassis-serial111:
Chassis-serial112:
Chassis-serial113:
Chassis-serial114:
Chassis-serial115:
Version: 0001
begin-base64_common
U1VOVyxTUEFSQy1FbnRlcnByaXNlAAAAAAAAAFmZ6gAAPrfADhbdAAAAAIyMTExMjM0MDAzAAAA
...
```

3. 更新 **Oracle Solaris** 和 **Oracle VM Server for SPARC**。
- 获取支持物理分区动态重新配置的 Oracle Solaris 和 Oracle VM Server for SPARC，并将其应用到系统。
- a. 更新控制域的 **Oracle VM Server for SPARC** 。
- 请参见您的服务器最新的《Product Notes》，检查最新版的 Oracle Solaris 和 Oracle VM Server for SPARC。然后使用以下步骤更新 Oracle VM Server for SPARC。
- i. 获取用于控制域的 **Oracle VM Server for SPARC** 。
- 如果控制域是 **Oracle Solaris 10**
- 获取支持物理分区动态重新配置的 Oracle VM Server for SPARC。有关如何获取的详细信息，请参见您的服务器的最新《Product Notes》。
- 如果控制域是 **Oracle Solaris 11**
- 获取 SRU11.1.14.0 或更高版本。
- ii. 无论何时完成更新 **Oracle VM Server for SPARC** 或控制域的 **Oracle Solaris**，都请重新启动控制域。
- 重新启动控制域可能导致 I/O 域崩溃或来宾域 I/O 停止。如果存在这样的逻辑域，请提前使用 shutdown 命令停止，或者从控制域执行 ldm stop-domain 命令将其停止。
- 使用下面的方法，可以检查逻辑域是否将要被停止。
- 从控制域分配 **PCIe** 端点的 **I/O** 域
- 执行 ldm list-io 命令确定分配给逻辑域的 PCIe 端点的根合成体 (BUS) 是否被分配给 primary。下面的示例表明分配给 iodom0 的 PCIe 端点 "/BB0/PCI3" 和 "/BB0/PCI4" 的总线 "PCIE2" 被分配给 primary (控制域)。请注意，由于这里阐述的配置与 "[A.1.1 配置示例](#)" 中的配置不同，所以对逻辑域和一些其他信息进行了变更。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
....				
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/PCI0	PCIE	PCIE1	primary	OCC
/BB0/PCI3	PCIE	PCIE2	iodom0	OCC
/BB0/PCI4	PCIE	PCIE2	iodom0	OCC
/BB0/PCI7	PCIE	PCIE3	primary	OCC
/BB0/PCI8	PCIE	PCIE3	primary	EMP
....				

- 分配控制域虚拟服务的来宾域
- 执行 ldm list-bindings primary 检查虚拟网络交换机 (VSW) 和连接目标 (PEER) 之间的对应关系，以及虚拟磁盘服务 (VDS) 和使用虚拟磁盘的逻辑域 (CLIENT) 之间的对应关系。在下面的示例中，可以确认 VSW

PEER 和 VDS CLIENT 设置为 "guestdom0" 。

```
# ldm list-bindings primary
....
VSW
  NAME          MAC          NET-DEV  ID  DEVICE      LINKPROP
  DEFAULT-VLAN-ID PVID
VID
  vsw0          MTU    MODE  INTER-VNET-LINK
  1            1500    on
  PEER          MAC          PVID VID
  MTU    MAXBW    LINKPROP
INTERVNETLINK
  vnet0@guestdom0          00:14:4f:fa:64:dd 1
  1500
VDS
  NAME          VOLUME          OPTIONS          MPGROUP          DEVICE
....
  CLIENT          VOLUME
  vdisk0@guestdom0  vol0
```

执行 `ldm stop-domain` 命令停止上述已检查的逻辑域。在下面的示例中，执行 `ldm stop-domain` 命令来停止 `iodom0` 和 `guestdom0`，然后执行 `ldm list-domain` 命令来检查是否设置为 `bound` 状态。

```
# ldm stop-domain guestdom0
LDom guestdom0 stopped
# ldm stop-domain iodom0
LDom iodom0 stopped
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    64      58G       0.0%    6h 3m
guestdom0     bound     -----  5100    64      64G
iodom0        bound     ----v-   5000    32      32G
```

iii.更新控制域的 Oracle VM Server for SPARC 。

- Oracle Solaris 10 的情形

卸载旧版 Oracle VM Server for SPARC，然后安装新版 Oracle VM Server for SPARC。有关详情，请参见所获得的 Oracle VM Server for SPARC 随附的 README 文件。

- Oracle Solaris 11 的情形

应用 SRU。有关详情，请参见所获得的 SRU 的安装手册。

iv.执行 `shutdown` 命令重启控制域。

```
# shutdown -i6 -g0 -y
....
```

v. 执行 `ldm start-domain` 命令启动在上述 ii 中停止的逻辑域。

在下面的示例中, 执行 `ldm start-domain` 命令启动 "iodom0" 和 "guestdom0", 然后执行 `ldm list-domain` 命令检查是否设置为 active 状态。

```
# ldm start-domain guestdom0
# ldm start-domain iodom0
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	58G	0.0%	6h 3m
guestdom0	active	-n----	5100	64	64G		
iodom0	active	-n--v-	5000	32	32G		

b. 更新除控制域之外的逻辑域的 **Oracle Solaris**。

请参见您的服务器的最新《Product Notes》，并为除控制域之外的逻辑域更新 Oracle Solaris。有关更新步骤的详情，请参见与各更新有关的信息。

4. 更新 **XCP** 固件。

将搭载 SPARC64 X+ 处理器的 PSB (BB) 安装在配置有 SPARC64 X 处理器的物理分区。若需进行此操作，请将配置有 SPARC64 X 处理器的物理分区更新为 XCP 2220 或更高版本的 XCP 固件。

a. 获取最新版 **XCP** 固件。

i. 下载 **XCP** 固件的程序文件。

从网站上将 XCP 固件的程序文件（XCPxxxx.tar.gz 或 XCPxxxx.exe）下载到与本系统相连接的 PC 上的任意一个文件夹。

使用以下其中一种方法获取您正在使用的服务器的固件。

- 日文网站

曾经登录过 SupportDesk 的用户可从 SupportDesk-Web 获取固件。

- 全球网站

有关获取固件最新文件方法的详情，请联系本公司销售人员。

提供以下文件。

- 固件程序文件

（XSCF Control Package (XCP) 文件）

- XSCF 扩展 MIB (XSCF-SP-MIB) 定义文件

ii. 检查所下载程序文件的 **XCP** 版本。

检查所下载 XCP 固件程序文件的版本。对于 XCP 版本，请参考固件程序（tar.gz 格式）的文件名中的 4 位数字，以便检查是否需要应用的 XCP 固件版本更新。例如，如果程序文件的名称是 "XCP2220.tar.gz"，则 XCP 版本是 2220。

iii. 解压缩下载的程序文件。

解压缩所下载 XCP 固件的程序文件。导入到系统中的 XCP 映像文件被展开。例如，如果解压缩 "XCP2220.tar.gz"，则 "BBXCP2220.tar.gz" 被展开。

b. 检查当前 **XCP** 固件版本。

i. 登录主 **XSCF**。

执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。如果正在使用备用 XSCF，请再次登录主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

ii. 执行 **version** 命令检查当前系统的 **XCP** 版本。

更新固件前，请检查当前系统的 XCP 版本。在下面的示例中，执行带有 -c xcp 选项的 version 命令，以便检查 XCP 版本是否是不支持物理分区动态重新配置的 XCP 2041。

```
XSCF> version -c xcp
BB#00-XSCF#0 (Master)
XCP0 (Current): 2041
XCP1 (Reserve): 2041
BB#01-XSCF#0 (Standby)
XCP0 (Current): 2041
XCP1 (Reserve): 2041
```

c. 更新 **XCP** 固件。

i. 执行 **poweroff** 命令关闭所有物理分区的电源。

```
XSCF> poweroff -a
```

执行 showpparstatus 命令检查是否所有物理分区电源已关闭。

```
XSCF> showpparstatus -a
PPAR-ID          PPAR Status
0                 Powered Off
```

ii. 执行 **showhardconf** 命令检查主、备 XSCF 的 [Status] 是否是 "Normal"。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Service;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB;
:
```

iii. 执行 **getflashimage** 命令导入 **XCP** 镜像文件。

在下面的示例中，将 USB 设备连接到主 XSCF 的 XSCF 单元面板（后面板）上的 USB 端口（印刷 MAINTENANCE ONLY），并导入 XCP 映像文件。

```
XSCF> getflashimage file:///mnt/share/scf-firm/xscf/user/scfadmin/BBXCP2220.tar.gz
  0MB received
  1MB received
  2MB received
...
 86MB received
 87MB received
 88MB received
Download successful: 90539 Kbytes in 58 secs (1562.668 Kbytes/sec) Checking
file...
MD5: 2b89c06548205ce35a8ecb6c2321d999
```

当正常结束消息"Download successful: ..." 和 "MD5: ..." 出现时，表示 XCP 镜像文件的导入已经结束。

注 - 当 XCP 镜像文件导入时，可能出现消息 "Warning: About to delete existing old versions"。此信息要求您确认是否要删除已经导入的旧版 XCP 映像文件。如果出现 "Continue?"，请输入 "y" 继续导入处理。

注 - 如果导入 XCP 镜像文件后出现 "Error:File is invalid or corrupt" 消息，这表明导入的 XCP 镜像文件不合适。此 XCP 镜像文件可能已损坏。因此，请获取正确的 XCP 镜像文件，然后将其导入。

iv. 执行 **getflashimage -l** 命令检查导入的 XCP 镜像文件的版本。

```
XSCF> getflashimage -l
Existing versions:
      Version              Size   Date
BBXCP2220.tar.gz      92712351  Thu May 23 15:01:42 JST 2014
```

v. 执行 **flashupdate -c check** 命令检查导入的 XCP 镜像文件是否可用于更新。

执行 flashupdate 命令后，立即执行 showresult 命令。如果结束值是 0，可进行更新。

```
XSCF> flashupdate -c check -m xcp -s 2220
XCP update is started. [3600sec]
0XSCF>
XSCF> showresult
0
XSCF>
```

vi. 执行 **flashupdate** 命令更新固件。


```

XSCF> flashupdate -c update -m xcp -s 2220
The XSCF will be reset. Continue? [y|n]: y
XCP update is started. [3600sec]
0.....30.....60.....90.....120.....150.....180.....210.....240.....-
270.....300.....330.....360.....390.....420.....450.....480.....510.....|
540.....570.....600.....630.....660.....690.....720.....750.....780.....-
810.....840.....870.....900.....930
:

```

此时，XSCF 被重新启动，XSCF 会话被断开。此时，XCP 固件更新尚未完成。

注 - 关于固件的操作时间，更新时间约 60 分钟，完成更新后 XSCF 的自动切换大约需要 10 分钟。

注 - 如需安全地进行更新，请勿操作物理分区的电源，直到出现 "XCP update has been completed" 信息，表明 XCP 固件已更新。

vii.再次连接到主 XSCF。

XSCF 重新启动之后，主 XSCF 和备用 XSCF 将与原来的状态恰好相反。例如，如果 BB-ID 0 的主 XSCF 执行固件更新，然后再次进行 XSCF 连接，则 BB-ID 1 进入主状态，BB-ID 0 进入备用状态。

注 - 如果设置了继承的 IP 地址，然后用于连接，将自动连接到主 XSCF 。

viii.执行 showbbstatus 命令来检查是否已经登录主 XSCF 。如果正在使用备用 XSCF，请再次连接到主 XSCF 。

```

XSCF> showbbstatus
BB#01 (Master)

```

ix.执行 showlogs monitor 命令来检查 XCP 固件更新是否已完成。

如果出现 "XCP update has been completed" 信息，则 XCP 固件更新已经完成。

```

XSCF> showlogs monitor
May 29 09:38:05 M10-4S-0 Event: SCF:XCP update is started (XCP version=2220:
last version=2041)
May 29 09:40:31 M10-4S-0 Event: SCF:XSCF update is started (BBID=0, bank=0)
May 29 09:40:46 M10-4S-0 Event: SCF:XSCF update is started (BBID=1, bank=0)
May 29 09:41:03 M10-4S-0 Event: SCF:XSCF writing is performed (BBID=0, XSCF
version=02220000)
:
May 29 09:50:41 M10-4S-0 Event: SCF:XSCF bank apply has been completed
(BBID=1, bank=0, XCP version=2220: last version=2041)
May 29 10:01:50 M10-4S-0 Event: SCF:XSCF bank apply has been completed
(BBID=0, bank=0, XCP version=2220: last version=2041)
May 29 10:01:51 M10-4S-0 Event: SCF:Change Master Start (BB#01)May 29 10:03:26

```

```
M10-4S-1 Event: SCF:Change Master Complete (BB#01)
May 29 10:05:00 M10-4S-1 Event: SCF:Standby XSCF Ready (BBID#00)
:
May 29 10:32:38 M10-4S-1 Event: SCF:XCP update has been completed (XCP
version=2220: last version=2041)
May 29 10:32:39 M10-4S-1 Event: SCF:This XSCF will be switched back to standby
mode after completion of firmware update
May 29 10:32:39 M10-4S-1 Event: SCF:Change Master Start (BB#00)
May 29 10:33:29 M10-4S-1 Event: SCF:Change Master Complete (BB#00)
May 29 10:42:29 M10-4S-1 Event: SCF:Standby XSCF Ready (BBID#01)
```

随后，XSCF 的主、备状态将自动切换。

当出现 "This XSCF will be switched back to standby mode after completion of firmware update"、"Change Master Complete" 和 "Stand by XSCF Ready" 信息时，已经完成切换。

注 - 如果没有出现 "XCP update has been completed" 信息，则更新尚未完成。再次执行 `showlogs monitor` 命令来检查更新是否已完成。

注 - 如果执行固件更新，主、备状态将切换两次。其结果是，主 XSCF 返回到 `flashupdate` 命令被执行的状态。当主、备状态切换时，XSCF 会话可能断开。如果 XSCF 会话已经断开，请再次进行连接。

如需检查切换是否已经完成，请执行 `showhardconf` 命令检查主、备 XSCF 的 [Status] 是否是 "Normal"。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Service;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB;
:
```

注 - 假设执行 `showhardconf` 命令，出现 "Cannot communicate with the other XSCF.Check the other XSCF's state." 信息。此信息表明 XSCF 的主、备状态之间的切换尚未完成。再次执行 `showhardconf` 命令来检查切换是否已完成。

x.若需将主、备 XSCF 的状态返回更新前的状态，请执行 `switchscf` 命令。

```
XSCF> switchscf -t Standby
The XSCF unit switch between the Active and Standby states.
Continue? [y|n] :y
```

由于 XSCF 会话由 XSCF 重新启动被断开，请再次连接到主 XSCF。

如需检查切换是否已经完成，请执行 **showhardconf** 命令检查主、备 XSCF 的 [Status] 是否是 "Normal"。

注 - 假设执行 **showhardconf** 命令，出现 "Cannot communicate with the other XSCF.Check the other XSCF's state." 信息。此信息表明 XSCF 的主、备状态之间的切换尚未完成。再次执行 **showhardconf** 命令来检查切换是否已完成。

xi. 执行 **version** 命令检查固件是否是新版本的固件。

```
XSCF> version -c xcp
BB#00-XSCF#0 (Master)
XCP0 (Current): 2220
XCP1 (Reserve): 2220
BB#01-XSCF#0 (Standby)
XCP0 (Current): 2220
XCP1 (Reserve): 2220
```

注 - 如果开启物理分区的电源并更新固件，则 CMU 固件的当前排列将是新排列。如果关闭物理分区的电源并更新固件，则 CMU 固件的预约排列和当前排列将是新排列。有关 XCP 版本相关的 CMU 固件的版本详情，请参见最新产品说明中的 "现有 XCP 固件版本和支持信息"。

5. 指定 **XSCF PPAR DR** 模式和将要启动的逻辑域，然后启动物理分区。

注 - 如需使用物理分区动态重新配置，有必要使用 **setpparmode** 命令来启用 PPAR DR 模式。但是，如果启用了 PPAR DR 模式，逻辑域配置信息将变更为出厂默认值。因此，无法使用现有的逻辑域配置信息。在这种情况下，您必须新建逻辑域配置信息。

若需使用现有的逻辑域配置信息，请禁用 PPAR DR 模式。在使用现有的逻辑域配置信息时，假设启用了 PPAR DR 模式，并且开启了物理分区的电源。将发生虚拟机监控程序中止和未能正常启动 Oracle VM Server for SPARC 等问题。

a. 启用物理分区动态重新配置功能（PPAR DR 功能）。

i. 执行 **showpparmode** 命令查看 PPAR DR 模式设置。

下面的示例指定物理分区编号 0 (PPAR ID 0)。

```
XSCF> showpparmode -p 0
Host-ID                :9007002b
Diagnostic Level       :min
Message Level          :normal
Alive Check            :on
Watchdog Reaction      :reset
Break Signal           :on
Autoboot (Guest Domain) :on
```

Elastic Mode	:off
IOreconfigure	:false
PPAR DR(Current)	:-
PPAR DR(Next)	:off

ii.如果 **PPAR DR** 模式被禁用, 请执行 **setpparmode** 命令来启用该模式。

```
XSCF> setpparmode -p 0 -m ppar_dr=on
```

iii.执行 **showpparmode** 命令来检查 **PPAR DR** 模式是否已启用。

```
XSCF> showpparmode -p 0
Host-ID                :9007002b
Diagnostic Level        :min
Message Level           :normal
Alive Check             :on
Watchdog Reaction       :reset
Break Signal            :on
Autoboot(Guest Domain) :on
Elastic Mode            :off
IOreconfigure           :false
PPAR DR(Current)       :-
PPAR DR(Next)          :on
```

b. 开启物理分区的电源。

i. 执行 **showdomainconfig** 命令检查将要启动的逻辑域的配置信息是否设置为出厂默认值。

下面的示例指定物理分区编号 0 (PPAR ID 0)。

```
XSCF> showdomainconfig -p 0
PPAR-ID :0
Booting config
(Current) : factory-default
(Next) : factory-default
-----
Index :1
config_name :factory-default
domains :1
date_created:-
-----
Index :2
config_name :ldm-set1
domains :8
date_created:'2012-08-0811:34:56'
-----
Index :3
config_name :ldm-set2
domains :20
date_created:'2012-08-0912:43:56':
```

ii. 执行 **poweron** 命令来开启物理分区的电源。

下面的示例指定物理分区编号 0 (PPAR ID 0)。

```
XSCF> poweron -p 0
```

6. 使用 **XML** 文件，重新配置逻辑域。

完成物理分区的启动后，从步骤 1 中保存的 XML 文件重新配置逻辑域。

a. 检查系统是否已经采用控制域的 **factory-default** 设置启动。

执行 **ldm list-spconfig** 命令检查 [current] 是否在 factory-default 行显示。

```
# ldm list-spconfig
factory-default [current]
ldm-set1
ldm-set2
ldm-set3
```

b. 使用 **ldm init-system** 命令，从 XML 文件应用逻辑域配置信息。

执行 **ldm init-system** 命令应用所保存的 XML 文件设置。然后，执行 **shutdown** 命令重新启动控制域。

```
# ldm init-system -i /ldm-set1.xml
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
# shutdown -y -g0 -i6
```

c. 若需启用物理分区动态重新配置，请调整各逻辑域的内存容量。

执行 **ldm list-domain** 命令或相似命令，检查是否已经从 XML 文件配置逻辑域。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	58G	0.0%	6m
guest0	inactive	-----		64	64G		
guest1	inactive	-----		64	64G		
root-dom0	inactive	-----		32	32G		
root-dom1	inactive	-----		32	32G		

当 Oracle VM Server for SPARC 的版本早于 3.2 时，若需启用物理分区动态重新配置，请将需要分配给逻辑域的内存容量设置为“分配给逻辑域的内核数量 × 256 MB”的倍数。如需重新设置除控制域之外的逻辑域的内存容量，请使用 **ldm set-memory** 命令。

如需重置控制域的内存容量，如下进行操作：使用 **ldm start-reconf** 命令进入延迟重新配置模式，使用 **ldm set-core** 命令设置相同的内核数量，然后使用 **ldm set-memory** 命令重新设置内存容量。然后，重新启动 Oracle Solaris。

当 Oracle VM Server for SPARC 的版本早于 3.2 时，首先获取 8192 MB 的倍数，使该值更接近初始设置值 (58 GB = 59392 MB)。其结果是 $59392/8192 = 7.25$ 。舍位至 7。因此，需要重新分配给控制域的内存容量为 $8192 \text{ MB} \times 7 = 56 \text{ GB}$ 。

接下来, 下面的示例显示重新设置控制域内存为 56 GB 的命令的执行。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
# ldm set-core 32 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm set-memory 56G primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# shutdown -i6 -g0 -y
....
```

执行 `ldm list-domain` 命令检查内存容量 (MEMORY) 是否已经正确设置。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active     -n-cv-   UART    64     56G     0.0%   6m
guest0        inactive  -----           64     64G
guest1        inactive  -----           64     64G
root-dom0     inactive  -----           32     32G
root-dom1     inactive  -----           32     32G
```

执行 `ldm bind-domain` 命令绑定各逻辑域。

```
# ldm bind-domain root-dom0
# ldm bind-domain root-dom1
# ldm bind-domain guest0
# ldm bind-domain guest1
```

然而，假设当禁用 PPAR DR 功能时，逻辑域配置信息从其被保存的 XML 文件恢复。在这种情况下，当启用 PPAR DR 功能时，虚拟机监控程序区的内存容量增大，使得分配的内存量将不足。如果在没有足够的内存被分配时，执行了 `ldm bind-domain` 命令，则处理将终止并出现以下错误信息，而且 inactive 状态仍保持原状态。

```
# ldm bind-domain guest1
Not enough free memory present to meet this request.
Could not bind requested memory for LDom guest1.
```

在此种情况下，请参见步骤 6.c 重新设置分配给逻辑域的内存容量。要调整已经处于绑定状态的逻辑域的内存分配，请执行 `ldm unbind-domain` 命令来释放 `bound` 状态，然后进行设置。当重新设置已经完成时，重新执行 `ldm bind-domain` 命令进行重新绑定。

d. 检查内存资源分配状态。

执行 `ldm list-devices -a` 命令检查分配给各逻辑域的内存的连续区域（内存块）的状态。然后，检查是否存在大量的小内存块。可将检查是否存在大量的 `SIZE` 约 256 MB 到 512 GB 的内存块作为参考。

```
# ldm list-devices -a memory
MEMORY
PA                               SIZE                               BOUND
0x7000000000000000              32G                           root-dom0
0x7200000000000000              32G                           root-dom1
0x7400000000000000              32G                           guest0
0x7600008000000000              1272M                         _sys_
0x7600500000000000              2G                             primary
0x7600d00000000000              29440M                       guest1
0x7800000000000000              32G                           guest0
0x7a00000000000000              32G                           guest1
0x7c00000000000000              32G                           primary
0x7e00008000000000              1272M                         _sys_
0x7e00500000000000              512M                         _sys_
0x7e00700000000000              256M                         _sys_
0x7e00800000000000              3328M                       guest1
0x7e01500000000000              4864M
0x7e02800000000000              22G                           primary
```

如果内存块被分解，执行 `ldm unbind-domain` 命令将各逻辑域恢复到 `unbind` 状态。然后，按照步骤 6.c 所述执行重新设置。

e. 启动所有逻辑域。

执行 `ldm start-domain` 命令启动除控制域之外的逻辑域。假设存在物理 I/O 分配关系（根域，I/O 域）和虚拟 I/O 服务和客户端之间关系等从属关系。在这种情况下，首先启动提供物理和虚拟 I/O 服务的域。

下面的示例显示以假定的配置示例使用的顺序启动域。使用 `ldm start-domain` 命令以 `root-dom0`、`root-dom1`、`guest0` 和 `guest1` 的顺序对其进行启动，使用 `ldm list-domain` 命令检查是否已启动这些域。如果 `[STATE]` 是 "active" 并且 `FLAGS` 字符串从左侧起第二个字符是 "n"，表明 Oracle Solaris 已经启动。

```
# ldm start-domain root-dom0
LDom root-dom0 started
# ldm start-domain root-dom1
LDom root-dom1 started
# ldm start-domain guest0
LDom guest0 started
# ldm start-domain guest1
LDom guest1 started
# ldm list-domain
NAME                STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary             active    -n-cv-   UART     64      56G       0.2%    15m
```

guest0	active	-n----	5002	64	64G	0.4%	18s
guest1	active	-n----	5003	64	64G	7.9%	9s
root-dom0	active	-n--v-	5000	32	32G	0.5%	24s
root-dom1	active	-n--v-	5001	32	32G	1.6%	16s

f. 将逻辑域配置信息保存到 **XSCF**。

恢复逻辑域配置信息后，执行 `ldm add-spconfig` 命令将所恢复的逻辑域配置信息保存到 XSCF。下面的示例将 `ldm-set4` 配置信息保存到 XSCF。

```
# ldm add-spconfig ldm-set4
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3
ldm-set4 [current]
```

g. 如果存在多项逻辑域配置信息需要恢复，请恢复出厂默认配置，然后重复步骤 **6.b** 至 **6.g**。

下面的示例显示恢复到出厂默认配置。
从 XSCF 执行 `showdomainconfig` 命令检查需要启动的逻辑域的配置信息。

```
XSCF> showdomainconfig -p 0
PPAR-ID :0
Booting config
(Current) :ldm-set1
(Next) :ldm-set1
-----
Index :1
config_name :factory-default
domains :1
date_created:-
-----
Index :2
config_name :ldm-set1
domains :8
date_created:'2012-08-0811:34:56'
-----
....
```

执行 `setdomainconfig` 命令在物理分区编号 0 (PPAR ID 0) 中指定出厂默认 Index1。

```
XSCF> setdomainconfig -p 0 -i 1
```

执行 `poweroff` 命令，然后执行 `poweron` 命令重新启动物理分区。启动完成后，重复步骤 6.b 至 6.g。


```
XSCF> poweroff -p 0
...
XSCF> poweron -p 0
...
```

A.2 对于从 XCP 2220 或更高版本的新安装，逻辑域配置无可可用硬件资源

“A.2 对于从 XCP 2220 或更高版本的新安装，逻辑域配置无可可用硬件资源”至“A.4 采用 XCP 2220 或更高版本新安装的系统从 1BB 至 2BB 配置的扩展”介绍具有 SPARC M10-4S 的 2BB 配置，使用物理分区动态重新配置的系统配置示例，并介绍构建新配置的步骤。

然而，需要说明的是假定 SPARC M10-4S 使用 XCP 2220 或更高版本作为其固件。也假定已经设置 XSCF，并且如《Fujitsu M10-4S/SPARC M10-4S 安装指南》所述，具有 2BB 配置的系统已经安装。

本节介绍在由两个 SPARC M10-4S，即两个系统板 (PSB<BB>) 组成的物理分区中，构建在 SPARC M10-4S 断开时移动资源所必需的硬件资源（CPU 内核和内存）没有可用空间配置的步骤示例。这部分还介绍了执行 SPARC M10-4S 活动状态更换的步骤示例。然而，在 SPARC M10-4S 活动状态更换时，这种情况将导致 CPU 内核数量和暂时分配给逻辑域的内存量的减少。

本介绍也适用于 SPARC M12。

A.2.1 配置示例

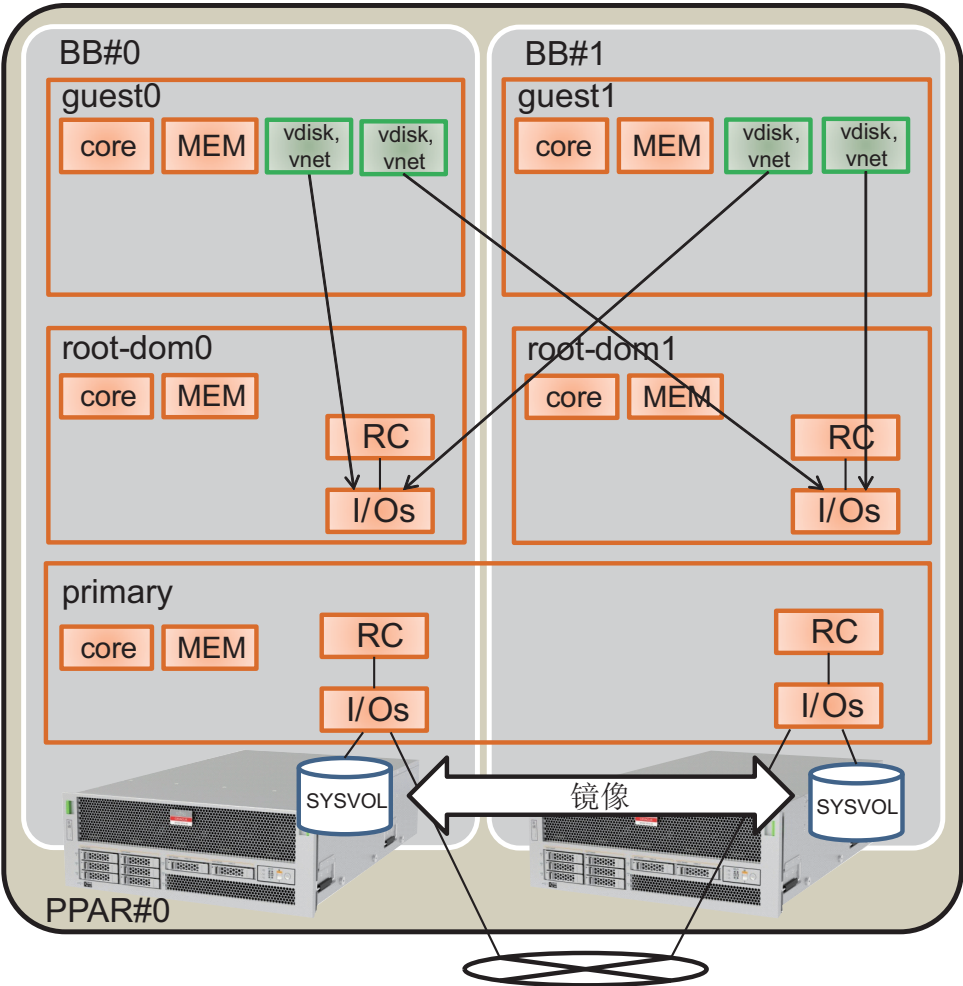
本节介绍由两个 SPARC M10-4S 即两个系统板 (PSB<BB>)（2BB 配置）组成逻辑域的配置示例。

如需在物理分区中启用物理分区动态重新配置，I/O 配置必须满足以下条件。

- 控制域的系统卷磁盘和网络接口必须通过连接各 SPARC M10-4S 的 I/O 进行冗余。这样操作的目的是：当一个 SPARC M10-4S 被移除时，使用剩余的 SPARC M10-4S 的 I/O 可继续操作。
- 有必要为各个 SPARC M10-4S 设置根域，并且将该根域用作专用域以提供虚拟 I/O 给来宾域。原因如下：如果根域正在用于任务，在 SPARC M10-4S 断开时，根域必须被停止，因此任务将停止。
- 任务必须在来宾域中被操作。必须将各个根域的虚拟 I/O 服务分配给此来宾域的虚拟 I/O 设备 (vdisk, vnet)，以便使那些虚拟 I/O 设备冗余。原因如下：即使在一侧断开 SPARC M10-4S，可以通过使用剩余 SPARC M10-4S 上的根域的虚拟 I/O 服务继续 I/O 访问。通过这种方式，断开 SPARC M10-4S 将不影响来宾域，可继续操作任务。

下面是满足上述条件的配置的示意图。而且，简化了各个逻辑域的 I/O 配置。

图 A-2 在 2BB 配置中可继续操作的配置示例



下表示例将说明在 2BB 配置中，需要分配给控制域、根域和来宾域的 CPU 内核、内存和 I/O 配置。不存在足够大的空资源，使得在 SPARC M10-4S 断开时启用分配给各个逻辑域的 CPU 内核和内存的移动。

表 A-1 无空硬件资源的 2BB 配置示例

逻辑域名称	primary	root-dom0	root-dom1	guest0	guest1
CPU 内核 (激活: 128)	32	16	16	32	32
内存 (总: 256 GB)	56 GB (*1)	32 GB	32 GB	64 GB	64 GB
物理 I/O (根合成体分配)	On-board #0 (PCIE0, PCIE4) On-board #1 (PCIE8, PCIE12)	PCIE1, PCIE2, PCIE3, PCIE5, PCIE6, PCIE7	PCIE9, PCIE10, PCIE11, PCIE13, PCIE14, PCIE15	-	-

表 A-1 无空硬件资源的 2BB 配置示例 (续)

逻辑域名称	primary	root-dom0	root-dom1	guest0	guest1
虚拟 I/O 设备	-	vds0 vsw0, vsw1	vds1 vsw10, vsw11	vdisk0, vdisk10 vnet0, vnet10	vdisk1, vdisk11 vnet1, vnet11

*1 在上述配置示例中，需要分配给控制域的内存容量被计算为分配给其他逻辑域后内存的余数。

首先，当获得通过从物理内存的总容量减去分配给除控制域之外的域的容量所获得的剩余容量时，它是 64 GB。

$256\text{ GB} - 32\text{ GB} - 32\text{ GB} - 64\text{ GB} - 64\text{ GB} = 64\text{ GB}$

实际分配给控制域的最大容量可以达到 60.75 GB，即从上面的结果中减去分配给虚拟机监控程序的内存容量 (2 GB + 1.25 GB) 所获得的容量。

$64\text{ GB} - 2\text{ GB} - 1.25\text{ GB} = 60.75\text{ GB}$

此外，当 Oracle VM Server for SPARC 的版本早于 3.2 时，在使用小于 60.75 GB 的最大容量进行计算时，按照执行物理分区动态重新配置的条件，为了使其成为倍数（内核数量 x 256 MB），它是 56 GB。

$60.75\text{ GB} / (32\text{ 内核} \times 256\text{ MB}) = 7.59$

为了成为整数倍，舍位至“7”。

$(32\text{ 内核} \times 256\text{ MB}) \times 7 = 56\text{ GB}$

注 - 您应该通过注意以下两点来设计分配给逻辑域的内存容量。有关详情，请参见 ["2.4.1 逻辑域配置注意事项"](#)。

- 分配给逻辑域的内存容量小于物理内存容量。
- 当 Oracle VM Server for SPARC 的版本早于 3.2 时，使用（内核的数量 x 256 MB）的倍数来启用物理分区动态重新配置。

A.2.2 物理分区配置步骤示例

1. 登录主 **XSCF**。
执行 **showbbstatus** 命令来检查您已经登录的 XSCF 是否是主 XSCF。
如果您已经登录到备用 XSCF，注销然后再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

2. 设置内存的镜像模式。
 - a. 执行 **showfru** 命令来检查 **SPARC M10-4S** 系统板 (**PSB<BB>**) 上内存的镜像模式。
下面是显示 PSB 00-0 下设备的设置信息示例。

```
XSCF> showfru sb 00-0
Device Location Memory Mirror Mode
sb 00-0
  cpu 00-0-0 no
  cpu 00-0-1 no
  cpu 00-0-2 no
  cpu 00-0-3 no
```

- b. 若需使用内存镜像模式，请执行 **setupfru** 命令来设置该模式。

如果您未正在使用内存镜像模式，将不需要此步骤。

有关内存镜像模式的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "14.1 配置内存镜像"。

下面的示例将 PSB 00-0 下的所有 CPU 都设置为内存镜像模式。

```
XSCF> setupfru -m y sb 00-0
```

注- 使用 SPARC M12 时，指定 -c mirror=yes 可对内存创建镜像配置。
示例: XSCF> **setupfru -c mirror=yes sb 00-0**

执行 showfru 命令检查内存镜像模式的设置。

```
XSCF> showfru sb 00-0
Device Location Memory Mirror Mode
sb 00-0
    cpu 00-0-0 yes
    cpu 00-0-1 yes
    cpu 00-0-2 yes
    cpu 00-0-3 yes
```

- 3. 创建物理分区配置信息。
 - a. 执行 **showpctl** 命令检查物理分区配置信息。

```
XSCF> showpctl -p 0
PPAR-ID LSB PSB Status
```

- b. 执行 **setpctl** 命令在物理分区配置信息中登录系统板。
 - 执行 setpctl 命令在内置目标的物理分区配置信息中登录 SPARC M10-4S 系统板。
 - 在下面的示例中，物理系统板（PSB 00-0 和 PSB 01-0）映射到物理分区 0 的逻辑系统板（LSB 00 和 LSB 01）。

```
XSCF> setpctl -p 0 -a 00=00-0 01=01-0
```

- c. 执行 **showpctl** 命令检查物理分区配置信息。
 - 检查物理分区配置信息。

```
XSCF> showpctl -p 0
PPAR-ID      LSB      PSB      Status
00
              00      00-0
              01      01-0
```

指定 -v 选项显示配置策略、IO 无效选项 (No-IO) 和内存无效选项 (No-Mem) 的详细信息。

XSCF> showpcl -v -p 0						
PPAR-ID	LSB	PSB	Status	No-Mem	No-IO	Cfg-policy
00			Running			
						System
	00	00-0		False	False	
	01	01-0		False	False	

使用 setpcl 命令变更配置策略、IO 无效选项 (No-IO) 和内存无效选项 (No-Mem) 的设置。

有关 setpcl 命令的详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

4. 将系统板 (PSB<BB>) 分配给物理分区。

a. 执行 showboards-a 命令检查 PSB 状态。

执行 showboards -a 命令检查各 PSB 的状态是否是 "SP"（系统板池）。

XSCF> showboards -a							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	SP	Available	n	n	n	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal

b. 执行 addboard -c assign 命令分配 PSB。

XSCF> addboard -c assign -p 0 00-0 01-0							
---	--	--	--	--	--	--	--

c. 执行 showboards-p 命令检查 PSB 状态。

执行 showboards -p 命令检查分配给物理分区的各 PSB 的状态。

由于各 PSB 已经被正常分配给物理分区 0，此示例检查各 PSB 的 [Assignment] 字段是否成为 "Assigned"。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	n	n	n	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

5. 登录 CPU 激活密钥以便分配 CPU 内核资源。

a. 执行 showcodactivation 命令，检查 CPU 激活密钥上的信息。

执行 showcodactivation 命令，检查物理分区是否包含可分配的 CPU 激活密钥。

如果仅显示标题，则 CPU 激活密钥未在 XSCF 中登录。

```
XSCF> showcodactivation
Index Description Count
-----
```

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。

b. 执行 **addcodactivation** 命令以登录 CPU 激活密钥。

有关如何登录 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "5.3 添加 CPU 内核资源"。

```
XSCF> addcodactivation "Product: SPARC M10-4S
SequenceNumber:10005
Cpu: noExpiration 2
Text-Signature-SHA256-RSA2048:
PSSrElBrse/r69AVSVFd38sT6AZm2bxeUDdPQHKbtxgvZPsrtYguqiNUieB+mTDC
:
:
b1GCKFx1RH27FdVHiB2H0A=="
AboveKeywillbeadded,Continue?[y|n]:y
```

c. 执行 **showcodactivation** 命令，检查 CPU 激活密钥上的信息。

执行 showcodactivation 命令，检查物理分区是否包含可分配的 CPU 激活密钥。

```
XSCF> showcodactivation
Index Description Count
-----
0 PROC 2
1 PROC 2
2 PROC 2
3 PROC 2

--- 省略 ---

62 PROC 2
63 PROC 2
```

d. 执行 **setcod** 命令分配 CPU 内核资源。

执行 setcod 命令将 CPU 内核资源分配给物理分区。

以下是将 128 个 CPU 内核资源分配到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c set 128
PROC Permits assigned for PPAR 0 : 0 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y
```

```
Completed.
```

注 -XCP 2250 和更早版本的 XSCF 固件不支持 `-c add`、`-c delete` 和 `-c set` 选项。按如下所示指定 `setcod` 命令的选项可交互式添加和删除 CPU 内核资源。

```
XSCF> setcod -s cpu
```

执行 `showcod` 命令检查分配给物理分区的 CPU 内核资源的信息。

下面的示例，使用刚刚执行的 `setcod` 命令，证实 128 个 CPU 内核资源被分配给物理分区 0。

```
XSCF> showcod -p 0  
PROC Permits assigned for PPAR 0: 128
```

6. 执行 **resetdateoffset** 命令以 XSCF 管理的时间重置差异。

执行 `resetdateoffset` 命令来重置 XSCF 管理的时间和物理分区管理的时间之间的差异。

```
XSCF> resetdateoffset -p 0
```

7. 执行 **showpparmode** 命令来检查诊断消息的详细级别的设置以及 PPAR DR 模式的设置。

执行 `showpparmode` 命令来检查诊断信息的详细级别 (Message Level) 是否是 "normal" (标准) 以及 PPAR DR 模式的 Next 是否设置为 "on" (启用)。

```
XSCF> showpparmode -p 0  
Host-ID : 9007002b  
Diagnostic Level : min  
Message Level : normal  
Alive Check : on  
Watchdog Reaction : reset  
Break Signal : on  
Autoboot (Guest Domain) : on  
Elastic Mode : off  
IOreconfigure : false  
PPAR DR (Current) : -  
PPAR DR (Next) : on
```

如果诊断信息的详细级别是除 "normal" 之外的级别，执行 `setpparmode` 命令可将其设置为 "normal"。

有关 `setpparmode` 命令的详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

```
XSCF> setpparmode -p 0 -m message=normal
```

如果 PPAR DR 模式设置为 "off" (禁用)，执行 `setpparmode` 命令可将其设置为

"on".

```
XSCF> setpparmode -p 0 -m ppar_dr=on
```

8. 执行 **poweron** 命令来开启物理分区的电源。

```
XSCF> poweron -p 0
```

9. 执行 **console** 命令将控制台连接到物理分区。

```
XSCF> console -p 0
```

10. 安装 **Oracle Solaris** 和 **Oracle VM Server for SPARC**。

在控制域上安装 Oracle Solaris 和 Oracle VM Server for SPARC。

有关物理分区动态重新配置所要求的 Oracle Solaris 的版本和条件的详情，请参见表 1-3 和 表 1-4。

有关安装的详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com/>) 上的以下文档。

- Oracle Solaris 10

《Oracle Solaris 10 1/13 Installation Guide》

- Oracle Solaris 11

《Installing Oracle Solaris 11.x Systems》

- Oracle VM Server for SPARC

《Oracle VM Server for SPARC Administration Guide》中的"Installing and Enabling Software"

11. 配置逻辑域。

此项介绍如何配置在 "表 A-1 无空硬件资源的 2BB 配置示例" 中定义的逻辑域。

- a. 减少控制域资源的数量。

对于出厂默认配置而言，所有 CPU 内核、内存和 PCIe 根合成体都被分配给控制域 (**primary**)。如需将这些资源分配给其他逻辑域，首先从控制域删除某些资源并配置控制台服务。

下面的示例显示了命令的执行。

执行 **ldm start-reconf** 命令切换到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

使用 **ldm remove-io** 命令移除根合成体。

下面的示例部分地说明了按照配置示例移除 PCIE1、PCIE2、PCIE3、PCIE5、

PCIE6、PCIE7、PCIE9、PCIE10、PCIE11、PCIE13、PCIE14 和 PCIE15 的命令。

```
# ldm remove-io PCIE1 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
....
# ldm remove-io PCIE15 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

通过使用 `ldm set-core` 和 `ldm set-memory` 命令指定小于初始容量的容量来减少分配给控制域的 CPU 内核数量和内存容量。

下面举例说明按照配置示例，将控制域的 CPU 内核数量设置为 32，以及将内存容量设置为 56 GB。

```
# ldm set-core 32 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm set-memory 56G primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

注 - 建议首先使用 `ldm set-core` 命令配置 CPU 内核，然后使用 `ldm set-memory` 命令配置内存。这有利于集合内存的连续区域的分配。这也易于可进行物理分区动态重新配置的 CPU 内核配置条件的满足，以及内存配置条件的满足。有关 CPU 内核和内存的配置条件的详情，请参见“[2.5.2 动态重新配置的系统操作注意事项](#)”中的“[CPU 内核和内存的配置](#)”。

使用 `ldm add-vconscon` 命令创建称为 `vcc0` 的虚拟控制台终端集中器服务，然后使用 `svcadm` 命令启动虚拟网络终端服务器 (`vntsd`) 的守护程序。通过此 `vcc0` 配置连接到各逻辑域的控制台。

```
# ldm add-vconscon port-range=5000-5200 vcc0 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# svcadm enable vntsd
```

保存配置信息，然后重新启动 Oracle Solaris。

下面的示例检查使用 `ldm list-spconfig` 命令保存的配置信息，然后使用 `ldm`

add-spconfig 命令以名称为 ldm-set1 保存该配置。然后再次使用 ldm list-spconfig 命令检查配置是否已经保存，最后举例说明重新启动 Oracle Solaris。

```
# ldm list-spconfig
factory-default [current]
# ldm add-spconfig ldm-set1
# ldm list-spconfig
factory-default
ldm-set1 [current]
# shutdown -i6 -g0 -y
```

b. 创建控制域系统卷的冗余配置。

创建控制域系统卷的冗余配置。此项目介绍使用 ZFS 创建镜像配置的命令示例。有关详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com>) 上的以下文档。

- Oracle Solaris 10 的情形

《Oracle Solaris ZFS Administration Guide》上的 "How to Create a Mirrored Root Pool (Post Installation)"

- Oracle Solaris 11 的情形

《Oracle Solaris 11.x Administration: ZFS File Systems》中的 "How to Configure a Mirrored Root Pool (SPARC or x86/VTOC)"

如需使用其他冗余配置软件，请参见该软件的手册。

执行 zpool status 命令检查根池的状态。

下面的示例表明已经将 c2t50000393E802CCE2d0s0 分配给默认根池 (rpool)。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: none requested
config:
      NAME                                STATE      READ  WRITE CKSUM
      rpool                                ONLINE      0     0     0
        c2t50000393E802CCE2d0s0          ONLINE      0     0     0
errors: No known data errors
```

执行 format 命令检查可添加哪个磁盘。

下面的示例表明 c3t50000393A803B13Ed0 作为其他磁盘而存在。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
    0. c2t50000393E802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8000/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
        /dev/chassis/FUJITSU-BBEXP.500000e0e06d027f/0123_HDD00/disk
    1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
```

```
sec 625>
    /pci@8800/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
    /dev/chassis/FUJITSU-BBEXP.500000e0e06d243f/022U_HDD01/disk
Specify disk (enter its number): ^C
```

执行 `zpool attach` 命令将第二个磁盘添加到 `rpool`，以便提供镜像配置。

下面的示例使用 `zpool attach` 命令添加 `c3t50000393A803B13Ed0s0`，然后使用 `zpool status` 命令检查同步处理 (resilver) 的状态。通过参照状态和动作，您可以确定正在进行同步处理。定期执行 `zpool status` 命令来检查同步处理的状态，直到处理结束。

```
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
       continue to function in a degraded state.
action: Wait for the resilver to complete.
       Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Wed Jan 29 21:35:39 2014
      3.93G scanned out of 70.6G at 71.9M/s, 0h15m to go
      3.90G resilvered, 5.56% done
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                                DEGRADED   0     0     0
        mirror-0                            DEGRADED   0     0     0
          c2t50000393E802CCE2d0s0          ONLINE    0     0     0
          c3t50000393A803B13Ed0s0          DEGRADED   0     0     0 (resilvering)
```

当完成同步处理时，[state] 将被设置为 "ONLINE"，如下所示。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h10m with 0 errors on Wed Jan 29 21:45:42 2014
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                                ONLINE     0     0     0
        mirror-0                            ONLINE     0     0     0
          c2t50000393E802CCE2d0s0          ONLINE    0     0     0
          c3t50000393A803B13Ed0s0          ONLINE    0     0     0
errors: No known data errors
```

c. 配置根域。

本项目介绍配置根域的步骤。

执行 `ldm add-domain` 命令添加名为 `root-dom0` 的逻辑域。

```
# ldm add-domain root-dom0
```

执行 `ldm set-variable` 命令将被设计为自动启动 Oracle Solaris 的 OpenBoot

PROM 环境变量 "auto-boot?" 变更为 "false"（禁用）。默认状态下，此设置为 "true"（启用）。因此，当未安装 Oracle Solaris 时，OpenBoot PROM 尝试自动启动 Oracle Solaris。安装 Oracle Solaris 前，变更此设置为禁用将有利于将要执行的操作。

```
# ldm set-variable auto-boot\?=false root-dom0
```

首先，使用 ldm set-core 命令分配 CPU 内核，然后使用 ldm set-memory 命令分配内存。

下面的示例根据配置示例，使用 ldm set-core 命令分配 16 个 CPU 内核，使用 ldm set-memory 命令分配 32 GB 内存。

```
# ldm set-core 16 root-dom0
# ldm set-memory 32G root-dom0
```

注 - 建议首先使用 ldm set-core 命令配置 CPU 内核，然后使用 ldm set-memory 命令配置内存。这有利于集合内存的连续区域的分配。这也易于可进行物理分区动态重新配置的 CPU 内核配置条件的满足，以及内存配置条件的满足。有关 CPU 内核和内存的配置条件的详情，请参见 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中的 ["CPU 内核和内存的配置"](#)。

执行 ldm set-vconsole 命令分配虚拟控制台 (vcons)。

下面的示例执行 ldm set-vconsole 命令将控制域中虚拟控制台终端集中器的服务 (vcc0) 的端口号 5000 分配给虚拟控制台。

```
# ldm set-vconsole service=vcc0 port=5000 root-dom0
```

下面的示例执行 ldm list-io -l 命令显示 PCI 分配状态。

NAME 以 "/BB0" 开始所在行和 [TYPE] 栏有 "PCIE" 表示 SPARC M10-4S0 (BB0) 上的 PCIe 端点。[DOMAIN] 栏所在行为空表示未分配的 PCIe 端点，有关的根合成体显示在 [BUS] 栏。

因此，您可以快速了解到 PCIE1、PCIE2、PCIE3、PCIE5、PCIE6 和 PCIE7 是 BB0 上未分配的根合成体。

# ldm list-io -l				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
(省略)				
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
[pci@8000/pci@4/pci@0/pci@9]				
network@0				
network@0,1				
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
[pci@8000/pci@4/pci@0/pci@0]				
scsi@0/iport@f/disk@w50000393e802cce2,0				
scsi@0/iport@f/disk@w50000393d8285226,0				
scsi@0/iport@f/smp@w500000e0e06d027f				

```

        scsi@0/iport@f/enclosure@w500000e0e06d027d,0
        scsi@0/iport@v0
/BB0/PCI0                                PCIE    PCIE1            EMP
[pci@8100/pci@4/pci@0/pci@0]
/BB0/PCI3                                PCIE    PCIE2            EMP
[pci@8200/pci@4/pci@0/pci@0]
/BB0/PCI4                                PCIE    PCIE2            EMP
[pci@8200/pci@4/pci@0/pci@8]
/BB0/PCI7                                PCIE    PCIE3            EMP
[pci@8300/pci@4/pci@0/pci@0]
/BB0/PCI8                                PCIE    PCIE3            EMP
[pci@8300/pci@4/pci@0/pci@8]
/BB0/CMUL/NET2                           PCIE    PCIE4      primary OCC
[pci@8400/pci@4/pci@0/pci@a]
        network@0
        network@0,1
/BB0/PCI1                                PCIE    PCIE5            EMP
[pci@8500/pci@4/pci@0/pci@8]
/BB0/PCI2                                PCIE    PCIE5            EMP
[pci@8500/pci@4/pci@0/pci@9]
/BB0/PCI5                                PCIE    PCIE6            EMP
[pci@8600/pci@4/pci@0/pci@9]
/BB0/PCI6                                PCIE    PCIE6            EMP
[pci@8600/pci@4/pci@0/pci@11]
/BB0/PCI9                                PCIE    PCIE7            EMP
[pci@8700/pci@4/pci@0/pci@9]
/BB0/PCI10                               PCIE    PCIE7            EMP
[pci@8700/pci@4/pci@0/pci@11]
(省略)

```

请参见显示在上述结果中的设备路径（显示为 [pci@....] 的字符串）和《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "A.6 SPARC M10-4S 设备路径"，确认哪一个根合成体需要分配给根域。

在配置示例中，上述检查的 BB0 上所有未分配的根合成体 (PCIE1、PCIE2、PCIE3、PCIE5、PCIE6 和 PCIE7) 将被分配。因此，执行 `ldm add-io` 命令将其分配给 `root-dom0`。

下面的示例显示命令的执行。

```

# ldm add-io PCIE1 root-dom0
# ldm add-io PCIE2 root-dom0
# ldm add-io PCIE3 root-dom0
# ldm add-io PCIE5 root-dom0
# ldm add-io PCIE6 root-dom0
# ldm add-io PCIE7 root-dom0

```

使用 `ldm bind-domain` 命令将根域设置为 `bound` 状态，然后执行 `ldm list-io` 命令检查根合成体是否已经被分配。

下面的示例使用 `ldm bind-domain` 命令检查 `root-dom0` 是否绑定，使用 `ldm list-io` 命令检查根合成体是否已经被分配。

[TYPE] 栏是 "BUS" 并且 [DOMAIN] 栏是 "root-dom0" 的行表示分配给 `root-dom0` 的根合成体。该行中的 BUS 是已分配的根合成体的名称。

在下面的示例中，您可以确认已经将 PCIE1、PCIE2、PCIE3、PCIE5、PCIE6 和 PCIE7 分配给 root-dom0。

```
# ldm bind-domain root-dom0
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8	primary	IOV

(省略)

根据步骤 11. c，配置其他 SPARC M10-4S 的根域（示例中为 root-dom1）。

d. 在根域中安装 Oracle Solaris。

本项目介绍在根域中安装 Oracle Solaris 的步骤。

执行 ldm start-domain 命令启动根域 root-dom0。

```
# ldm start-domain root-dom0
LDom root-dom0 started
```

执行 ldm set-variable 命令将被设计为自动启动 Oracle Solaris 的 OpenBoot PROM 环境变量 "auto-boot?" 变更为 "true"（启用）。安装 Oracle Solaris 后，执行 ldm start-domain 命令也可启动 Oracle Solaris。

```
# ldm set-variable auto-boot?=true root-dom0
```

执行 telnet 命令连接到根域的控制台。

下面的示例通过执行 ldm list-domain 命令确认 root-dom0 的端口号是 5000。通过使用 telnet 命令连接到 localhost 的端口号 5000 也可以确认 root-dom0 在 OpenBoot PROM (OBP) 状态停止。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.0%	7h 7m
root-dom0	active	-t----	5000	32	32G	0.0%	20s
root-dom1	bound	-----	5001	32	32G		

```
# telnet localhost 5000
....
{0} ok
```

在根域中安装 Oracle Solaris。

有关物理分区动态重新配置所要求的 Oracle Solaris 的版本和条件的详情，请参见表 1-3 和表 1-4。

下面的示例通过网络执行命令启动 Oracle Solaris 11 安装。

```
{0} ok boot net:dhcp
....
```

有关安装的详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com/>) 上的以下文档。

- Oracle Solaris 10

 - 《Oracle Solaris 10 1/13 Installation Guide》

- Oracle Solaris 11

 - 《Installing Oracle Solaris 11.x Systems》

根据步骤 11. d，采用相同的方式，安装其他根域（示例中为 root-dom1）。

e. 将虚拟 I/O 服务分配给根域。

此项目介绍将虚拟 I/O 服务设置给根域的步骤示例。

此示例显示将根域的整个物理磁盘借为虚拟磁盘服务 (vds)。

此示例显示将以太网卡端口分配给虚拟交换网络设备 (vsw)。

有关虚拟 I/O 服务的详情，请参见 Oracle 公司出版的《Oracle VM Server for SPARC Administration Guide》。

以 root（根）权限登录根域。

```
root-dom0 console login: root
Password:
....
```

如需指定需要分配给虚拟磁盘服务 (vds) 的后端设备（物理磁盘），请执行 format 命令显示磁盘和其磁盘路径。

在格式 cXtXXXXXXXXXXXXXXXXdX 中的字符串表示磁盘，以 "/pci@" 开始的字符串表示设备路径。

```
root-dom0# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
    0. c2t50000393A802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8100/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
    1. c3t50000393D8285226d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8200/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393d8285226,0
    2. c4t50000393A804B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@3200/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
....
```

请参见显示在上述结果中的设备路径和《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "A.6 SPARC M10-4S 设备路径"，并检查磁盘的物理位置。然后，确定需要分配给虚拟磁盘服务的后端磁盘 (cXtXXXXXXXXXXXXXXXXdX)。

配置示例分配了两个磁盘 (c3t50000393D8285226d0 和 c4t50000393A804B13Ed0) 作为来宾域 (guest0, guest1) 的虚拟磁盘的后端。

接下来, 若需指定需要分配给虚拟交换服务 (vsw) 的网络接口, 请执行 `dladm show-phys` 命令显示与 PCIe 插槽的位置 (LOC) 相关联的网络接口名 (LINK)。从显示的结果, 检查网络接口的位置并确认需要分配给虚拟交换服务的网络接口。

配置示例将两个网络接口 (net1 和 net2) 分配给与各来宾域的虚拟网络相关联的虚拟交换机。

```
root-dom0# dladm show-phys -L
LINK      DEVICE      LOC
net0      igb0        BB#0-PCI#0
net1      igb1        BB#0-PCI#1
net2      igb2        BB#0-PCI#2
net3      igb3        BB#0-PCI#3
...
```

执行 `ldm add-vdiskserver` 命令将虚拟磁盘服务添加到根域。

下面的示例使用 `ldm add-vdiskserver` 命令将虚拟磁盘服务 (vds0) 添加到根域 (root-dom0)。

```
# ldm add-vdiskserver vds0 root-dom0
```

执行 `ldm add-vdiskserverdevice` 命令从根域导出虚拟磁盘的后端。

下面的示例执行 `ldm add-vdiskserverdevice` 命令将两个后端磁盘分配给 vds0 以便将其导出。

```
# ldm add-vdiskserverdevice /dev/dsk/c3t50000393D8285226d0s2 vol10@vds0
# ldm add-vdiskserverdevice /dev/dsk/c4t50000393A804B13Ed0s2 vol11@vds0
```

执行 `ldm add-vswitch` 命令将虚拟交换机添加到根域。

下面的示例添加了 vsw0 和 vsw1, 以便添加各来宾域的虚拟交换机, 并将不同的物理网络接口 (net1, net2) 分配给各来宾域。

```
# ldm add-vswitch net-dev=net1 vsw0 root-dom0
# ldm add-vswitch net-dev=net2 vsw1 root-dom0
```

同样, 将虚拟 I/O 服务添加到分配 BB1 侧上的 SPARC M10-4S 的 I/O 的根域 (root-dom1)。

分配以下虚拟磁盘服务名和虚拟交换机名给 root-dom1 当作示例。有关步骤的详情, 请参见 11.e。

虚拟磁盘服务名: vds1 (分配 vol10 和 vol11 作为两个后端卷的名称。)

虚拟交换机名: vsw10, vsw11

f. 配置来宾域。

本项目介绍配置来宾域的步骤。

执行 `ldm add-domain` 命令添加名为 `guest0` 的逻辑域。

```
# ldm add-domain guest0
```

执行 `dm set-variable` 命令将被设计为自动启动操作系统的 OBP 环境变量 "auto-boot?" 变更为 "false" (禁用)。默认状态下, 此设置为 "true" (启用)。因此, 当未安装 Oracle Solaris 时, OpenBoot PROM 尝试自动启动 Oracle Solaris。安装 Oracle Solaris 前, 变更此设置为禁用将有利于将要执行的操作。

```
# ldm set-variable auto-boot\?=false guest0
```

首先, 使用 `ldm set-core` 命令分配 CPU 内核, 然后使用 `ldm set-memory` 命令分配内存。

下面的示例根据配置示例, 使用 `ldm set-core` 命令分配 32 个 CPU 内核, 使用 `ldm set-memory` 命令分配 64 GB 内存。

```
# ldm set-core 32 root-dom0
# ldm set-memory 64G root-dom0
```

注 - 建议首先使用 `ldm set-core` 命令配置 CPU 内核, 然后使用 `ldm set-memory` 命令配置内存。这有利于集合内存的连续区域的分配。这也易于可进行物理分区动态重新配置的 CPU 内核配置条件的满足, 以及内存配置条件的满足。有关 CPU 内核和内存的配置条件的详情, 请参见 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中的 ["CPU 内核和内存的配置"](#)。

执行 `ldm set-vconsole` 命令分配虚拟控制台 (vcons)。

下面的示例执行 `ldm set-vconsole` 命令将控制域中虚拟控制台终端集中器的服务 (vcc0) 的端口号 5100 分配给虚拟控制台。

```
# ldm set-vconsole service=vcc0 port=5100 guest0
```

执行 `ldm add-vdisk` 命令分配虚拟磁盘 (vdisk)。

下面的示例按照配置示例分配了两个虚拟磁盘 (vdisk0, vdisk10)。各虚拟磁盘的后端指定添加到两个根域 (root-dom0, root-dom1) 的后端 (vds0 vol0, vds1 vol10)。

```
# ldm add-vdisk vdisk0 vol10@vds0 guest0
# ldm add-vdisk vdisk10 vol110@vds1 guest0
```

执行 `ldm add-vnet` 命令分配虚拟网络设备 (vnet)。

下面的示例按照配置示例分配了两个虚拟设备 (vnet0, vnet10)。连接到各虚拟网络设备的虚拟交换机指定添加到两个根域 (root-dom0, root-dom1) 的虚拟交换机 (vsw0, vsw10)。

```
# ldm add-vnet vnet0 vsw0 guest0
# ldm add-vnet vnet10 vsw10 guest0
```

注 - 如果分配多个虚拟磁盘或虚拟网络设备，请从 ldm list -l 命令的执行结果记录分配给虚拟设备的 ID 值。动态地被删除后，若需再次添加虚拟设备，需要此 ID，以便虚拟设备路径保持不变。

g. 在来宾域中安装 **Oracle Solaris**。

本项目介绍在来宾域中安装 Oracle Solaris 的步骤。

执行 ldm bind-domain 命令绑定来宾域，然后通过执行 start-domain 命令将其启动。

下面的示例显示启动 guest0 的命令的执行。

```
# ldm bind-domain guest0
# ldm start-domain guest0
LDom guest0 started
```

执行 ldm set-variable 命令将被设计为自动启动 Oracle Solaris 的 OpenBoot PROM 环境变量 "auto-boot?" 变更为 "true"（启用）。安装 Oracle Solaris 后，执行 ldm start-domain 命令也可启动 Oracle Solaris。

```
# ldm set-variable auto-boot\?=true guest0
```

执行 telnet 命令连接到来宾域的控制台。

下面的示例通过执行 ldm list-domain 命令确认 guest0 控制台的端口号是 "5100"。通过使用 telnet 命令连接到 localhost 的端口号 "5100" 也可以确认 guest0 在 OpenBoot PROM 状态停止。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active     -n-cv-   UART    64     56G     0.0%  8h 7m
guest0        active     -t----   5100    64     64G     0.0%  20s
root-dom0     active     -n--v-   5000    32     32G     0.0%  43s
root-dom1     active     -n--v-   5001    32     32G     0.0%  20s
guest1        inactive   ------   64     64G
# telnet localhost 5100
....
{0} ok
```

有关物理分区动态重新配置所要求的 Oracle Solaris 的版本和条件的详情，请参见 [表 1-3](#) 和 [表 1-4](#)。

下面的示例通过网络执行命令启动 Oracle Solaris 11 安装。

```
{0} ok boot net:dhcp
....
```

有关安装的详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com/>) 上的以下文档。

- Oracle Solaris 10

《Oracle Solaris 10 1/13 Installation Guide》

- Oracle Solaris 11

《Installing Oracle Solaris 11.x Systems》

根据步骤 11. g，采用相同的方式，安装其他来宾域（示例中为 guest1）。

h. 创建来宾域的虚拟 I/O 的冗余配置。

下面介绍使用 IPMP 创建分配给来宾域 guest0 的两个虚拟网络接口 (vnets) 的冗余配置步骤示例。有关其他冗余配置的步骤的详情，请参见各自冗余配置软件的文档。

登录来宾域 guest0。

在下面的示例中，使用 `ldm list-domain` 命令检查 guest0 的控制台的端口号，使用 `telnet` 命令连接到端口号 "5100"。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART     16      14G       0.0%    8h 7m
guest0        active    -n----   5100     32      32G       0.0%    20s
guest1        active    -n----   5101     32      32G       0.0%    19s
root-dom0     active    -n--v-   5000     24      24G       0.0%    43s
root-dom1     active    -n--v-   5001     24      24G       0.0%    20s
# telnet localhost 5100
....
guest0 console login: root
Password:
...
guest0#
```

执行 `dladm` 命令确认虚拟网络设备可见。

在下面的示例中,可把虚拟网络设备称为网络接口 `net0` 和 `net1` 。

```
guest0# dladm show-phys
LINK          MEDIA      STATE      SPEED    DUPLEX    DEVICE
net0          Ethernet  up         0        unknown  vnet0
net1          Ethernet  up         0        unknown  vnet1
```

本节介绍有关 Oracle Solaris 11 的步骤。执行 `ipadm show-if` 命令确认 `net0` 和 `net1` 未显示。

```
guest0# ipadm show-if
IFNAME      CLASS      STATE      ACTIVE OVER
lo0          loopback ok        yes    --
```

执行 ipadm create-ip 命令创建 IP 接口 net0 和 net1，然后使用 ipadm show-if 命令确认已经将其正常创建。

```
guest0# ipadm create-ip net0
guest0# ipadm create-ip net1
guest0# ipadm show-if
IFNAME      CLASS      STATE      ACTIVE OVER
lo0          loopback ok        yes    --
net0         ip         down       no     --
net1         ip         down       no     --
```

执行 ipadm create-ipmp 命令创建 IPMP 接口 ipmp0，然后执行 ipadm add-ipmp 命令将 IP 接口 net0 和 net4 添加到 IPMP 组。

```
guest0# ipadm create-ipmp ipmp0
guest0# ipadm add-ipmp -i net0 -i net1 ipmp0
```

执行 ipadm create-addr 命令将 IP 地址分配给 IPMP 接口 ipmp0，然后使用 ipadm show-addr 命令检查此设置。在下面的示例中，分配了固定 IP 地址。

```
guest0# ipadm create-addr -T static -a local=xx.xx.xx.xx/24 ipmp0/v4
guest0# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
ipmp0/v4     static    ok         xx.xx.xx.xx/24
lo0/v6       static    ok         ::1/128
```

执行 ipadm set-ifprop 命令设置备用接口，然后使用 ipmpstat -i 命令检查 IPMP 配置。

```
guest0# ipadm set-ifprop -p standby=on -m ip net1
guest0# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net1       no      ipmp0      is----- up         disabled  ok
net0       yes     ipmp0      --mbM--   up         disabled  ok
```

对其他来宾域（本示例中为 guest1）执行相同的步骤。

i. 检查内存资源分配状况。

此项目介绍检查和调整分配给各逻辑域的内存位置的步骤。

如需使用物理分区动态重新配置使 SPARC M10-4S 断开连接，需要满足 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中的 ["CPU 内核和内存的配置"](#) 所述的内存配置条件。

另一方面，本配置示例显示了其中空资源未被分配的配置。这些空资源提前用

作内存块的移动目标。因此，根据 "A.2.3 活动状态更换的步骤示例（适用于 Oracle VM Server for SPARC 3.1.x）" 中所述的步骤，提前调整内存块配置很重要，以便于考虑内存配置。

下面介绍示例的步骤。

执行 `ldm list-devices -a` 命令检查分配给各逻辑域的内存的连续区域（内存块）的状态。然后，检查是否存在大量的小内存块。

可将检查是否存在大量的 SIZE 约 256 MB 到 512 GB 的内存块作为参考。

下面的示例显示 `ldm list-devices -a memory` 命令的执行。

```
# ldm list-devices -a memory
MEMORY
      PA                               SIZE                               BOUND
0x7000000000000000                    4G                               root-dom0
0x7001000000000000                    4G                               root-dom1
0x7002000000000000                    8G                               root-dom0
0x7004000000000000                    4G                               root-dom1
0x7005000000000000                    4G                               root-dom0
0x7006000000000000                    8G                               root-dom1
0x7200000000000000                    8G                               root-dom0
0x7202000000000000                    8G                               root-dom0
0x7204000000000000                    8G                               root-dom1
0x7206000000000000                    8G                               root-dom1
0x7400000000000000                   32G                               guest0
0x7600008000000000                  1272M                               _sys_
0x7600500000000000                 31488M                               guest1
0x7800000000000000                   16G                               guest0
0x7804000000000000                   16G                               guest1
0x7a00000000000000                   16G                               guest0
0x7a04000000000000                   16G                               guest1
0x7c00000000000000                   28G                               primary
0x7c07000000000000                  256M                               guest1
0x7c07100000000000                  3840M
0x7e00008000000000                  1272M                               _sys_
0x7e00500000000000                   512M                               _sys_
0x7e00700000000000                   256M                               _sys_
0x7e00800000000000                    8G                               primary
0x7e02800000000000                   20G                               primary
0x7e07800000000000                    2G                               guest1
```

在上述示例中，`root-dom0` 和 `root-dom1` 的内存块被分解为相对小的内存块。因此，按照步骤 11.j 中所述重新分配内存块。

如下所示，如果各逻辑域的内存块未被分解，不必执行重新配置。转到步骤 11.k。

```
# ldm list-devices -a memory
MEMORY
      PA                               SIZE                               BOUND
0x7000000000000000                   32G                               root-dom0
0x7200000000000000                   32G                               root-dom1
0x7400000000000000                   32G                               guest0
```

0x760000800000	1272M	_sys_
0x760050000000	2G	primary
0x7600d0000000	29440M	guest1
0x780000000000	32G	guest0
0x7a0000000000	32G	guest1
0x7c0000000000	32G	primary
0x7e0000800000	1272M	_sys_
0x7e0050000000	512M	_sys_
0x7e0070000000	256M	_sys_
0x7e0080000000	3328M	guest1
0x7e0150000000	4864M	
0x7e0280000000	22G	primary

j.调整逻辑域的内存块配置。

从步骤 11. i 执行的检查结果可以得出，如果逻辑域的内存块被分解，则重新分配已分配给逻辑域的资源。下面是步骤的示例。

使用 `ldm stop-domain` 命令停止各逻辑域，使用 `ldm unbind-domain` 命令使其处于 `inactive` 状态。

注 - 假设物理 I/O 分配（根合成体，PCIe 端点，SR-IOV Virtual Function）之间存在从属关系以及存在虚拟 I/O 从属关系。按照分配虚拟 I/O 的来宾域、PCIe 端点及分配 SR-IOV VF 的 I/O 域、根域的顺序进行停止和取消绑定。

下面的示例显示使 `guest0`、`guest1`、`root-dom0` 和 `root-dom1` 处于 `inactive` 的命令。

```
# ldm stop-domain guest0
LDom guest0 stopped
# ldm stop-domain guest1
LDom guest1 stopped
# ldm stop-domain root-dom0
LDom root-dom0 stopped
# ldm stop-domain root-dom1
LDom root-dom1 stopped
# ldm unbind-domain guest0
# ldm unbind-domain guest1
# ldm unbind-domain root-dom0
# ldm unbind-domain root-dom1
```

使用 `ldm list-domain` 命令，检查是否处于 `inactive` 状态。

下面的示例表明 `guest0`、`guest1`、`root-dom0` 和 `root-dom1` 已经处于 `inactive` 状态。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.2%	5h 28m
guest0	inactive	-----		64	64G		
guest1	inactive	-----		64	64G		
root-dom0	inactive	-----		32	32G		
root-dom1	inactive	-----		32	32G		

使用 `ldm start-reconf` 命令，从控制域切换到延迟重新配置模式。

```
# ldm start-reconf primary
```

```
Initiating a delayed reconfiguration operation on the primary domain.  
All configuration changes for other domains are disabled until the primary  
domain reboots, at which time the new configuration for the primary domain  
will also take effect.
```

以 `ldm set-core` 和 `ldm set-memory` 命令所使用的顺序，再次设置与已经分配的内核数量和内存容量相同的数值。然后，重新启动控制域的 Oracle Solaris。

下面的示例使用 `ldm set-core` 命令再次设置内核数量为 32，并使用 `ldm set-memory` 命令再次设置内存容量为 56 GB，以便重新启动 Oracle Solaris。使用 `shutdown` 命令来完成。

```
# ldm set-core 32 primary
```

```
-----  
Notice: The primary domain is in the process of a delayed reconfiguration.  
Any changes made to the primary domain will only take effect after it reboots.  
-----
```

```
# ldm set-memory 56G primary
```

```
-----  
Notice: The primary domain is in the process of a delayed reconfiguration.  
Any changes made to the primary domain will only take effect after it reboots.  
-----
```

```
# shutdown -i6 -g0 -y
```

通过执行 `ldm bind-domain` 和 `ldm start-domain` 命令绑定和启动各逻辑域。以停止所使用的顺序相反的顺序执行绑定和启动。

下面的示例显示通过执行 `ldm bind-domain` 和 `ldm start-domain` 命令启动了 `root-dom0`、`root-dom1`、`guest0` 和 `guest1`（以该顺序）。

```
# ldm bind-domain root-dom0  
# ldm bind-domain root-dom1  
# ldm bind-domain guest0  
# ldm bind-domain guest1  
# ldm start-domain root-dom0  
LDom root-dom0 started  
# ldm start-domain root-dom1  
LDom root-dom1 started  
# ldm start-domain guest0  
LDom guest0 started  
# ldm start-domain guest1  
LDom guest1 started
```

执行 `ldm list-domain` 命令检查是否所有逻辑域都处于 active 状态。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.4%	5h 55m

guest0	active	-n----	5100	64	64G	1.7%	12s
guest1	active	-n----	5101	64	64G	10%	9s
root-dom0	active	-n--v-	5000	32	32G	3.6%	15s
root-dom1	active	-n--v-	5001	32	32G	2.2%	11s

执行 `ldm list-devices` 命令检查内存分配状态。

```
# ldm list-devices -a memory
MEMORY
  PA                SIZE                BOUND
  0x700000000000    32G                root-dom0
  0x720000000000    32G                root-dom1
  0x740000000000    32G                guest0
  0x760000800000    1272M              _sys_
  0x760050000000    31488M            guest1
  0x780000000000    32G                guest0
  0x7a0000000000    32G                guest1
  0x7c0000000000    28G                primary
  0x7c0700000000    4G
  0x7e0000800000    1272M              _sys_
  0x7e0050000000    512M              _sys_
  0x7e0070000000    256M              _sys_
  0x7e0080000000    28G                primary
  0x7e0780000000    1280M            guest1
  0x7e07d0000000    768M
```

12. 将配置的逻辑域配置信息保存在 **XSCF** 中。
 执行 `ldm set-spconfig` 命令保存配置的信息。
 下面的示例检查保存的配置信息，然后使用与现有的配置信息相同的名称将其保存。
 执行 `ldm list-spconfig` 命令检查当前配置信息。

```
# ldm list-spconfig
factory-default
ldm-set1 [next poweron]
```

执行 `ldm remove-spconfig` 命令删除需要覆盖的配置信息。

```
# ldm remove-spconfig ldm-set1
```

执行 `ldm add-spconfig` 命令重新保存配置信息。

```
# ldm add-spconfig ldm-set1
```

执行 `ldm list-spconfig` 命令检查保存的配置信息是否成为 `[current]`。

```
# ldm list-spconfig
factory-default
ldm-set1 [current]
```


13. 将配置的逻辑域配置信息保存到 **XML** 文件。
为了防止保存到 XSCF 的配置信息无法使用，请将配置信息保存到 XML 文件。建议将 XML 文件保存到不同的介质。
下面介绍示例的步骤。
执行 `ldm list-domain` 命令检查是否所有逻辑域都处于 `active` 状态。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.0%	6h 9m
guest0	active	-n----	5100	64	64G	0.0%	15m
guest1	active	-n----	5101	64	64G	0.0%	15m
root-dom0	active	-n--v-	5000	32	32G	0.0%	15m
root-dom1	active	-n--v-	5001	32	32G	0.0%	15m

执行 `ldm list-constraints` 命令将配置信息保存到 XML 文件。

```
# ldm list-constraints -x > /ldm-set1.xml
```

A.2.3 活动状态更换的步骤示例（适用于 Oracle VM Server for SPARC 3.1.x）

本节介绍使用 [图 A-2](#)所述的 2BB 配置系统的 PPAR DR 活动状态更换 BB#01 SPARC M10-4S 的步骤示例。

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。
在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

1. **登录主 XSCF。**
执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。
如果您已经登录到备用 XSCF，注销然后再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

2. 执行 `showhardconf` 命令检查需要更换的 **SPARC M10-4S** 中的 **XSCF** 的 **[Status]** 是否是 **"Normal"**。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Locked;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
```

```

BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;
+ Freq:3.000 GHz; Type:0x10;
+ Core:16; Strand:2;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
:

```

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。
 在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

3. 执行 **showbbstatus** 命令来确认需要更换的 **SPARC M10-4S** 中的 **XSCF** 不是主 **XSCF**。

```

XSCF> showbbstatus
BB#00 (Master)

```

如果需要更换的 SPARC M10-4S 是主 XSCF，则执行 **switchscf** 命令切换 XSCF。

```

XSCF> switchscf -t Standby
The XSCF unit switch between the Active and Standby states.
Continue? [y|n] :y

```

注 - 释放系统板前，请确认 XSCF 已经切换并重新启动。

4. 执行 **console** 命令连接到控制域的控制台，然后登录。

```

XSCF> console -p 0

```

5. 检查逻辑域的运行状态和资源使用状况。
 - a. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果

[STATE] 显示 "active"， [FLAGS] 中的字符串左起第二个字符具有以下含义：

- "n": Oracle Solaris 正在运行
- "t": OpenBoot PROM 状态
- "-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

下面的示例表明控制域、两个根域和两个来宾域正在运行。

检查所有的域是处于表明 Oracle Solaris 处于运行状态的 "active"，还是 "inactive" 状态。如果存在处于 OpenBoot PROM 状态或绑定状态的域，则物理分区的动态重新配置可能失败。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.0%	1h 33m
guest0	active	-n----	5100	64	64G	3.1%	2s
guest1	active	-n----	5101	64	64G	1.6%	18m
root-dom0	active	-n--v-	5000	32	32G	3.1%	17m
root-dom1	active	-n--v-	5001	32	32G	3.1%	17m

b. 执行指定 -a 选项的 ldm list-devices 命令来检查资源的使用状况。

在下面的示例中，指定 -a 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。

# ldm list-devices -a		
CORE		
ID	%FREE	CPUSET
0	0	(0, 1)
4	0	(8, 9)
8	0	(16, 17)
(省略)		
944	0	(1888, 1889)
948	0	(1896, 1897)
952	0	(1904, 1905)
956	0	(1912, 1913)
VCPU		
PID	%FREE	PM
0	0	no
1	0	no
8	0	no
9	0	no
(省略)		
1904	0	no
1905	0	no
1912	0	no
1913	0	no
(省略)		

6. 释放控制域中系统卷和 I/O 设备的冗余配置。
- 若需释放 BB#01，请释放需要更换的 SPARC M10-4S 的 I/O 设备和控制域中使用的 I/O 设备。有关取消冗余配置的步骤的详情，请参见该冗余配置软件的文档。
- a. 取消控制域中系统卷的冗余配置。

下面的示例介绍如何取消控制域中系统卷的 ZFS 镜像功能。
在控制域中执行 `zpool status` 命令检查镜像配置状态。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                               ONLINE    0     0     0
    mirror-0                          ONLINE    0     0     0
      c2t50000393E802CCE2d0s0        ONLINE    0     0     0
      c3t50000393A803B13Ed0s0        ONLINE    0     0     0
errors: No known data errors
```

执行 `zpool detach` 命令从镜像配置中释放磁盘。

```
# zpool detach rpool c3t50000393A803B13Ed0
```

执行 `zpool status` 命令确认镜像配置已经取消。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                               ONLINE    0     0     0
    mirror-0                          ONLINE    0     0     0
      c2t50000393E802CCE2d0s0        ONLINE    0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则移除冗余配置或停止使用那些设备。
有关如何取消冗余配置或停止使用设备的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

b. 删除控制域的 I/O 配置。

通过延迟重新配置，从分配给控制域的物理 I/O 设备中，删除 BB#01 中的根合成体。

首先，将控制域设置到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

执行 `ldm list-io` 命令检查分配给 `primary` 的根合成体。

下面的示例表明 BB1 设备的根合成体是 PCIE8 和 PCIE12。

```
# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary    IOV
PCIE4          BUS      PCIE4      primary    IOV
PCIE8          BUS      PCIE8      primary    IOV
PCIE12         BUS      PCIE12     primary    IOV
/BB0/CMUL/NET0 PCIE      PCIE0      primary    OCC
/BB0/CMUL/SASHBA PCIE      PCIE0      primary    OCC
/BB0/CMUL/NET2 PCIE      PCIE4      primary    OCC
/BB1/CMUL/NET0 PCIE      PCIE8      primary    OCC
/BB1/CMUL/SASHBA PCIE      PCIE8      primary    OCC
/BB1/CMUL/NET2 PCIE      PCIE12     primary    OCC
```

执行 `ldm remove-io` 命令从 `primary` 删除 `PCIE8` 和 `PCIE12`，然后重新启动 Oracle Solaris。

```
# ldm remove-io PCIE8 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

# ldm remove-io PCIE12 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

# shutdown -i6 -g0 -y
....
```

Oracle Solaris 启动后，执行 `ldm list-io` 命令确认 `BB#01` 中的根合成体已经从控制域删除。

```
# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary    IOV
PCIE4          BUS      PCIE4      primary    IOV
/BB0/CMUL/NET0 PCIE      PCIE0      primary    OCC
/BB0/CMUL/SASHBA PCIE      PCIE0      primary    OCC
/BB0/CMUL/NET2 PCIE      PCIE4      primary    OCC
```

c. 取消分配给来宾域的虚拟 I/O 设备的冗余配置。

若需首先关闭分配 `BB#01` 中的根合成体的根域 (`root-dom1`)，请登录各来宾域，然后从 `root-dom1` 中取消虚拟 I/O 设备的冗余配置。

有关如何使用冗余配置软件的详情，请参见该冗余配置软件的文档。

在下面的示例中，将虚拟网络设备 (`vnet1`) 从 `IPMP` 配置中取消。有关命令的详情，请参见 Oracle Solaris 的手册。

登录来宾域 (`guest0`)。

```
# ldm list-domain
NAME          STATE      FLAGS      CONS      VCPU      MEMORY      UTIL      UPTIME
primary       active    -n-cv-    UART       64        56G         0.0%     4h 17m
```

```
guest0          active      -n----  5100    64    64G      0.0%  1h 13m
guest1          active      -n----  5101    64    64G      0.0%  1h  4m
root-dom0       active      -n--v-  5000    32    32G      0.0%  1h 47m
root-dom1       active      -n--v-  5001    32    32G      0.0%  1h 19m
# telnet localhost 5100
....
guest0#
```

执行 `dladm show-phys` 命令检查虚拟网络接口 (`vnet1`) 和网络接口的名称 (`net1`) 之间的映射。

```
guest0# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      up        0      unknown  vnet0
net1          Ethernet      up        0      unknown  vnet1
```

执行 `ipmpstat -i` 命令检查配置 IPMP 的网络接口的配置信息。

```
guest0# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net0       yes    ipmp0      -smbM--    up        disabled   ok
net1       no     ipmp0      is-----  up        disabled   ok
```

执行 `if_mpadm -d net1` 命令从 IPMP 组释放 `net1`，然后执行 `ipmpstat -i` 命令确认已经将其释放。下面的示例确认 `STATE` 是 `offline`。

```
guest0# if_mpadm -d net1
guest0# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net0       yes    ipmp0      -smbM--    up        disabled   ok
net1       no     ipmp0      -s---d-    up        disabled   offline
```

执行 `ipadm delete-ip` 命令删除 `net1`。

```
guest0# ipadm delete-ip net1
```

同样，对来宾域 (`guest1`) 执行相同的释放处理。

d. 从根域移除已分配的虚拟 I/O 设备，以便将其停止。

按照下述步骤，执行 `ldm remove-vdisk` 和 `ldm remove-vnet` 命令从根域移除已分配的虚拟磁盘 (`vdisk`) 和虚拟网络设备 (`vnet`)，以便将其停止。

下面的示例显示移除使用 `BB#01` 根域 (`root-dom1`) 的虚拟 I/O 服务的虚拟磁盘 (`vdisk10`) 和虚拟网络设备 (`vnet10`) 的命令的执行。

```
# ldm remove-vdisk vdisk10 guest0
# ldm remove-vnet vnet10 guest0
```

对来宾域 (`guest1`) 执行相同的删除操作。

7. 检查 I/O 设备的资源使用状况，然后取消需要更换的 **SPARC M10-4S** 中的所有 I/O 设备。

a. 检查分配需要释放的 **SPARC M10-4S** 中的根合成体的逻辑域。

执行 `ldm list-io` 命令，以检查分配 BB#01 中的根合成体的逻辑域。

下面的示例表明仅 `root-dom1` 具有以 `"/BB1/"` 开始的 PCIe 端点。您可以看到 PCIe 端点根合成体 (BUS) `PCIE9`、`PCIE10`、`PCIE11`、`PCIE13`、`PCIE14` 和 `PCIE15` 被分配给 `root-dom1`。

```
# ldm list-io
NAME                                     TYPE    BUS      DOMAIN    STATUS
----
PCIE0                                   BUS     PCIE0    primary   IOV
PCIE1                                   BUS     PCIE1    root-dom0 IOV
PCIE2                                   BUS     PCIE2    root-dom0 IOV
PCIE3                                   BUS     PCIE3    root-dom0 IOV
PCIE4                                   BUS     PCIE4    primary   IOV
PCIE5                                   BUS     PCIE5    root-dom0 IOV
PCIE6                                   BUS     PCIE6    root-dom0 IOV
PCIE7                                   BUS     PCIE7    root-dom0 IOV
PCIE8                                   BUS     PCIE8
PCIE9                                   BUS     PCIE9    root-dom1 IOV
PCIE10                                  BUS     PCIE10   root-dom1 IOV
PCIE11                                  BUS     PCIE11   root-dom1 IOV
PCIE12                                  BUS     PCIE12
PCIE13                                  BUS     PCIE13   root-dom1 IOV
PCIE14                                  BUS     PCIE14   root-dom1 IOV
PCIE15                                  BUS     PCIE15   root-dom1 IOV
....
/BB1/CMUL/NET0                         PCIE    PCIE8
/BB1/CMUL/SASHBA                       PCIE    PCIE8
/BB1/PCI0                              PCIE    PCIE9    root-dom1OCC
/BB1/PCI3                              PCIE    PCIE10   root-dom1OCC
/BB1/PCI4                              PCIE    PCIE10   root-dom1OCC
/BB1/PCI7                              PCIE    PCIE11   root-dom1OCC
/BB1/PCI8                              PCIE    PCIE11   root-dom1OCC
/BB1/CMUL/NET2                         PCIE    PCIE12
/BB1/PCI1                              PCIE    PCIE13   root-dom1OCC
/BB1/PCI2                              PCIE    PCIE13   root-dom1OCC
/BB1/PCI5                              PCIE    PCIE14   root-dom1OCC
/BB1/PCI6                              PCIE    PCIE14   root-dom1OCC
/BB1/PCI9                              PCIE    PCIE15   root-dom1OCC
/BB1/PCI10                             PCIE    PCIE15   root-dom1OCC
```

b. 停止分配需要释放的 **SPARC M10-4S** 中的根合成体的逻辑域。

下面的示例执行 `ldm stop-domain` 和 `ldm unbind-domain` 命令释放根域 (`root-dom1`)，并表明根域处于 `inactive` 状态。

```
# ldm stop-domain root-dom1
LDom root-dom1 stopped
# ldm unbind-domain root-dom1
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.2%	4h 59m
guest0	active	-n----	5100	64	64G	0.0%	1h 55m
guest1	active	-n----	5101	64	64G	0.0%	1h 46m
root-dom0	active	-n--v-	5000	32	32G	0.0%	2h 29m
root-dom1	inactive	-----		32	32G		

- c. 确认需要更换的 **SPARC M10-4S** 中的所有 I/O 设备已经被取消。
执行 `ldm list-io` 命令确认已经释放所有 I/O 设备。

# <code>ldm list-io</code>				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10		
PCIE11	BUS	PCIE11		
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13		
PCIE14	BUS	PCIE14		
PCIE15	BUS	PCIE15		
(省略)				

8. 手动减少分配给逻辑域的 **CPU** 内核数量和内存资源。
下面解释手动减少分配给逻辑域的 CPU 内核数量和内存资源的步骤，而不通过指定 `-m unbind=resource` 选项来满足 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中的 ["CPU 内核和内存的配置"](#) 的条件。
例如，如果由于错误，`deleteboard` 命令失败，在某些情况下按照以下步骤可以执行 `deleteboard` 命令。
当指定 `deleteboard` 命令的 `-m unbind=resource` 选项时，转到步骤 9。

注 - 在早于 Oracle VM Server for SPARC 3.2 的 Oracle VM Server for SPARC 版本中，不支持 `deleteboard -c disconnect` 命令的 `-m unbind=resource` 选项。执行此步骤，然后使用不指定 `-m unbind=resource` 的 `deleteboard` 命令释放 SPARC M10-4S。

- a. 检查 **CPU** 内核数量，然后将其删除。
由于释放 SPARC M10-4S，可使用的 CPU 内核数量将减少，因此首先使用以下步骤来提前减少分配给逻辑域的 CPU 内核数量。
i. 释放 **SPARC M10-4S** 后，检查 **CPU** 内核数量。
从 XSCF 执行 `showpparinfo` 命令来检查 CPU 内核数量，不包括需要释放的 SPARC M10-4S 中的 CPU 内核数量。

在下面的示例中，如果将要释放物理系统板 (PSB) 编号为 01-0 的 SPARC M10-4S，则必须计算和另一个 PSB 编号为 00-0 的 CPU 内核的总和。因此， $16 + 16 + 16 + 16 = 64$ 个内核。

```
XSCF> showpparinfo -p 0
PPAR#00 Information:
-----
CPU(s)                :      8
CPU Cores              :     128
CPU Threads           :     256
Memory size (GB)      :     256
CoD Assigned (Cores)  :     256
CPU(s):
-----
PID  PSB   CPU#  Cores  Threads
00   00-0  0       16     32
00   00-0  1       16     32
00   00-0  2       16     32
00   00-0  3       16     32
00   01-0  0       16     32
00   01-0  1       16     32
00   01-0  2       16     32
00   01-0  3       16     32
(省略)
```

ii.检查被分配给各逻辑域的 CPU 内核的总数。

执行 `ldm list-devices -a core` 命令。在 %FREE 栏中除 100 之外的数值所在行的数量是分配给逻辑域的 CPU 内核的总数。

在下面的示例中，执行 `ldm list-devices -a core` 命令，使用 `-p` 选项执行了检查。因此，可以看到 112 个内核被绑定到整个逻辑域。

```
# ldm list-devices -a core
CORE
  ID      %FREE  CPUSSET
  0        0    (0, 1)
  4        0    (8, 9)
  8        0   (16, 17)
 12        0   (24, 25)
(省略)
# ldm list-devices -a -p core | egrep -v "CORE|VERSION|free=100" | wc -l
112
```

iii.从 SPARC M10-4S 释放的结果计算内核不足的数量。

使用下面的公式，计算释放 SPARC M10-4S 后将要产生的 CPU 内核不足的数量。

CPU 内核不足的数量 = 逻辑域中使用的内核数量（步骤 ii）- 释放后的物理内核的数量（步骤 i）

在步骤 i 和步骤 ii 示例中, 可以得出不足的数量将是 112 个内核（使用中）- 64 个内核（剩余）= 48 个内核。

iv. 如果发生了 CPU 内核不足，请考虑要删除哪个逻辑域。

如果从步骤 iii 的结果可以得出将要发生 CPU 内核不足，有必要从逻辑域删除 CPU 内核。

执行 `ldm list-domain` 命令检查分配给各逻辑域处于 active 或 bound 状态的 CPU 内核数量，以及检查 CPU 内核需要从中删除的逻辑域。

在下面的示例中，可以得出将 32 个内核 (64vcpu) 分配给 primary，将 32 个内核 (64vcpu) 分配给 guest0，将 32 个内核 (64vcpu) 分配给 guest1，将 16 个内核 (32vcpu) 分配给 root-dom0 以及将 16 个内核 (32vcpu) 分配给 root-dom1。在此示例中，需要删除 48 个内核。这样，将从每个 primary、guest0 和 guest1 中删除 16 个内核。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.1%	18h 17m
guest0	active	-n----	5100	64	64G	0.0%	15h 13m
guest1	active	-n----	5101	64	64G	0.0%	15h 4m
root-dom0	active	-n--v-	5000	32	32G	0.0%	15h 47m
root-dom1	inactive	-----		32	32G		

v. 执行 `ldm remove-core` 命令从目标逻辑域删除 CPU 内核。

在下面的示例中，从每个 primary、guest0 和 guest1 中删除 16 个内核，然后进行检查确认是否已经实际将其删除。

```
# ldm remove-core 16 primary
# ldm remove-core 16 guest0
# ldm remove-core 16 guest1
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	32	56G	0.0%	18h 19m
guest0	active	-n----	5100	32	64G	0.0%	15h 15m
guest1	active	-n----	5101	32	64G	0.0%	15h 5m
root-dom0	active	-n--v-	5000	32	32G	0.0%	15h 49m
root-dom1	inactive	-----	5001	32	32G		

```
# ldm list-devices -a -p core | egrep -v "CORE|VERSION|free=100" | wc -l
64
```

b. 检查并删除内存资源。

由于释放 SPARC M10-4S，可使用的内存区将减少，因此使用以下步骤删除分配给逻辑域的内存资源。

i. 检查内存的连续区域（内存块）的使用状况。

执行 `prtdiag` 命令和 `ldm list-devices -a memory` 命令检查分配给各逻辑域的内存块，并检查分配未分配的内存块的 SPARC M10-4S。

首先，执行 `prtdiag` 命令检查内存中物理地址和 SPARC M10-4S 之间的对应关系。

# prtdiag (省略)				
===== Physical Memory Configuration =====				
Segment Table:				

Base Address	Segment Size	Interleave Factor	Bank Size	Contains Modules

0x7e0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUL/CMP0/MEM00A
0x7c0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUL/CMP1/MEM10A
0x7a0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUU/CMP0/MEM00A
0x780000000000 (省略)	32 GB	4	8 GB	/BB0/CMUU/CMP1/MEM10A
0x760000000000 (省略)	32 GB	4	8 GB	/BB1/CMUL/CMP0/MEM00A
0x740000000000 (省略)	32 GB	4	8 GB	/BB1/CMUL/CMP1/MEM10A
0x720000000000 (省略)	32 GB	4	8 GB	/BB1/CMUU/CMP0/MEM00A
0x700000000000 (省略)	32 GB	4	8 GB	/BB1/CMUU/CMP1/MEM10A

本示例的结果按内存中物理地址的升序重新排列。下表列出了物理地址和 SPARC M10-4S 之间的对应关系。

表 A-2 物理地址和 SPARC M10-4S 之间对应关系的示例

基地址（物理地址）	SPARC M10-4S 的模块化架构配置
0x700000000000 和之后	BB1
0x720000000000 和之后	BB1
0x740000000000 和之后	BB1
0x760000000000 和之后	BB1
0x780000000000 和之后	BB0
0x7a0000000000 和之后	BB0
0x7c0000000000 和之后	BB0
0x7e0000000000 和之后	BB0

然后，执行 `ldm list-devices -a memory` 命令检查分配给逻辑域的内存连续区域（本文档下文中称为内存块）和未分配的内存块。

在下面的示例中，执行了 `ldm list-devices -a memory` 命令。
各参数的含义如下。

PA: 内存块的开始物理地址

SIZE: 内存块的容量

BOUND: 分配内存块的逻辑域的名称。空白是未分配的区域，_sys_ 是未被分配给逻辑域的控制区。

# ldm list-devices -a memory		
MEMORY		
PA	SIZE	BOUND
0x700000000000	32G	root-dom0
0x720000000000	32G	
0x740000000000	32G	guest0
0x760000800000	1272M	_sys_
0x760050000000	31488M	guest1
0x780000000000	32G	guest0
0x7a0000000000	32G	guest1
0x7c0000000000	28G	primary
0x7c0700000000	4G	
0x7e0000800000	1272M	_sys_
0x7e0050000000	512M	_sys_
0x7e0070000000	256M	_sys_
0x7e0080000000	28G	primary
0x7e0780000000	1280M	guest1
0x7e07d0000000	768M	

通过将上述显示的结果与使用 prtdiag 命令检查的物理位置相结合，可以得到内存块的使用状况如下。

表 A-3 内存块使用状况的示例

SPARC M10-4S	物理地址	容量	逻辑域
BB1（更换的目标）	0x700000000000	32 GB	root-dom0
	0x720000000000	32 GB	未分配
	0x740000000000	32 GB	guest0
	0x760050000000	31,488 MB	guest1
BB0	0x780000000000	32 GB	guest0
	0x7a0000000000	32 GB	guest1
	0x7c0000000000	28 GB	primary
	0x7c0700000000	4 GB	未分配
	0x7e0080000000	28 GB	primary
	0x7e0780000000	1,280 MB	guest1
	0x7e07d0000000	768 MB	未分配

ii.检查移动源内存块的容量和数量。

参照内存块使用状况的检查结果时，检查分配给需要更换的 SPARC M10-4S 的内存块（在下文中称为“源内存块”）。

对于“表 A-3 内存块使用状况的示例”而言，32 GB x 2 和 31,488 MB x 1 的内存块是正在 BB1 中被用作分配给逻辑域的内存块。

iii.确定从其中删除内存的逻辑域，以及数量。

然后，检查分配给各逻辑域的内存块的位置，并通过删除内存块单元中的内存以及通过减少内存块的容量来确保源内存块可被移动至将不被释放的 SPARC M10-4S 的未分配的内存块（本文档下文中称为 "目标内存块"）。因此，可作出最终决定，将删除多少内存和从中将其删除的逻辑域。

支持以下方法。

- 通过完全删除源内存块，减少需要被移动的内存块的数量。
- 完全删除不作为释放对象的 SPARC M10-4S 上被分配到逻辑域的内存块，通过增加可用空间来增加移动目标内存空间。
- 减少源内存块的容量，以便其适用于目标上的可用区域。
- 通过减少源内存块的容量和通过减少使用中的目标的内存块的容量，增加目标上的可用内存块的数量以便可进行移动。

注 - 减少容量后，可用内存块将不连续（将成为碎片）。即使将多个小内存块删除以增加可用区域的数量，可用区域也不连续。如果源内存块的连续区域较大，将无法移动。在这种情况下，删除源内存块来调整其容量。

注 - 当考虑此种情况时，可以通过选择可删除的与现有的内存块尽可能具有相同容量的内存块，可减少删除之后成为碎片的连续区域的可能性。这将增加内存块成功移动的可能性。

注 - 如果过多的内存被删除，这将增大逻辑域中的内存的负担，可能造成 Oracle Solaris 挂起等问题。通过使用 vmstat 命令和检查 "free" 容量作为粗略参考，有必要注意勿删除过多内存。

基于 "表 A-3 内存块使用状况的示例" 研究删除计划。

根据此种考虑，制定从 root-dom0 中删除 4 GB，从 guest0 中删除 32 GB，从 guest1 中删除 31,488 MB 以及从 primary 中删除 28 GB 的计划，以 "表 A-4 内存块删除计划" 为例，如下所示。

表 A-4 内存块删除计划

SPARC M10-4S	容量	逻辑域	删除计划
BB1（更换的目标）	32 GB	root-dom0	此区域减少 4 GB 至 28 GB，从 BB0 上的 primary 删除 28 GB，然后实现移动。
	32 GB	未分配	-
	32 GB	guest0	由于在 BB0 上存在 32-GB guest0 内存，将其删除。
	31,488 MB	guest1	由于在 BB0 上存在 32-GB guest1 内存，将其删除。
BB0	32 GB	guest0	保留。
	32 GB	guest1	保留。
	28 GB	primary	保留。
	4 GB	未分配	-
	28 GB	primary	将其删除，以便移动 root-dom0 的 28 GB。

表 A-4 内存块删除计划 (续)

SPARC M10-4S	容量	逻辑域	删除计划
	1,280 MB	guest1	保留。
	768 MB	未分配	-

iv. 从逻辑域手动删除内存。

按照在步骤 iii 中制定的内存删除计划，使用 `ldm remove-memory` 命令从逻辑域删除内存。

下面的示例显示了根据“表 A-4 内存块删除计划”删除内存的命令的执行。

```
# ldm remove-memory 4G root-dom0
# ldm remove-memory 32G guest0
# ldm remove-memory 31488M guest1
# ldm remove-memory 28G primary
```

v. 检查已删除的内存块的状态。

执行 `ldm list-devices -a memory` 命令，通过参考删除的结果来检查该布置是否可实现移动。如果存在任何无法移动的内存块，根据结果考虑哪些额外的内存块需要删除，然后将其删除。

在下面的示例中，通过顺次比较分配给 BB1 的大容量和 BB0 上可用区域内的大容量可很容易检查是否可移动。

```
# ldm list-devices -a memory
MEMORY
  PA                SIZE                BOUND
(BB1)
  0x700000000000    256M                root-dom0 →可通过分割 4-GB 目标移动
  0x700010000000    4G                  root-dom0 →可移动到 0x780000000000
  0x700110000000    28416M              root-dom0 (28 GB)
  0x720000000000    32G                  guest0 →可通过分割 4-GB 目标移动
  0x740000000000    256M                  guest0 →由于无目标需要再次删除 (*)
  0x740010000000    4G                    guest0
  0x740110000000    28416M              guest0
  0x760000800000    1272M                _sys_
  0x760050000000    256M                guest1 →可通过分割 4-GB 区域移动
  0x760060000000    1792M
  0x7600d0000000    29440M              guest1 →可移动到 0x7a0000000000
  (BB1)              (29-GB)
  0x780000000000    28G                  ←可从 root-dom0
  0x780700000000    4G                    (0x700110000000) 移动
  0x7a0000000000    29G                  ←可从 guest1
  0x7a0740000000    3G                    (0x7600d0000000) 移动
  0x7c0000000000    256M                primary
```

0x7c0010000000	4G		←可从 root-dom0、 guest0 和 guest1 移动 256 MB
0x7c0110000000	24320M	primary	
0x7c0700000000	4G		
0x7e0000800000	1272M	_sys_	
0x7e0050000000	512M	_sys_	
0x7e0070000000	256M	_sys_	
0x7e0080000000	24G		←guest0 (0x740110000000) 移动没有足够可用空间 (*)
0x7e0680000000	4G	primary	
0x7e0780000000	1280M	guest1	
0x7e07d0000000	768M		

在上述示例中, 标记有 (*) 的目标仅有 24 GB (24,576 MB) 的可用空间, 因此从源 (BB1) 上的 28416-MB 区域 (guest0) 删除 3,840 MB, 然后重复检查。在下面的示例中, 可以得出所有内存块都可移动。

# ldm remove-memory 3840M guest0			
# ldm list-devices -a memory			
MEMORY			
PA	SIZE	BOUND	
(BB1)			
0x700000000000	256M	root-dom0	
0x700010000000	4G		
0x700110000000	28416M	root-dom0	
0x720000000000	32G		
0x740000000000	256M	guest0	
0x740010000000	7936M		
0x740200000000	24G	guest0	→可移动到 0x7e0080000000 (24 G)
0x760000800000	1272M	_sys_	
0x760050000000	256M	guest1	
0x760060000000	1792M		
0x7600d0000000	29440M	guest1	
(BB1)			
0x780000000000	28G		
0x780700000000	4G	guest0	
0x7a0000000000	29G		
0x7a0740000000	3G	guest1	
0x7c0000000000	256M	primary	
0x7c0010000000	4G		
0x7c0110000000	24320M	primary	
0x7c0700000000	4G		
0x7e0000800000	1272M	_sys_	
0x7e0050000000	512M	_sys_	
0x7e0070000000	256M	_sys_	
0x7e0080000000	24G		←可从 guest0 (0x740200000000)
移动			
0x7e0680000000	4G	primary	
0x7e0780000000	1280M	guest1	
0x7e07d0000000	768M		

9. 从物理分区释放对应于 **SPARC M10-4S** 的系统板 (PSB<BB>)。
 - a. 执行 **deleteboard -c disconnect** 命令将 PSB 从物理分区释放。

```
XSCF> deleteboard -c disconnect 01-0
PSB#01-0 will be unconfigured from PPAR immediately. Continue?
[y|n] :y
Start unconfigure preparation of PSB. [1200sec]
0end
Unconfigure preparation of PSB has completed.
Start unconfiguring PSB from PPAR. [7200sec]
0..... 30..... 60....end
Unconfigured PSB from PPAR.
PSB power off sequence started. [1200sec]
0..... 30..... 60..... 90.....120.....150.....end
Operation has completed.
```

- b. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。
结束值 0 表明 **deleteboard** 命令正常终止。
如果结束值非 0，或如果在执行 **deleteboard** 命令时显示错误信息，则表明 **deleteboard** 命令异常终止。根据错误信息，通过参考 "C.1.2 **deleteboard**" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

- c. 执行 **showboards** 命令检查 **PSB** 状态。
确认需要更换的 SPARC M10-4S 内的 PSB 是否处于 "Assigned" 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 "n"。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault

00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

10. 执行 **replacefru** 命令来更换 **SPARC M10-4S**。

```
XSCF> replacefru
```

注 -关于使用 **replacefru** 命令更换 SPARC M10-4S 的详情，请参见《Fujitsu M10-4/Fujitsu M10-4S/SPARC M10-4/SPARC M10-4S Service Manual》中的 "5.8 Releasing a SPARC M10-4/M10-4S FRU from the System with the **replacefru** Command" 和 "6.2 Incorporating a SPARC M10-4/M10-4S FRU into the System with the **replacefru** Command"。

- 11. 将已更换的 **SPARC M10-4** 的系统板 (**PSB<BB>**) 整合到物理分区。
 - a. 执行 **showboards** 命令检查 **PSB** 状态。
确认需要更换的 SPARC M10-4S 内的 PSB 是否处于 Assigned 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 "n"。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

- b. 执行 **addboard -c configure** 命令将 **PSB** 整合到物理分区。
- 若需恢复初始逻辑域配置，请执行指定 **-m bind=resource** 选项的 **addboard -c configure** 命令。

```
XSCF> addboard -c configure -m bind=resource -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
810.....840.....870.....900.....930.....960.....end
Connected PSB to PPAR.
Start configuring PSB to Logical Domains (LDoms) Manager.
[1800sec] 0.....end
Configured PSB to Logical Domains (LDoms) Manager.
Operation has completed.
```

注 - 如果执行 **addboard** 命令过程中出现错误信息，请参见 "[C.1.1 addboard](#)"，然后识别错误并采取应对措施。

- c. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。
- 结束值 0 表明 **addboard** 命令正常终止。
- 如果结束值非 0，或如果在执行 **addboard** 命令时显示错误信息，则表明 **addboard** 命令异常终止。根据错误信息，通过参考 "[C.1.1 addboard](#)" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

- d. 执行 **showboards** 命令检查 **PSB** 状态。
- 在已更换的 SPARC M10-4S 中的 PSB 已经被成功整合后，确认 **[Conn]** 和 **[Conf]** 栏是否都显示 "y" 。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

12. 检查逻辑域的运行状态。
- a. 执行 **console** 命令连接到控制域的控制台，然后登录。

- b. 执行 **ldm list-domain** 命令来确认 **PSB** 添加后逻辑域的运行状态未发生变化。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义：

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

```
# ldm list-domain
NAME          STATE      FLAGS    CONS   VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   UART   32     28G     64%   2h 54m
guest0        active    -n----   5100   32     28928M  42%   2h 54m
guest1        active    -n----   5101   32     34048M  11%   2h 54m
root-dom0     active    -n--v-   5000   32     28G     10%   2h 54m
root-dom1     inactive  ------  32     32G
```

13. 恢复已删除的资源。

如果使用 **deleteboard** 命令已经删除分配给逻辑域的资源，则使用 **ldm add-core** 和 **ldm add-memory** 命令添加资源，以便将其恢复。

在下面的示例中，删除的 CPU 内核和内存被添加，以便在更换 SPARC M10-4S 之前恢复资源。

```
# ldm add-core 16 primary
# ldm add-core 16 guest0
# ldm add-core 16 guest1
# ldm add-memory 28G primary
# ldm add-memory 4G root-dom0
# ldm add-memory 36608M guest0
# ldm add-memory 31488M guest1
# ldm list-domain
NAME          STATE      FLAGS    CONS   VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   UART   64     56G     0.2%  4h 59m
guest0        active    -n----   5100   64     64G     0.0%  1h 55m
guest1        active    -n----   5101   64     64G     0.0%  1h 46m
root-dom0     active    -n--v-   5000   32     32G     0.0%  2h 29m
root-dom1     inactive  ------  32     32G
```

14. 重新启动 I/O 设备的使用。

- a. 重新分配根合成体。

执行 **ldm bind-domain** 和 **ldm start-domain** 命令，以便启动在 **unbind** 状态下分配已更换的 SPARC M10-4S 中的根合成体的根域。

下面的示例启动在 **unbind** 状态下的根域 (root-dom1)，然后确认是否已经启动。

```
# ldm bind-domain root-dom1
# ldm start-domain root-dom1
LDom root-dom1 started
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.2%	3h 8m
guest0	active	-n----	5100	64	64G	0.0%	3h 8m
guest1	active	-n----	5101	64	64G	0.0%	3h 8m
root-dom0	active	-n--v-	5000	32	32G	0.0%	3h 8m
root-dom1	active	-n--v-	5001	32	32G	7.3%	8s

执行 `ldm list-io` 命令确认物理 I/O 设备是否已经分配给刚刚启动的根域。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV
....				
/BB1/CMUL/NET0	PCIE	PCIE8		UNK
/BB1/CMUL/SASHBA	PCIE	PCIE8		UNK
/BB1/PCI0	PCIE	PCIE9	root-dom1	IOCC
/BB1/PCI3	PCIE	PCIE10	root-dom1	IOCC
/BB1/PCI4	PCIE	PCIE10	root-dom1	IOCC
/BB1/PCI7	PCIE	PCIE11	root-dom1	IOCC
/BB1/PCI8	PCIE	PCIE11	root-dom1	IOCC
/BB1/CMUL/NET2	PCIE	PCIE12		UNK
/BB1/PCI1	PCIE	PCIE13	root-dom1	IOCC
/BB1/PCI2	PCIE	PCIE13	root-dom1	IOCC
/BB1/PCI5	PCIE	PCIE14	root-dom1	IOCC
/BB1/PCI6	PCIE	PCIE14	root-dom1	IOCC
/BB1/PCI9	PCIE	PCIE15	root-dom1	IOCC
/BB1/PCI10	PCIE	PCIE15	root-dom1	IOCC

b. 将虚拟 I/O 设备从根域添加到来宾域。

执行 `ldm add-vdisk` 和 `ldm add-vnet` 命令，将支持已经启动的根域的虚拟 I/O 服务的虚拟磁盘 (vdisk) 和虚拟网络设备 (vnet) 添加到各个来宾域。

下面的示例显示添加使用 BB#01 根域 (root-dom1) 的虚拟 I/O 服务的虚拟磁盘

(vdisk10) 和虚拟网络设备 (vnet10) 的命令的执行。
对来宾域 (guest1) 执行相同的添加操作。

```
# ldm add-vdisk id=1 vdisk10 vol10@vds1 guest0
# ldm add-vnet id=1 vnet10 vsw10 guest0
```

注 - 若需再次添加虚拟 I/O 设备，有必要提前指定已分配的 ID。您可以在虚拟 I/O 设备被删除之前，在被使用的状态下，从 ldm list -l 命令的执行结果检查 ID。

c. 将分配给来宾域的虚拟 I/O 设备整合到冗余配置中。

启动分配 BB#1 中的根合成体的根域 (root-dom1) 后，对应各来宾域的虚拟 I/O 设备服务也被启动。
登录各来宾域，然后从以前被取消的 root-dom1 中将虚拟 I/O 设备整合到冗余配置。有关如何使用冗余配置软件的详情，请参见该冗余配置软件的文档。
下面的示例介绍将虚拟网络设备 (vnet1) 整合到 IPMP 配置的示例。有关命令的详情，请参见 Oracle Solaris 的手册。
首先登录来宾域 (guest0)。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    64      56G       0.0%    4h 17m
guest0        active    -n----   5100    64      64G       0.0%    1h 13m
guest1        active    -n----   5101    64      64G       0.0%    1h  4m
root-dom0     active    -n----   5000    32      32G       0.0%    1h 47m
root-dom1     active    -n----   5001    32      32G       0.0%    1h 19m
# telnet localhost 5100
....
guest0#
```

执行 dladm show-phys 命令检查虚拟网络接口 (vnet1) 和网络接口的名称 (net1) 之间的映射。

```
guest0# dladm show-phys
LINK          MEDIA          STATE    SPEED    DUPLEX    DEVICE
net0          Ethernet      up        0        unknown  vnet0
net1          Ethernet      up        0        unknown  vnet1
```

执行 ipadm create-ip 命令、ipadm set-ifprop 命令以及 ipadm add-ipmp 命令将 net1 注册为 ipmp0 的备用设备。

```
guest0# ipadm create-ip net1
guest0# ipadm set-ifprop -p standby=on -m ip net1
guest0# ipadm add-ipmp -i net1 ipmp0
```

执行 ipmpstat -i 命令确认配置 IPMP 的网络接口的 STATE 是否显示 ok。

对另一个来宾域 (guest1)，执行相同的步骤。

```
guest0# ipmpstat -i
INTERFACE ACTIVE GROUP FLAGS LINK PROBE STATE
net0 yes ipmp0 -smbM-- up disabled ok
net1 no ipmp0 -s---d- up disabled ok
```

15. 将控制域上的系统卷和 I/O 设备恢复到冗余配置。

a. 添加控制域的根合成体配置。

通过延迟配置的方式，在以前从控制域中被移除的 BB#01 中添加根合成体。

首先，将控制域设置到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

执行 `ldm list-io` 命令检查未分配的根合成体。

下面的示例表明 BB1 设备的根合成体，PCIE8 和 PCIE12 未被分配。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV

(省略)

执行 `ldm add-io` 命令将 PCIE8 和 PCIE12 添加到 `primary`，然后重新启动 Oracle Solaris。

```
# ldm add-io PCIE8 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm add-io PCIE12 primary
```

```

-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# shutdown -i6 -g0 -y
....

```

Oracle Solaris 启动后，执行 `ldm list-io` 命令确认 BB#01 中的根合成体是否已经被添加到控制域。

```

# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary    IOV
PCIE4          BUS      PCIE4      primary    IOV
PCIE8          BUS      PCIE8      primary    IOV
PCIE12         BUS      PCIE12     primary    IOV
/BB0/CMUL/NET0 PCIE     PCIE0      primary    OCC
/BB0/CMUL/SASHBA PCIE     PCIE0      primary    OCC
/BB0/CMUL/NET2 PCIE     PCIE4      primary    OCC

```

b. 将控制域中的系统卷设置为冗余配置。

在控制域中执行 `zpool status` 命令检查镜像配置状态。

下面的示例介绍如何配置控制域中系统卷的 ZFS 镜像功能。

```

# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 29.1M in 0h0m with 0 errors on Thu Jan 23 17:27:59 2014
config:
    NAME                                STATE      READ  WRITE CKSUM
    rpool                                ONLINE    0     0     0
    c2t50000393E802CCE2d0s0             ONLINE    0     0     0
errors: No known data errors

```

执行 `zpool attach` 命令，以便将磁盘整合到镜像配置中。

```

# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting.
#

```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。

使用 `zpool status` 命令确认同步处理 (resilver) 是否完成。

下面显示同步处理过程中显示的示例。

```

# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function in a degraded state.

```

```
action: Wait for the resilver to complete.
       Run 'zpool status -v' to see device specific details.
scan:  resilver in progress since Mon Jan 27 15:55:47 2014
       21.1G scanned out of 70.6G at 120M/s, 0h7m to go
       21.0G resilvered, 29.84% done
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                               DEGRADED   0     0     0
        mirror-0                          DEGRADED   0     0     0
          c2t50000393E802CCE2d0s0        ONLINE    0     0     0
          c3t50000393A803B13Ed0s0        DEGRADED   0     0     0 (resilvering)
errors: No known data errors
```

同步处理完成后，将显示如下画面：

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                               ONLINE     0     0     0
        mirror-0                          ONLINE     0     0     0
          c2t50000393E802CCE2d0s0        ONLINE    0     0     0
          c3t50000393A803B13Ed0s0        ONLINE    0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则创建冗余配置或恢复使用那些设备。有关如何创建冗余配置或恢复设备使用的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

A.2.4

活动状态更换步骤示例（当 PCIe 总线的动态分配可用时）

本节介绍使用 "图 A-2 在 2BB 配置中可继续操作的配置示例" 所述的 2BB 配置系统的 PPAR DR 活动状态更换 BB#01 的步骤示例。此示例适用于 PCIe 总线可动态分配的环境（Oracle VM Server for SPARC 3.2 或更高版本的 XCP 2240 或更高版本和 Oracle Solaris 11.2 SRU11.2.8.4.0 或更高版本的根域）。

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。
在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

1. 登录主 XSCF。
- 执行 showbbstatus 命令来检查您已经登录的 XSCF 是否是主 XSCF。
- 如果您已经登录到备用 XSCF，注销然后再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

2. 执行 **showhardconf** 命令检查需要更换的 **SPARC M10-4S** 中的 **XSCF** 的 [Status] 是否是 "Normal"。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Locked;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;
+ Freq:3.000 GHz; Type:0x10;
+ Core:16; Strand:2;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
:
```

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷, 您将无法使用 PPAR DR 执行活动状态更换。
在进行维护作业时, 必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

3. 执行 **showbbstatus** 命令来确认需要更换的 **SPARC M10-4S** 中的 **XSCF** 不是主 **XSCF**。

```
XSCF> showbbstatus
BB#00 (Master)
```

如果需要更换的 SPARC M10-4S 是主 XSCF, 则执行 **switchscf** 命令切换 XSCF。

```
XSCF> switchscf -t Standby
The XSCF unit switch between the Active and Standby states.
Continue? [y|n] :y
```


注 - 释放系统板前，请确认 XSCF 已经切换并重新启动。

4. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

5. 检查逻辑域的运行状态和资源使用状况。

a. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义。

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

下面的示例表明控制域、两个根域和两个来宾域正在运行。

检查所有的域是处于表明 Oracle Solaris 处于运行状态的 "active"，还是 "inactive" 状态。如果存在处于 OpenBoot PROM 状态或绑定状态的域，则物理分区的动态重新配置可能失败。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active     -n-cv-   UART    64      56G       0.0%    1h 33m
guest0        active     -n----   5100    64      64G       3.1%    2s
guest1        active     -n----   5101    64      64G       1.6%    18m
root-dom0     active     -n--v-   5000    32      32G       3.1%    17m
root-dom1     active     -n--v-   5001    32      32G       3.1%    17m
```

b. 执行指定 **-a** 选项的 **ldm list-devices** 命令来检查资源的使用状况。

在下面的示例中，指定 **-a** 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。

```
# ldm list-devices -a
CORE
  ID      %FREE    CPUSSET
  0        0        (0, 1)
  4        0        (8, 9)
  8        0        (16, 17)
(省略)
  944      0        (1888, 1889)
  948      0        (1896, 1897)
  952      0        (1904, 1905)
  956      0        (1912, 1913)
VCPU
  PID      %FREE    PM
  0        0        no
  1        0        no
  8        0        no
  9        0        no
```

(省略)		
1904	0	no
1905	0	no
1912	0	no
1913	0	no
(省略)		

6. 释放控制域中系统卷和 I/O 设备的冗余配置。
 若需释放 BB#01，请释放需要更换的 SPARC M10-4S 的 I/O 设备和控制域中使用的 I/O 设备。有关取消冗余配置的步骤的详情，请参见该冗余配置软件的文档。

a. 取消控制域中系统卷的冗余配置。

下面的示例介绍如何取消控制域中系统卷的 ZFS 镜像功能。

在控制域中执行 `zpool status` 命令检查镜像配置状态。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                               ONLINE     0     0     0
    mirror-0                          ONLINE     0     0     0
      c2t50000393E802CCE2d0s0        ONLINE     0     0     0
      c3t50000393A803B13Ed0s0        ONLINE     0     0     0
errors: No known data errors
```

执行 `zpool detach` 命令从镜像配置中释放磁盘。

```
# zpool detach rpool c3t50000393A803B13Ed0
```

执行 `zpool status` 命令确认镜像配置已经取消。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                               ONLINE     0     0     0
    mirror-0                          ONLINE     0     0     0
      c2t50000393E802CCE2d0s0        ONLINE     0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则移除冗余配置或停止使用那些设备。有关如何取消冗余配置或停止使用设备的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

b. 删除控制域的 I/O 配置。

通过动态重新配置，从分配给控制域的物理 I/O 设备中删除 BB#01 中的根合成

体。
执行 `ldm list-io` 命令检查分配给 `primary` 的根合成体。

下面的示例表明 BB1 设备的根合成体是 PCIE8 和 PCIE12。

```
# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary      IOV
PCIE4          BUS      PCIE4      primary      IOV
PCIE8          BUS      PCIE8      primary      IOV
PCIE12         BUS      PCIE12     primary      IOV
/BB0/CMUL/NET0 PCIE      PCIE0      primary      OCC
/BB0/CMUL/SASHBA PCIE      PCIE0      primary      OCC
/BB0/CMUL/NET2  PCIE      PCIE4      primary      OCC
/BB1/CMUL/NET0  PCIE      PCIE8      primary      OCC
/BB1/CMUL/SASHBA PCIE      PCIE8      primary      OCC
/BB1/CMUL/NET2  PCIE      PCIE12     primary      OCC
```

执行 `ldm remove-io` 命令从主域删除 PCIE8 和 PCIE12 。

```
# ldm remove-io PCIE8 primary
# ldm remove-io PCIE12 primary
```

执行 `ldm list-io` 命令确认 BB#01 中的根合成体已经从控制域删除。

```
# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary      IOV
PCIE4          BUS      PCIE4      primary      IOV
/BB0/CMUL/NET0 PCIE      PCIE0      primary      OCC
/BB0/CMUL/SASHBA PCIE      PCIE0      primary      OCC
/BB0/CMUL/NET2  PCIE      PCIE4      primary      OCC
```

c. 取消分配给来宾域的虚拟 I/O 设备的冗余配置。

若需首先关闭分配 BB#01 中的根合成体的根域 (`root-dom1`)，请登录各来宾域，然后从 `root-dom1` 中取消虚拟 I/O 设备的冗余配置。
有关如何使用冗余配置软件的详情，请参见该冗余配置软件的文档。
在下面的示例中，将虚拟网络设备 (`vnet1`) 从 IPMP 配置中取消。有关命令的详情，请参见 Oracle Solaris 的手册。
登录来宾域 (`guest0`)。

```
# ldm list-domain
NAME          STATE      FLAGS      CONS      VCPU      MEMORY     UTIL      UPTIME
primary       active     -n-cv-    UART      64        56G        0.0%      4h 17m
guest0        active     -n----    5100      64        64G        0.0%      1h 13m
guest1        active     -n----    5101      64        64G        0.0%      1h 4m
root-dom0     active     -n--v-    5000      32        32G        0.0%      1h 47m
root-dom1     active     -n--v-    5001      32        32G        0.0%      1h 19m
# telnet localhost 5100
....
guest0#
```

执行 `dladm show-phys` 命令检查虚拟网络接口 (vnet1) 和网络接口的名称 (net1) 之间的映射。

```
guest0# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	0	unknown	vnet0
net1	Ethernet	up	0	unknown	vnet1

执行 `ipmpstat -i` 命令检查配置 IPMP 的网络接口的配置信息。

```
guest0# ipmpstat -i
```

INTERFACE	ACTIVE	GROUP	FLAGS	LINK	PROBE	STATE
net0	yes	ipmp0	-smbM--	up	disabled	ok
net1	no	ipmp0	is-----	up	disabled	ok

执行 `if_mpadm -d` 命令从 IPMP 组释放 net1，然后执行 `ipmpstat -i` 命令确认已经将其释放。下面的示例确认 STATE 是 offline。

```
guest1# if_mpadm -d net1
guest1# ipmpstat -i
```

INTERFACE	ACTIVE	GROUP	FLAGS	LINK	PROBE	STATE
net0	yes	ipmp0	-smbM--	up	disabled	ok
net1	no	ipmp0	-s---d-	up	disabled	offline

执行 `ipadm delete-ip` 命令删除 net1。

```
guest0# ipadm delete-ip net1
```

同样，对来宾域 (guest1) 执行相同的释放处理。

d. 从根域移除已分配的虚拟 I/O 设备，以便将其停止。

按照下述步骤，执行 `ldm remove-vdisk` 和 `ldm remove-vnet` 命令从根域移除已分配的虚拟磁盘 (vdisk) 和虚拟网络设备 (vnet)，以便将其停止。

下面的示例显示移除使用 BB#01 根域 (root-dom0) 的虚拟 I/O 服务的虚拟磁盘 (vdisk10) 和虚拟网络设备 (vnet10) 的命令的执行。

```
# ldm remove-vdisk vdisk10 guest0
# ldm remove-vnet vnet10 guest0
```

对来宾域 (guest1) 执行相同的删除操作。

7. 检查 I/O 设备的资源使用状况，然后取消需要更换的 SPARC M10-4S 中的所有 I/O 设备。

a. 检查分配需要释放的 SPARC M10-4S 中的根合成体的逻辑域。

执行 `ldm list-io` 命令，以检查分配 BB#01 中 SPARC M10-4S 的根合成体的逻辑域。

下面的示例表明仅 root-dom1 具有以 "/BB1/" 开始的 PCIe 端点。您可以看到 PCIe 端点根合成体 (BUS) PCIE9、PCIE10、PCIE11、PCIE13、PCIE14 和 PCIE15 被分配给 root-dom1。

```
# ldm list-io
NAME                                TYPE    BUS      DOMAIN    STATUS
----                                -
PCIE0                              BUS     PCIE0    primary   IOV
PCIE1                              BUS     PCIE1    root-dom0 IOV
PCIE2                              BUS     PCIE2    root-dom0 IOV
PCIE3                              BUS     PCIE3    root-dom0 IOV
PCIE4                              BUS     PCIE4    primary   IOV
PCIE5                              BUS     PCIE5    root-dom0 IOV
PCIE6                              BUS     PCIE6    root-dom0 IOV
PCIE7                              BUS     PCIE7    root-dom0 IOV
PCIE8                              BUS     PCIE8
PCIE9                              BUS     PCIE9    root-dom1 IOV
PCIE10                             BUS     PCIE10   root-dom1 IOV
PCIE11                             BUS     PCIE11   root-dom1 IOV
PCIE12                             BUS     PCIE12
PCIE13                             BUS     PCIE13   root-dom1 IOV
PCIE14                             BUS     PCIE14   root-dom1 IOV
PCIE15                             BUS     PCIE15   root-dom1 IOV
....
/BB1/CMUL/NET0                     PCIE    PCIE8
/BB1/CMUL/SASHBA                    PCIE    PCIE8
/BB1/PCI0                           PCIE    PCIE9    root-dom1OCC
/BB1/PCI3                           PCIE    PCIE10   root-dom1OCC
/BB1/PCI4                           PCIE    PCIE10   root-dom1OCC
/BB1/PCI7                           PCIE    PCIE11   root-dom1OCC
/BB1/PCI8                           PCIE    PCIE11   root-dom1OCC
/BB1/CMUL/NET2                      PCIE    PCIE12
/BB1/PCI1                           PCIE    PCIE13   root-dom1OCC
/BB1/PCI2                           PCIE    PCIE13   root-dom1OCC
/BB1/PCI5                           PCIE    PCIE14   root-dom1OCC
/BB1/PCI6                           PCIE    PCIE14   root-dom1OCC
/BB1/PCI9                           PCIE    PCIE15   root-dom1OCC
/BB1/PCI10                          PCIE    PCIE15   root-dom1OCC
```

b. 停止分配需要释放的 **SPARC M10-4S** 中的根合成体的逻辑域。

下面的示例执行 `ldm stop-domain` 和 `ldm unbind-domain` 命令释放根域 (root-dom1)，并表明根域处于 inactive 状态。

```
# ldm stop-domain root-dom1
LDom root-dom1 stopped
# ldm unbind-domain root-dom1
# ldm list-domain
NAME            STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary         active  -n-cv-   UART    64      56G       0.2%    4h 59m
guest0          active  -n----   5100    64      64G       0.0%    1h 55m
guest1          active  -n----   5101    64      64G       0.0%    1h 46m
root-dom0       active  -n--v-   5000    32      32G       0.0%    2h 29m
```

- c. 确认需要更换的 **SPARC M10-4S** 中的所有 **I/O** 设备已经被取消。
执行 `ldm list-io` 命令确认已经释放所有 I/O 设备。

```
# ldm list-io
NAME                                TYPE    BUS      DOMAIN    STATUS
----                                -
PCIE0                              BUS     PCIE0    primary   IOV
PCIE1                              BUS     PCIE1    root-dom0 IOV
PCIE2                              BUS     PCIE2    root-dom0 IOV
PCIE3                              BUS     PCIE3    root-dom0 IOV
PCIE4                              BUS     PCIE4    primary   IOV
PCIE5                              BUS     PCIE5    root-dom0 IOV
PCIE6                              BUS     PCIE6    root-dom0 IOV
PCIE7                              BUS     PCIE7    root-dom0 IOV
PCIE8                              BUS     PCIE8
PCIE9                              BUS     PCIE9
PCIE10                             BUS     PCIE10
PCIE11                             BUS     PCIE11
PCIE12                             BUS     PCIE12
PCIE13                             BUS     PCIE13
PCIE14                             BUS     PCIE14
PCIE15                             BUS     PCIE15
(省略)
```

8. 手动减少分配给逻辑域的 **CPU** 内核数量和内存资源。
下面解释手动减少分配给逻辑域的 CPU 内核数量和内存资源的步骤，而不通过指定 `-m unbind=resource` 选项来满足 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中 ["CPU 内核和内存的配置"](#) 的条件。
- 例如，如果由于错误，`deleteboard` 命令失败，在某些情况下按照以下步骤可以执行 `deleteboard` 命令。
- 当指定 `deleteboard` 命令的 `-m unbind=resource` 选项时，转到步骤 9。
- a. 检查 **CPU** 内核数量，然后将其删除。
- 由于释放 SPARC M10-4S，可使用的 CPU 内核数量将减少，因此首先使用以下步骤来提前减少分配给逻辑域的 CPU 内核数量。
- i. 释放 **SPARC M10-4S** 后，检查 **CPU** 内核数量。
- 从 XSCF 执行 `showpparinfo` 命令来检查 CPU 内核数量，不包括需要释放的 SPARC M10-4S 中的 CPU 内核数量。
- 在下面的示例中，如果将要释放编号为 01-0 的物理系统板 (PSB)，则必须计算和另一个 PSB 编号为 00-0 的 CPU 内核的总和。因此， $16 + 16 + 16 + 16 = 64$ 个内核。

```
XSCF> showpparinfo -p 0
PPAR#00 Information:
-----
CPU(s)                :      8
CPU Cores              :     128
```

```
CPU Threads      :      256
Memory size (GB) :      256
CoD Assigned (Cores) :      256

CPU(s) :
-----
  PID  PSB   CPU#   Cores   Threads
  00   00-0   0       16       32
  00   00-0   1       16       32
  00   00-0   2       16       32
  00   00-0   3       16       32
  00   01-0   0       16       32
  00   01-0   1       16       32
  00   01-0   2       16       32
  00   01-0   3       16       32
(省略)
```

ii.检查被分配给各逻辑域的 CPU 内核的总数。

执行 ldm list-devices -a core 命令。在 %FREE 栏中除 100 之外的数值所在行的数量是分配给逻辑域的 CPU 内核的总数。

在下面的示例中，执行 ldm list-devices -a core 命令，使用 -p 选项执行了检查。因此，可以看到 112 个内核被绑定到整个逻辑域。

```
# ldm list-devices -a core
CORE
  ID      %FREE    CPUSSET
    0         0    (0, 1)
    4         0    (8, 9)
    8         0   (16, 17)
   12         0   (24, 25)
(省略)
# ldm list-devices -a -p core | egrep -v "CORE|VERSION|free=100" | wc -l
112
```

iii.从 SPARC M10-4S 释放的结果计算内核不足的数量。

使用下面的公式，计算释放 SPARC M10-4S 后将要产生的 CPU 内核不足的数量。

CPU 内核不足的数量 = 逻辑域中使用的内核数量（步骤 ii） - 释放后的物理内核的数量（步骤 i）

在步骤 i 和步骤 ii 示例中, 可以得出不足的数量将是 112 个内核（使用中） - 64 个内核（剩余） = 48 个内核。

iv.如果发生了 CPU 内核不足，请考虑要删除哪个逻辑域。

如果从步骤 iii 的结果可以得出将要发生 CPU 内核不足，有必要从逻辑域删除 CPU 内核。

执行 ldm list-domain 命令检查分配给各逻辑域处于 active 或 bound 状态的 CPU 内核数量，以及检查 CPU 内核需要从中删除的逻辑域。

在下面的示例中，可以得出将 32 个内核 (64vcpu) 分配给 primary，将 32 个内核 (64vcpu) 分配给 guest0，将 32 个内核 (64vcpu) 分配给 guest1，以及将 16 个内核 (32vcpu) 分配给 root-dom0。在此示例中，需要删除 48 个内核。

这样, 将从每个 primary、guest0 和 guest1 中删除 16 个内核。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.1%	18h 17m
guest0	active	-n----	5100	64	64G	0.0%	15h 13m
guest1	active	-n----	5101	64	64G	0.0%	15h 4m
root-dom0	active	-n--v-	5000	32	32G	0.0%	15h 47m
root-dom1	inactive	-----		32	32G		

v. 执行 `ldm remove-core` 命令从目标逻辑域删除 CPU 内核。

在下面的示例中, 从每个 primary、guest0 和 guest1 中删除 16 个内核, 然后进行检查确认是否已经实际将其删除。

```
# ldm remove-core 16 primary
# ldm remove-core 16 guest0
# ldm remove-core 16 guest1
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	32	56G	0.0%	18h 19m
guest0	active	-n----	5100	32	64G	0.0%	15h 15m
guest1	active	-n----	5101	32	64G	0.0%	15h 5m
root-dom0	active	-n--v-	5000	32	32G	0.0%	15h 49m
root-dom1	inactive	-----	5001	32	32G		

```
# ldm list-devices -a -p core | egrep -v "CORE|VERSION|free=100" | wc -l
64
```

b. 检查并删除内存资源。

由于释放 SPARC M10-4S, 可用的内存区将减少, 因此使用以下步骤删除分配给逻辑域的内存资源。

i. 检查内存的连续区域 (内存块) 的使用状况。

执行 `prtdiag` 命令和 `ldm list-devices -a memory` 命令检查分配给各逻辑域的内存块, 并检查分配未分配的内存块的 SPARC M10-4S。

首先, 执行 `prtdiag` 命令检查内存中物理地址和 SPARC M10-4S 之间的对应关系。

```
# prtdiag
(省略)
```

```
===== Physical Memory Configuration =====
Segment Table:
-----
```

Base Address	Segment Size	Interleave Factor	Bank Size	Contains Modules
0x7e0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUL/CMP0/MEM00A
0x7c0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUL/CMP1/MEM10A
0x7a0000000000	32 GB	4	8 GB	/BB0/CMUU/CMP0/MEM00A

(省略)					
0x780000000000	32 GB	4	8 GB	/BB0/CMUU/CMP1/MEM10A	
(省略)					
0x760000000000	32 GB	4	8 GB	/BB1/CMUL/CMP0/MEM00A	
(省略)					
0x740000000000	32 GB	4	8 GB	/BB1/CMUL/CMP1/MEM10A	
(省略)					
0x720000000000	32 GB	4	8 GB	/BB1/CMUU/CMP0/MEM00A	
(省略)					
0x700000000000	32 GB	4	8 GB	/BB1/CMUU/CMP1/MEM10A	
(省略)					

本示例的结果按内存中物理地址的升序重新排列。下表列出了物理地址和 SPARC M10-4S 之间的对应关系。

表 A-5 物理地址和 SPARC M10-4S 之间对应关系的示例

基地址（物理地址）	SPARC M10-4S 的模块化架构配置
0x700000000000 和之后	BB1
0x720000000000 和之后	BB1
0x740000000000 和之后	BB1
0x760000000000 和之后	BB1
0x780000000000 和之后	BB0
0x7a0000000000 和之后	BB0
0x7c0000000000 和之后	BB0
0x7e0000000000 和之后	BB0

然后，执行 `ldm list-devices -a memory` 命令检查分配给逻辑域的内存连续区域（本文档下文中称为内存块）和未分配的内存块。

在下面的示例中，执行了 `ldm list-devices -a memory` 命令。

各参数的含义如下。

PA: 内存块的开始物理地址

SIZE: 内存块的容量

BOUND: 分配内存块的逻辑域的名称。空白是未分配的区域，`_sys_` 是未被分配给逻辑域的控制区。

# ldm list-devices -a memory		
MEMORY		
PA	SIZE	BOUND
0x700000000000	32G	root-dom0
0x720000000000	32G	
0x740000000000	32G	guest0
0x760000800000	1272M	_sys_
0x760050000000	31488M	guest1
0x780000000000	32G	guest0
0x7a0000000000	32G	guest1

0x7c0000000000	28G	primary
0x7c0700000000	4G	
0x7e0000800000	1272M	_sys_
0x7e0050000000	512M	_sys_
0x7e0070000000	256M	_sys_
0x7e0080000000	28G	primary
0x7e0780000000	1280M	guest1
0x7e07d0000000	768M	

通过将上述显示的结果与使用 `prtdiag` 命令检查的物理位置相结合，可以得到内存块的使用状况如下。

表 A-6 内存块使用状况的示例

SPARC M10-4S	物理地址	容量	逻辑域
BB1（更换的目标）	0x700000000000	32 GB	root-dom0
	0x720000000000	32 GB	未分配
	0x740000000000	32 GB	guest0
	0x760050000000	31,488 MB	guest1
BB0	0x780000000000	32 GB	guest0
	0x7a0000000000	32 GB	guest1
	0x7c0000000000	28 GB	primary
	0x7c0700000000	4 GB	未分配
	0x7e0080000000	28 GB	primary
	0x7e0780000000	1,280 MB	guest1
	0x7e07d0000000	768 MB	未分配

ii.检查移动源内存块的容量和数量。

参照内存块使用状况的检查结果时，检查分配给需要更换的 SPARC M10-4S 的内存块（在下文中称为“源内存块”）。

对于“表 A-6 内存块使用状况的示例”而言，32 GB x 2 和 31,488 MB x 1 的内存块是正在 BB1 中被用作分配给逻辑域的内存块。

iii.确定从其中删除内存的逻辑域，以及数量。

然后，检查分配给各逻辑域的内存块的位置，并通过删除内存块单元中的内存以及通过减少内存块的容量来确保源内存块可被移动至将不被释放的 SPARC M10-4S 的未分配的内存块（本文档下文中称为“目标内存块”）。

因此，可作出最终决定，将删除多少内存和从中将其删除的逻辑域。

支持以下方法。

- 通过完全删除源内存块，减少需要被移动的内存块的数量。
- 完全删除不作为释放对象的 SPARC M10-4S 上被分配到逻辑域的内存块，通过增加可用空间来增加移动目标内存空间。
- 减少源内存块的容量，以便其适用于目标上的可用区域。
- 通过减少源内存块的容量和通过减少使用中的目标的内存块的容量，增加目

标上的可用内存块的数量以便可进行移动。

注 - 减少容量后，可用内存块将不连续（将成为碎片）。即使将多个小内存块删除以增加可用区域的数量，可用区域也不连续。如果源内存块的连续区域较大，将无法移动。在这种情况下，删除源内存块来调整其容量。

注 - 当考虑此种情况时，可以通过选择可删除的与现有的内存块尽可能具有相同容量的内存块，可减少删除之后成为碎片的连续区域的可能性。这将增加内存块成功移动的可能性。

注 - 如果过多的内存被删除，这将增大逻辑域中的内存的负担，可能造成 Oracle Solaris 挂起等问题。通过使用 `vmstat` 命令和检查 "free" 容量作为粗略参考，有必要注意勿删除过多内存。

基于 "表 A-6 内存块使用状况的示例" 研究删除计划。

根据此种考虑，制定从 `root-dom0` 中删除 4 GB，从 `guest0` 中删除 32 GB，从 `guest1` 中删除 31,488 MB 以及从 `primary` 中删除 28 GB 的计划，以 "表 A-7 内存块删除计划" 为例，如下所示。

表 A-7 内存块删除计划

SPARC M10-4S	容量	逻辑域	删除计划
BB1（更换的目标）	32 GB	root-dom0	此区域减少 4 GB 至 28 GB，从 BB0 上的 <code>primary</code> 删除 28 GB，然后实现移动。
	32 GB	未分配	-
	32 GB	guest0	由于在 BB0 上存在 32-GB <code>guest0</code> 内存，将其删除。
	31,488 MB	guest1	由于在 BB0 上存在 32-GB <code>guest1</code> 内存，将其删除。
BB0	32 GB	guest0	保留。
	32 GB	guest1	保留。
	28 GB	primary	保留。
	4 GB	未分配	-
	28 GB	primary	将其删除，以便移动 <code>root-dom0</code> 的 28 GB。
	1,280 MB	guest1	保留。
	768 MB	未分配	-

iv.从逻辑域手动删除内存。

按照在步骤 iii 中制定的内存删除计划，使用 `ldm remove-memory` 命令从逻辑域删除内存。

下面的示例显示了根据 "表 A-7 内存块删除计划" 删除内存的命令的执行。

```
# ldm remove-memory 4G root-dom0
# ldm remove-memory 32G guest0
# ldm remove-memory 31488M guest1
# ldm remove-memory 28G primary
```

v. 检查已删除的内存块的状态。

执行 `ldm list-devices -a memory` 命令，通过参考删除的结果来检查该布置是否可实现移动。如果存在任何无法移动的内存块，根据结果考虑哪些额外的内存块需要删除，然后将其删除。

在下面的示例中，通过顺次比较分配给 BB1 的大容量和 BB0 上可用区域内的大容量可很容易检查是否可移动。

```
# ldm list-devices -a memory
MEMORY
  PA                SIZE                BOUND
(BB1)
  0x700000000000    256M                root-dom0 →可通过分割 4-GB 目标移动
  0x700010000000    4G
  0x700110000000    28416M              root-dom0 →可移动到 0x780000000000
                                           (28 GB)
  0x720000000000    32G
  0x740000000000    256M                guest0 →可通过分割 4-GB 目标移动
  0x740010000000    4G
  0x740110000000    28416M              guest0 →由于无目标需要再次删除 (*)
  0x760000800000    1272M                _sys_
  0x760050000000    256M                guest1 →可通过分割 4-GB 区域移动
  0x760060000000    1792M
  0x7600d0000000    29440M              guest1 →可移动到 0x7a0000000000
                                           (29-GB)
(BB0)
  0x780000000000    28G
  0x780700000000    4G                  guest0 ←可从 root-dom0
                                           (0x700110000000) 移动
  0x7a0000000000    29G
  0x7a0740000000    3G                  guest1 ←可从 guest1
                                           (0x7600d0000000) 移动
  0x7c0000000000    256M                primary
  0x7c0010000000    4G
  0x7c0110000000    24320M              primary ←可从 root-dom0、 guest0 和
                                           guest1 移动 256 MB
  0x7c0700000000    4G
  0x7e0000800000    1272M                _sys_
  0x7e0050000000    512M                _sys_
  0x7e0070000000    256M                _sys_
  0x7e0080000000    24G                  ←guest0 (0x740110000000)
                                           移动没有足够可
  0x7e0680000000    4G                  primary
  0x7e0780000000    1280M              guest1
  0x7e07d0000000    768M
```

在上述示例中，标记有 (*) 的目标仅有 24 GB (24,576 MB) 的可用空间，因此从源 (BB1) 上的 28416-MB 区域 (guest0) 删除 3,840 MB，然后重复检查。在

下面的示例中, 可以得出所有内存块都可移动。

```
# ldm remove-memory 3840M guest0
# ldm list-devices -a memory
MEMORY
  PA                SIZE                BOUND
(BB1)
  0x700000000000    256M                root-dom0
  0x700010000000     4G
  0x700110000000    28416M              root-dom0
  0x720000000000    32G
  0x740000000000    256M                guest0
  0x740010000000    7936M
  0x740200000000    24G                guest0 →可移动到 0x7e0080000000 (24 G)
  0x760000800000    1272M              _sys_
  0x760050000000    256M                guest1
  0x760060000000    1792M
  0x7600d0000000    29440M             guest1
(BB0)
  0x780000000000    28G
  0x780700000000     4G                guest0
  0x7a0000000000    29G
  0x7a0740000000     3G                guest1
  0x7c0000000000    256M              primary
  0x7c0010000000     4G
  0x7c0110000000    24320M             primary
  0x7c0700000000     4G
  0x7e0000800000    1272M              _sys_
  0x7e0050000000    512M              _sys_
  0x7e0070000000    256M              _sys_
  0x7e0080000000    24G                ←可从 guest0
                                           (0x740200000000) 移动
  0x7e0680000000     4G                primary
  0x7e0780000000    1280M             guest1
  0x7e07d0000000    768M
```

9. 从物理分区释放 SPARC M10-4S 的系统板 (PSB<BB>)。

a. 执行 **deleteboard -c disconnect** 命令将 PSB 从物理分区释放。

```
XSCF> deleteboard -c disconnect 01-0
PSB#01-0 will be unconfigured from PPAR immediately. Continue?
[y|n] :y
Start unconfigure preparation of PSB. [1200sec]
0end
Unconfigure preparation of PSB has completed.
Start unconfiguring PSB from PPAR. [7200sec]
0..... 30..... 60....end
Unconfigured PSB from PPAR.
PSB power off sequence started. [1200sec]
0..... 30..... 60..... 90.....120.....150.....end
Operation has completed.
```

b. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。

结束值 0 表明 deleteboard 命令正常终止。

如果结束值非 0，或如果在执行 deleteboard 命令时显示错误信息，则表明 deleteboard 命令异常终止。根据错误信息，通过参考 "C.1.2 deleteboard" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

- c. 执行 showboards 命令检查 PSB 状态。
- 确认需要更换的 SPARC M10-4S 内的 PSB 是否处于 “Assigned” 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 “n”。

XSCF> showboards -p 0							
PSB	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

10. 执行 replacefru 命令来更换 SPARC M10-4S。

```
XSCF> replacefru
```

注 - 关于使用 replacefru 命令更换 SPARC M10-4S 的详情，请参见《Fujitsu M10-4/Fujitsu M10-4S/SPARC M10-4/SPARC M10-4S Service Manual》中的 "5.8 Releasing a SPARC M10-4/M10-4S FRU from the System with the replacefru Command" 和 "6.2 Incorporating a SPARC M10-4/M10-4S FRU into the System with the replacefru Command"。

- 11. 将 PSB 整合到物理分区。
 - a. 执行 showboards 命令检查 PSB 状态。
- 确认需要更换的 SPARC M10-4S 内的 PSB 是否处于 Assigned 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 “n”。

XSCF> showboards -p 0							
PSB	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

- b. 执行 addboard -c configure 命令将 PSB 整合到物理分区。
- 若需恢复初始逻辑域配置，请执行指定 -m bind=resource 选项的 addboard -c configure 命令。

```
XSCF> addboard -c configure -m bind=resource -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
```

```
0..... 30..... 60..... 90.....120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
810.....840.....870.....900.....930.....960.....end
Connected PSB to PPAR.
Start configuring PSB to Logical Domains (LDoms) Manager.
[1800sec] 0.....end
Configured PSB to Logical Domains (LDoms) Manager.
Operation has completed.
```

注 - 如果执行 **addboard** 命令过程中出现错误信息，请参见 ["C.1.1 addboard"](#)，然后识别错误并采取应对措施。

c. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。

结束值 0 表明 **addboard** 命令正常终止。

如果结束值非 0，或如果在执行 **addboard** 命令时显示错误信息，则表明 **addboard** 命令异常终止。根据错误信息，通过参考 ["C.1.1 addboard"](#) 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

d. 执行 **showboards** 命令检查 **PSB** 状态。

在已更换的 SPARC M10-4S 中的 PSB 已经被成功整合后，确认 [Conn] 和 [Conf] 栏是否都显示 "y"。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

12. 检查逻辑域的运行状态。

a. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

b. 执行 **ldm list-domain** 命令来确认 **SPARC M10-4S PSB (BB)** 添加后逻辑域的运行状态未发生变化。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义。

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	32	28G	64%	2h 54m
guest0	active	-n----	5100	32	61876M	42%	2h 54m
guest1	active	-n----	5101	32	62388M	11%	2h 54m
root-dom0	active	-n--v-	5000	32	28G	0.0%	2h 54m
root-dom1	inactive	-----		32	32G	0.0%	2h 54m

13. 恢复已删除的资源。

如果使用 `deleteboard` 命令已经删除分配给逻辑域的资源，则使用 `ldm add-core` 和 `ldm add-memory` 命令添加资源，以便将其恢复。

在下面的示例中，删除的 CPU 内核和内存被添加，以便在更换 SPARC M10-4S 之前恢复资源。

```
# ldm add-core 16 primary
# ldm add-core 16 guest0
# ldm add-core 16 guest1
# ldm add-memory 28G primary
# ldm add-memory 4G root-dom0
# ldm add-memory 36608M guest0
# ldm add-memory 31488M guest1
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.2%	4h 59m
guest0	active	-n----	5100	64	64G	0.0%	1h 55m
guest1	active	-n----	5101	64	64G	0.0%	1h 46m
root-dom0	active	-n--v-	5000	32	32G	0.0%	2h 29m
root-dom1	inactive	-----		32	32G		

14. 重新启动 I/O 设备的使用。

a. 重新分配根合成体。

执行 `ldm bind-domain` 和 `ldm start-domain` 命令，以便启动在 `unbind` 状态下分配已更换的 SPARC M10-4S 中的根合成体的根域。

下面的示例启动在 `unbind` 状态下的根域 (`root-dom1`)，然后确认是否已经启动。

```
# ldm bind-domain root-dom1
# ldm start-domain root-dom1
LDom root-dom1 started
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.2%	3h 8m
guest0	active	-n----	5100	64	64G	0.0%	3h 8m
guest1	active	-n----	5101	64	64G	0.0%	3h 8m
root-dom0	active	-n--v-	5000	32	32G	0.0%	3h 8m
root-dom1	active	-n--v-	5001	32	32G	7.3%	8s

执行 `ldm list-io` 命令确认物理 I/O 设备是否已经分配给刚刚启动的根域。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV
....				
/BB1/CMUL/NET0	PCIE	PCIE8		UNK
/BB1/CMUL/SASHBA	PCIE	PCIE8		UNK
/BB1/PCI0	PCIE	PCIE9	root-dom1OCC	
/BB1/PCI3	PCIE	PCIE10	root-dom1OCC	
/BB1/PCI4	PCIE	PCIE10	root-dom1OCC	
/BB1/PCI7	PCIE	PCIE11	root-dom1OCC	
/BB1/PCI8	PCIE	PCIE11	root-dom1OCC	
/BB1/CMUL/NET2	PCIE	PCIE12		UNK
/BB1/PCI1	PCIE	PCIE13	root-dom1OCC	
/BB1/PCI2	PCIE	PCIE13	root-dom1OCC	
/BB1/PCI5	PCIE	PCIE14	root-dom1OCC	
/BB1/PCI6	PCIE	PCIE14	root-dom1OCC	
/BB1/PCI9	PCIE	PCIE15	root-dom1OCC	
/BB1/PCI10	PCIE	PCIE15	root-dom1OCC	

b. 将虚拟 I/O 设备从根域添加到来宾域。

执行 ldm add-vdisk 和 ldm add-vnet 命令，将支持已经启动的根域的虚拟 I/O 服务的虚拟磁盘 (vdisk) 和虚拟网络设备 (vnet) 添加到各个来宾域。

下面的示例显示添加使用 BB#01 根域 (root-dom1) 的虚拟 I/O 服务的虚拟磁盘 (vdisk10) 和虚拟网络设备 (vnet10) 的命令的执行。

```
# ldm add-vdisk id=1 vdisk10 vol110@vds1 guest0
# ldm add-vnet id=1 vnet10 vsw10 guest0
```

对来宾域 (guest1) 执行相同的添加操作。

注 - 若需再次添加虚拟 I/O 设备，有必要提前指定已分配的 ID。您可以在虚拟 I/O 设备被删除之前，在被使用的状态下，从 ldm list -l 命令的执行结果检查 ID。

c. 将分配给来宾域的虚拟 I/O 设备整合到冗余配置中。

启动分配 BB#1 中的根合成体的根域 (root-dom1) 后, 对应各来宾域的虚拟 I/O 设备服务也被启动。

登录各来宾域, 然后从以前被取消的 root-dom1 中将虚拟 I/O 设备整合到冗余配置。有关如何使用冗余配置软件的详情, 请参见该冗余配置软件的文档。

下面的示例介绍将虚拟网络设备 (vnet1) 整合到 IPMP 配置的示例。有关命令的详情, 请参见 Oracle Solaris 的手册。

首先登录来宾域 (guest0)。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    64      56G       0.0%    4h 17m
guest0        active    -n----- 5100    64      64G       0.0%    1h 13m
guest1        active    -n----- 5101    64      64G       0.0%    1h  4m
root-dom0     active    -n--v-   5000    32      32G       0.0%    1h 47m
root-dom1     active    -n--v-   5001    32      32G       0.0%    1h 19m
# telnet localhost 5100
....
guest0#
```

执行 `dladm show-phys` 命令检查虚拟网络接口 (vnet1) 和网络接口的名称 (net1) 之间的映射。

```
guest0# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      up        0      unknown   vnet0
net1          Ethernet      up        0      unknown   vnet1
```

执行 `ipadm create-ip` 命令、`ipadm set-ifprop` 命令以及 `ipadm add-ipmp` 命令将 net1 注册为 ipmp0 的备用设备。

```
guest0# ipadm create-ip net1
guest0# ipadm set-ifprop -p standby=on -m ip net1
guest0# ipadm add-ipmp -i net1 ipmp0
```

执行 `ipmpstat -i` 命令确认配置 IPMP 的网络接口的 STATE 是否显示 ok。

```
guest0# ipmpstat -i
INTERFACE ACTIVE GROUP FLAGS LINK PROBE STATE
net0 yes ipmp0 -smbM-- up disabled ok
net1 no ipmp0 -s---d- up disabled ok
```

对另一个来宾域 (guest1), 执行相同的步骤。

15. 将控制域上的系统卷和 I/O 设备恢复到冗余配置。

a. 添加控制域的根合成体配置。

在以前从控制域中被移除的 BB#01 中添加根合成体。

执行 `ldm list-io` 命令检查未分配的根合成体。

下面的示例表明 BB1 设备的根合成体，PCIE8 和 PCIE12 未被分配。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
(省略)				

执行 ldm add-io 命令将 PCIE8 和 PCIE12 添加到主域。

```
# ldm add-io PCIE8 primary
# ldm add-io PCIE12 primary
```

执行 ldm list-io 命令确认 BB#01 中的根合成体已经添加至控制域。

# ldm list-io grep primary					
PCIE0	BUS	PCIE0	primary	IOV	
PCIE4	BUS	PCIE4	primary	IOV	
PCIE8	BUS	PCIE8	primary	IOV	
PCIE12	BUS	PCIE12	primary	IOV	
/BB1/CMUL/NET0	PCIE	PCIE0	primary	OCC	
/BB1/CMUL/SASHBA	PCIE	PCIE0	primary	OCC	
/BB1/CMUL/NET2	PCIE	PCIE4	primary	OCC	

b. 将控制域中的系统卷设置为冗余配置。

在控制域中执行 zpool status 命令检查镜像配置状态。

下面的示例介绍如何配置控制域中系统卷的 ZFS 镜像功能。

```
# zpool status rpool
pool: rpool
state: ONLINE
  scan: resilvered 29.1M in 0h0m with 0 errors on Thu Jan 23 17:27:59 2014
config:
  NAME                                STATE      READ  WRITE CKSUM
  rpool                               ONLINE    0     0     0
    c2t50000393E802CCE2d0s0          ONLINE    0     0     0
errors: No known data errors
```

执行 `zpool attach` 命令，以便将磁盘整合到镜像配置中。

```
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting.
#
```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。

使用 `zpool status` 命令确认同步处理 (resilver) 是否完成。

下面显示同步处理过程中显示的示例。

```
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function in a degraded state.
action: Wait for the resilver to complete.
        Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Jan 27 15:55:47 2014
      21.1G scanned out of 70.6G at 120M/s, 0h7m to go
      21.0G resilvered, 29.84% done
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                                DEGRADED   0      0      0
        mirror-0                          DEGRADED   0      0      0
          c2t50000393E802CCE2d0s0         ONLINE     0      0      0
          c3t50000393A803B13Ed0s0         DEGRADED   0      0      0 (resilvering)
errors: No known data errors
```

同步处理完成后，将显示如下画面：

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                                ONLINE     0      0      0
        mirror-0                          ONLINE     0      0      0
          c2t50000393E802CCE2d0s0         ONLINE     0      0      0
          c3t50000393A803B13Ed0s0         ONLINE     0      0      0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则创建冗余配置或恢复使用那些设备。有关如何创建冗余配置或恢复设备使用的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

A.3 对于从 XCP 2220 或更高版本的新安装，逻辑域配置有可用硬件资源

本节介绍如何创建一个具有两个 SPARC M10-4S，即两个系统板 (PSB<BB>)，在物理分区动态重新配置过程中具有移动资源所需的可用硬件资源（CPU 内核和内存）的配置。也就是说，本节介绍如何创建一个配置，对于单一的系统板被分配给逻辑域而言有足够的硬件资源的示例。也介绍了如何通过物理分区动态重新配置执行 SPARC M10-4S 的活动状态更换的示例。

在这种情况下，您可以在活动状态更换主板而不减少分配给逻辑域的 CPU 内核和内存。本介绍也适用于 SPARC M12。

A.3.1 配置示例

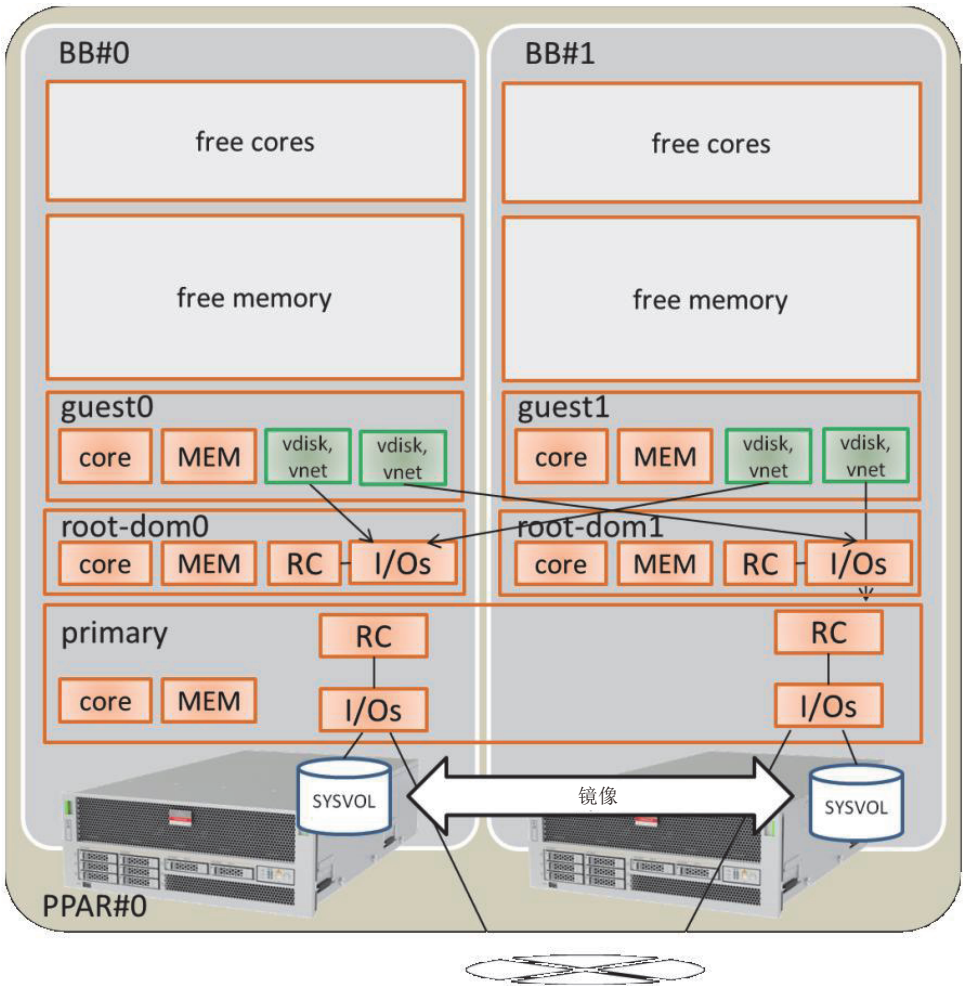
本节介绍由有相同编号的 CPU 和内存容量的两个 SPARC M10-4S 即两个系统板 (PSB<BB>)（2BB 配置）组成逻辑域的配置示例。

如需在物理分区中启用物理分区动态重新配置，I/O 配置必须满足以下条件。

- 控制域的系统卷磁盘和网络接口必须通过连接各 SPARC M10-4S 的 I/O 进行冗余。这样操作的目的是：当一个 SPARC M10-4S 被移除时，使用剩余的 SPARC M10-4S 的 I/O 可继续操作。
- 有必要为各个 SPARC M10-4S 设置根域，并且将该根域用作专用域以提供虚拟 I/O 给来宾域。原因如下：如果根域正在用于任务，在 SPARC M10-4S 断开时，根域必须被停止，因此任务将停止。
- 任务必须在来宾域中被操作。必须将各个根域的虚拟 I/O 服务分配给此来宾域的虚拟 I/O 设备 (vdisk, vnet)，以便使那些虚拟 I/O 设备冗余。原因如下：即使在一侧断开 SPARC M10-4S，可以通过使用剩余 SPARC M10-4S 上的根域的虚拟 I/O 服务继续 I/O 访问。通过这种方式，断开 SPARC M10-4S 将不影响来宾域，可继续操作任务。

下面是满足上述条件的配置的示意图。而且，简化了各个逻辑域的 I/O 配置。

图 A-3 操作连续性（具有可用资源）的 2BB 配置示例



下表示例将说明在 2BB 配置中，需要分配给控制域、根域和来宾域的 CPU 内核、内存和 I/O 配置。
必须将分配给各逻辑域的 CPU 内核总数和内存总量进行配置以适用于当释放其中一个 SPARC M10-4S 时，单个 SPARC M10-4S 可将其代替。

表 A-8 具有可用硬件资源的 2BB 配置示例

逻辑域名称	primary	root-dom0	root-dom1	guest0	guest1	可用空间	总数
CPU 内核	8	12	12	16	16	64	128 (激活: 64)
内存	14 GB (*1)	24 GB	24 GB	32 GB	32 GB	128 GB	256 GB

表 A-8 具有可用硬件资源的 2BB 配置示例 (续)

逻辑域名称	primary	root-dom0	root-dom1	guest0	guest1	可用空间	总数
物理 I/O (根合成体分配)	On-board #0 (PCIE0, PCIE4) On-board #1 (PCIE8, PCIE12)	PCIE1, PCIE2, PCIE3, PCIE5, PCIE6, PCIE7	PCIE9, PCIE10, PCIE11, PCIE13, PCIE14, PCIE15	-	-	-	-
虚拟 I/O 设备	-	vds0 vsw0, vsw1	vds1 vsw10, vsw11	vdisk0, vdisk10 vnet0, vnet10	vdisk1, vdisk11 vnet1, vnet11	-	-

*1 在上述配置示例中，需要分配给控制域的内存容量被计算为分配给其他逻辑域后内存的余数。
当获得通过从单个 SPARC M10-4S 的物理内存的容量 (128 GB) 减去分配给除控制域之外的域的容量所获得的剩余容量时，它是16 GB。
128 GB - 32 GB - 32 GB - 24 GB - 24 GB = 16 GB
实际分配给控制域的最大容量可以达到 14 GB，即从上面的结果中减去分配给单个 SPARC M10-4S 中虚拟机监控程序的最大内存容量 (2 GB) 所获得的容量。
16 GB - 2 GB = 14 GB
此外，当 Oracle VM Server for SPARC 的版本早于 3.2 时，无论它是什么，都查看倍数（内核数量 x 256 MB）。这是执行物理分区动态重新配置的条件。由于可以证明满足了此条件，内存容量成为 14 GB。
14 GB / (8 内核 x 256 MB) = 7
可以除尽。因此可以肯定是 14 GB。

注 - 您应该通过注意以下两点来设计分配给逻辑域的内存容量。有关详情，请参见 "2.4.1 逻辑域配置注意事项"。

- 分配给逻辑域的内存容量小于物理内存容量。
- 当 Oracle VM Server for SPARC 的版本早于 3.2 时，使用（内核的数量 x 256 MB）的倍数来启用物理分区动态重新配置。

A.3.2 物理分区配置步骤示例

1. 登录主 **XSCF**。
执行 **showbbstatus** 命令来检查您已经登录的 XSCF 是否是主 XSCF。
如果您已经登录到备用 XSCF，注销然后再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```
2. 设置内存的镜像模式。
 - a. 执行 **showfru** 命令来检查 **SPARC M10-4S** 系统板 (**PSB<BB>**) 上内存的镜像模式。
下面是显示物理系统板 (PSB 00-0) 下设备的设置信息示例。

```
XSCF> showfru sb 00-0
Device Location Memory Mirror Mode
sb 00-0
  cpu 00-0-0 no
  cpu 00-0-1 no
  cpu 00-0-2 no
  cpu 00-0-3 no
```

- b. 若需使用内存镜像模式，请执行 **setupfru** 命令来设置该模式。

如果您未正在使用内存镜像模式，将不需要此步骤。

有关内存镜像模式的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的“14.1 配置内存镜像”。

下面的示例将 PSB 00-0 下的所有 CPU 都设置为内存镜像模式。

```
XSCF> setupfru -m y sb 00-0
```

执行 **showfru** 命令检查内存镜像模式的设置。

```
XSCF> showfru sb 00-0
Device Location Memory Mirror Mode
sb 00-0
  cpu 00-0-0 yes
  cpu 00-0-1 yes
  cpu 00-0-2 yes
  cpu 00-0-3 yes
```

3. 创建物理分区配置信息。

- a. 执行 **showpctl** 命令检查物理分区配置信息。

```
XSCF> showpctl -p 0
PPAR-ID LSB PSB Status
```

- b. 执行 **setpctl** 命令在物理分区配置信息中登录系统板。

执行 **setpctl** 命令在内置目标的物理分区配置信息中登录 SPARC M10-4S 系统板。

在下面的示例中，物理系统板 (PSBs) 00-0 和 01-0 映射到物理分区 0 的逻辑系统板 (LSBs) 00 和 01。

```
XSCF> setpctl -p 0 -a 00=00-0 01=01-0
```

- c. 执行 **showpctl** 命令检查物理分区配置信息。

检查物理分区配置信息。

XSCF> showpcl -p 0					
PPAR-ID	LSB	PSB	Status		
00			Running		
	00	00-0			
	01	01-0			

指定 -v 选项显示配置策略、IO 无效选项 (No-IO) 和内存无效选项 (No-Mem) 的详细信息。

XSCF> showpcl -v -p 0						
PPAR-ID	LSB	PSB	Status	No-Mem	No-IO	Cfg-policy
00			Running			
						System
	00	00-0		False	False	
	01	01-0		False	False	

使用 setpcl 命令变更配置策略、IO 无效选项 (No-IO) 和内存无效选项 (No-Mem) 的设置。

有关 setpcl 命令的详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/ SPARC M10 XSCF Reference Manual》。

4. 将系统板 (**PSB<BB>**) 分配给物理分区。
 - a. 执行 **showboards-a** 命令检查 **PSB** 状态。

执行 showboards -a 命令检查各 PSB 的状态是否是 "SP"（系统板池）。

XSCF> showboards -a							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	SP	Available	n	n	n	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal

- b. 执行 **addboard -c assign** 命令分配 **PSB**。

XSCF> addboard -c assign -p 0 00-0 01-0	
--	--

- c. 执行 **showboards-p** 命令检查 **PSB** 状态。

执行 showboards -p 命令检查分配给物理分区的各 PSB 的状态。

由于各 PSB 已经被正常分配给物理分区 0，此示例检查各 PSB 的 [Assignment] 字段是否成为 "Assigned"。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	n	n	n	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

5. 登录 **CPU** 激活密钥以便分配 **CPU** 内核资源。

- a. 执行 **showcodactivation** 命令，检查 CPU 激活密钥上的信息。
- 执行 **showcodactivation** 命令，检查物理分区是否包含可分配的 CPU 激活密钥。
- 如果仅显示标题，则 CPU 激活密钥未在 XSCF 中登录。

```
XSCF> showcodactivation
Index Description Count
-----
```

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。

- b. 执行 **addcodactivation** 命令以登录 CPU 激活密钥。
- 有关如何注册 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "5.3 添加 CPU 内核资源"。

```
XSCF> addcodactivation "Product: SPARC M10-4S
SequenceNumber:10005
Cpu: noExpiration 2
Text-Signature-SHA256-RSA2048:
PSSrElBrse/r69AVSVFd38sT6AZm2bxUDdPQHKbtxgvZPsrtYguqiNUieB+mTDC
:
:
b1GCKFx1RH27FdVHiB2H0A=="
AboveKeywillbeadded,Continue?[y|n]:y
```

- c. 执行 **showcodactivation** 命令，检查 CPU 激活密钥上的信息。
- 执行 **showcodactivation** 命令，检查物理分区是否包含可分配的 CPU 激活密钥。

```
XSCF> showcodactivation
Index Description Count
-----
0 PROC 2
1 PROC 2
2 PROC 2
3 PROC 2

--- 省略 ---

30 PROC 2
31 PROC 2
```

- d. 执行 **setcod** 命令分配 CPU 内核资源。
- 执行 **setcod** 命令将 CPU 内核资源分配给物理分区。
- 以下是将 64 个 CPU 内核资源分配到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c set 64
PROC Permits assigned for PPAR 0 : 0 -> 64

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 -XCP 2250 和更早版本的 XSCF 固件不支持 **-c add**、**-c delete** 和 **-c set** 选项。按如下所示指定 **setcod** 命令的选项可交互式添加和删除 CPU 内核资源。

```
XSCF> setcod -s cpu
```

执行 **showcod** 命令检查分配给物理分区的 CPU 内核资源的信息。

下面的示例，使用刚刚执行的 **setcod** 命令，证实 64 个 CPU 内核资源被分配给物理分区 0。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 64
```

6. 执行 **resetdateoffset** 命令以 XSCF 管理的时间重置差异。

执行 **resetdateoffset** 命令来重置 XSCF 管理的时间和物理分区管理的时间之间的差异。

```
XSCF> resetdateoffset -p 0
```

7. 执行 **showpparmode** 命令来检查诊断消息的详细级别的设置以及 PPAR DR 模式的设置。

执行 **showpparmode** 命令来检查诊断信息的详细级别 (Message Level) 是否是 "normal" (标准) 以及 PPAR DR 模式的 Next 是否设置为 "on" (启用)。

```
XSCF> showpparmode -p 0
Host-ID                :9007002b
Diagnostic Level        :min
Message Level           :normal
Alive Check             :on
Watchdog Reaction       :reset
Break Signal            :on
Autoboot (Guest Domain) :on
Elastic Mode            :off
IOreconfigure           :false
PPAR DR (Current)       :-
PPAR DR (Next)          :on
```

如果诊断信息的详细级别是除 "normal" 之外的级别, 执行 **setpparmode** 命令可将其设置为 "normal"。

有关 **setpparmode** 命令的详情, 请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

```
XSCF> setpparmode -p 0 -m message=normal
```

如果 PPAR DR 模式设置为 "off"（禁用），执行 setpparmode 命令可将其设置为 "on"。

```
XSCF> setpparmode -p 0 -m ppar_dr=on
```

8. 执行 **poweron** 命令来开启物理分区的电源。

```
XSCF> poweron -p 0
```

9. 执行 **console** 命令将控制台连接到物理分区。

```
XSCF> console -p 0
```

10. 安装 **Oracle Solaris** 和 **Oracle VM Server for SPARC**。

在控制域上安装 Oracle Solaris 和 Oracle VM Server for SPARC。

有关物理分区动态重新配置所要求的 Oracle Solaris 的版本和条件的详情，请参见表 1-3 和表 1-4。

有关安装的详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com/>) 上的以下文档。

- Oracle Solaris 10

《Oracle Solaris 10 1/13 Installation Guide》

- Oracle Solaris 11

《Installing Oracle Solaris 11.x Systems》

- Oracle VM Server for SPARC

《Oracle VM Server for SPARC Administration Guide》中的 "Installing and Enabling Software"

11. 配置逻辑域。

此项介绍如何配置在 "图 A-3 操作连续性（具有可用资源）的 2BB 配置示例" 中定义的逻辑域。

- a. 减少控制域资源的数量。

对于出厂默认配置而言，所有 CPU 内核、内存和 PCIe 根合成体都被分配给控制域 (primary)。如需将这些资源分配给其他逻辑域，首先从控制域删除某些资源并配置控制台服务。

下面的示例显示了命令的执行。

执行 **ldm start-reconf** 命令切换到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
```

will also take effect.

使用 `ldm remove-io` 命令移除根合成体。

下面的示例部分地说明了按照配置示例移除 PCIE1、PCIE2、PCIE3、PCIE5、PCIE6、PCIE7、PCIE9、PCIE10、PCIE11、PCIE13、PCIE14 和 PCIE15 的命令。

```
# ldm remove-io PCIE1 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
....
# ldm remove-io PCIE15 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

通过使用 `ldm set-core` 和 `ldm set-memory` 命令指定小于初始容量的容量来减少分配给控制域的 CPU 内核数量和内存容量。

下面举例说明按照配置示例，将控制域的 CPU 内核数量设置为 8，以及将内存容量设置为 14 GB。

```
# ldm set-core 8 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm set-memory 14G primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

注 - 建议首先使用 `ldm set-core` 命令配置 CPU 内核，然后使用 `ldm set-memory` 命令配置内存。这有利于集合内存的连续区域的分配。这也易于可进行物理分区动态重新配置的 CPU 内核配置条件的满足，以及内存配置条件的满足。有关 CPU 内核和内存的配置条件的详情，请参见“2.5.2 动态重新配置的系统操作注意事项”中的“CPU 内核和内存的配置”。

使用 `ldm add-vconscn` 命令创建称为 `vcc0` 的虚拟控制台终端集中器服务，然后使用 `svcadm` 命令启动虚拟网络终端服务器 (`vntsd`) 的守护程序。通过此 `vcc0` 配置连接到各逻辑域的控制台。

```
# ldm add-vconscn port-range=5000-5200 vcc0 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
# svcadm enable vntsd
```

保存配置信息，然后重新启动 Oracle Solaris。

下面的示例检查使用 `ldm list-spconfig` 命令保存的配置信息，然后使用 `ldm add-spconfig` 命令以名称为 `ldm-set1` 保存该配置。然后再次使用 `ldm list-spconfig` 命令检查配置是否已经保存，最后举例说明重新启动 Oracle Solaris。

```
# ldm list-spconfig
factory-default [current]
# ldm add-spconfig ldm-set1
# ldm list-spconfig
factory-default
ldm-set1 [current]
# shutdown -i6 -g0 -y
```

b. 创建控制域系统卷的冗余配置。

创建控制域系统卷的冗余配置。此项目介绍使用 ZFS 创建镜像配置的命令示例。有关详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com>) 上的以下文档。

- Oracle Solaris 10 的情形

《Oracle Solaris ZFS Administration Guide》上的 "How to Create a Mirrored Root Pool (Post Installation)"

- Oracle Solaris 11 的情形

《Oracle Solaris 11.1 Administration: ZFS File Systems》中的 "How to Configure a Mirrored Root Pool (SPARC or x86/VTOC)"

如需使用其他冗余配置软件，请参见该软件的手册。

执行 `zpool status` 命令检查根池的状态。

下面的示例表明已经将 `c2t50000393E802CCE2d0s0` 分配给默认根池 (`rpool`)。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: none requested
config:
    NAME                                STATE      READ WRITE CKSUM
    rpool                                ONLINE    0     0     0
        c2t50000393E802CCE2d0s0        ONLINE    0     0     0
errors: No known data errors
```

执行 `format` 命令检查可添加哪个磁盘。

下面的示例表明 `c3t50000393A803B13Ed0` 作为其他磁盘而存在。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
    0. c2t50000393E802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8000/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
        /dev/chassis/FUJITSU-BBEXP.500000e0e06d027f/0123_HDD00/disk
    1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8800/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
        /dev/chassis/FUJITSU-BBEXP.500000e0e06d243f/022U_HDD01/disk
Specify disk (enter its number): ^C
```

执行 `zpool attach` 命令将第二个磁盘添加到 `rpool`，以便提供镜像配置。

下面的示例使用 `zpool attach` 命令添加 `c3t50000393A803B13Ed0s0`，然后使用 `zpool status` 命令检查同步处理 (resilver) 的状态。通过参照状态和动作，您可以确定正在进行同步处理。定期执行 `zpool status` 命令来检查同步处理的状态，直到处理结束。

```
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function in a degraded state.
action: Wait for the resilver to complete.
        Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Wed Jan 29 21:35:39 2014
      3.93G scanned out of 70.6G at 71.9M/s, 0h15m to go
      3.90G resilvered, 5.56% done
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                                DEGRADED   0      0      0
        mirror-0                            DEGRADED   0      0      0
          c2t50000393E802CCE2d0s0          ONLINE    0      0      0
          c3t50000393A803B13Ed0s0          DEGRADED   0      0      0 (resilvering)
```

当完成同步处理时，`[state]` 将被设置为 "ONLINE"，如下所示。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h10m with 0 errors on Wed Jan 29 21:45:42 2014
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                                ONLINE    0      0      0
        mirror-0                            ONLINE    0      0      0
          c2t50000393E802CCE2d0s0          ONLINE    0      0      0
          c3t50000393A803B13Ed0s0          ONLINE    0      0      0
errors: No known data errors
```

c. 配置根域。

本项目介绍配置根域的步骤。

执行 `ldm add-domain` 命令添加名为 `root-dom0` 的逻辑域。

```
# ldm add-domain root-dom0
```

执行 `ldm set-variable` 命令将被设计为自动启动 Oracle Solaris 的 OpenBoot PROM 环境变量 `"auto-boot?"` 变更为 `"false"`（禁用）。默认状态下，此设置为 `"true"`（启用）。因此，当未安装 Oracle Solaris 时，OpenBoot PROM 尝试自动启动 Oracle Solaris。安装 Oracle Solaris 前，变更此设置为禁用将有利于将要执行的操作。

```
# ldm set-variable auto-boot\?=false root-dom0
```

首先，使用 `ldm set-core` 命令分配 CPU 内核，然后使用 `ldm set-memory` 命令分配内存。

下面的示例根据配置示例，使用 `ldm set-core` 命令分配 12 个 CPU 内核，使用 `ldm set-memory` 命令分配 24 GB 内存。

```
# ldm set-core 12 root-dom0
# ldm set-memory 24G root-dom0
```

注 - 建议首先使用 `ldm set-core` 命令配置 CPU 内核，然后使用 `ldm set-memory` 命令配置内存。这有利于集合内存的连续区域的分配。这也易于可进行物理分区动态重新配置的 CPU 内核配置条件的满足，以及内存配置条件的满足。有关 CPU 内核和内存的配置条件的详情，请参见 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中的 ["CPU 内核和内存的配置"](#)。

执行 `ldm set-vconsole` 命令分配虚拟控制台 (vcons)。

下面的示例执行 `ldm set-vconsole` 命令将控制域中虚拟控制台终端集中器的服务 (vcc0) 的端口号 5000 分配给虚拟控制台。

```
# ldm set-vconsole service=vcc0 port=5000 root-dom0
```

下面的示例执行 `ldm list-io -l` 命令显示 PCI 分配状态。

NAME 以 `"/BB0"` 开始所在行和 [TYPE] 栏有 `"PCIE"` 是指 SPARC M10-4S (BB0) 上的 PCIe 端点。[DOMAIN] 栏所在行为空表示未分配的 PCIe 端点，有关的根合成体显示在 [BUS] 栏。

因此，您可以快速了解到 PCIE1、PCIE2、PCIE3、PCIE5、PCIE6 和 PCIE7 是 BB0 上未分配的根合成体。

```
# ldm list-io -l
NAME                                     TYPE    BUS      DOMAIN    STATUS
----
(省略)
```


/BB0/CMUL/NET0 [pci@8000/pci@4/pci@0/pci@9] network@0 network@0,1	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA [pci@8000/pci@4/pci@0/pci@0] scsi@0/iport@f/disk@w50000393e802cce2,0 scsi@0/iport@f/disk@w50000393d8285226,0 scsi@0/iport@f/smp@w500000e0e06d027f scsi@0/iport@f/enclosure@w500000e0e06d027d,0 scsi@0/iport@v0	PCIE	PCIE0	primary	OCC
/BB0/PCI0 [pci@8100/pci@4/pci@0/pci@0]	PCIE	PCIE1		EMP
/BB0/PCI3 [pci@8200/pci@4/pci@0/pci@0]	PCIE	PCIE2		EMP
/BB0/PCI4 [pci@8200/pci@4/pci@0/pci@8]	PCIE	PCIE2		EMP
/BB0/PCI7 [pci@8300/pci@4/pci@0/pci@0]	PCIE	PCIE3		EMP
/BB0/PCI8 [pci@8300/pci@4/pci@0/pci@8]	PCIE	PCIE3		EMP
/BB0/CMUL/NET2 [pci@8400/pci@4/pci@0/pci@a] network@0 network@0,1	PCIE	PCIE4	primary	OCC
/BB0/PCI1 [pci@8500/pci@4/pci@0/pci@8]	PCIE	PCIE5		EMP
/BB0/PCI2 [pci@8500/pci@4/pci@0/pci@9]	PCIE	PCIE5		EMP
/BB0/PCI5 [pci@8600/pci@4/pci@0/pci@9]	PCIE	PCIE6		EMP
/BB0/PCI6 [pci@8600/pci@4/pci@0/pci@11]	PCIE	PCIE6		EMP
/BB0/PCI9 [pci@8700/pci@4/pci@0/pci@9]	PCIE	PCIE7		EMP
/BB0/PCI10 [pci@8700/pci@4/pci@0/pci@11] (省略)	PCIE	PCIE7		EMP

请参见显示在上述结果中的设备路径（显示为 [pci@...] 的字符串）和《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "A.6 SPARC M10-4S 设备路径"，确认哪一个根合成体需要分配给根域。

在配置示例中，上述检查的 BB0 上所有未分配的根合成体 (PCIE1、PCIE2、PCIE3、PCIE5、PCIE6 和 PCIE7) 将被分配。因此，执行 `ldm add-io` 命令将其分配给 `root-dom0`。

下面的示例显示命令的执行。

```
# ldm add-io PCIE1 root-dom0
# ldm add-io PCIE2 root-dom0
# ldm add-io PCIE3 root-dom0
# ldm add-io PCIE5 root-dom0
# ldm add-io PCIE6 root-dom0
```

```
# ldm add-io PCIE7 root-dom0
```

使用 ldm bind-domain 命令将根域设置为 bound 状态，然后执行 ldm list-io 命令检查根合成体是否已经被分配。

下面的示例使用 ldm bind-domain 命令检查 root-dom0 是否绑定，使用 ldm list-io 命令检查根合成体是否已经被分配。

[TYPE] 栏是 "BUS" 并且 [DOMAIN] 栏是 "root-dom0" 的行表示分配给 root-dom0 的根合成体。该行中的 BUS 是已分配的根合成体的名称。

在下面的示例中，您可以确认已经将 PCIE1、PCIE2、PCIE3、PCIE5、PCIE6 和 PCIE7 分配给 root-dom0。

```
# ldm bind-domain root-dom0
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8	primary	IOV

(省略)

根据步骤 11. c，配置其他 SPARC M10-4S 的根域（示例中为 root-dom1）。

d. 在根域中安装 **Oracle Solaris**。

本项目介绍在根域中安装 Oracle Solaris 的步骤。

执行 ldm start-domain 命令启动根域 root-dom0。

```
# ldm start-domain root-dom0
LDom root-dom0 started
```

执行 ldm set-variable 命令将被设计为自动启动 Oracle Solaris 的 OpenBoot PROM 环境变量 "auto-boot?" 变更为 "true"（启用）。安装 Oracle Solaris 后，执行 ldm start-domain 命令也可启动 Oracle Solaris。

```
# ldm set-variable auto-boot\?=true root-dom0
```

执行 telnet 命令连接到根域的控制台。

下面的示例通过执行 ldm list-domain 命令确认 root-dom0 的端口号是 5000。通过使用 telnet 命令连接到 localhost 的端口号 5000 也可以确认 root-dom0 在 OpenBoot PROM (OBP) 状态停止。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   UART    16    14G     0.0%  7h 7m
root-dom0     active    -t----   5000    24    24G     0.0%  20s
root-dom1     bound     ------  5001    24    24G
# telnet localhost 5000
....
{0} ok
```

在根域中安装 Oracle Solaris。

有关物理分区动态重新配置所要求的 Oracle Solaris 的版本和条件的详情，请参见表 1-3 和表 1-4。

下面的示例通过网络执行命令启动 Oracle Solaris 11 安装。

```
{0} ok boot net:dhcp
....
```

有关安装的详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com/>) 上的以下文档。

- Oracle Solaris 10

《Oracle Solaris 10 1/13 Installation Guide》

- Oracle Solaris 11

《Installing Oracle Solaris 11.x Systems》

根据步骤 11. d，采用相同的方式，安装其他根域（示例中为 root-dom1）。

e. 将虚拟 I/O 服务分配给根域。

此项目介绍将虚拟 I/O 服务设置给根域的步骤示例。

此示例显示将根域的整个物理磁盘借为虚拟磁盘服务 (vds)。

此示例显示将以太网卡端口分配给虚拟交换网络设备 (vsw)。

有关虚拟 I/O 服务的详情，请参见 Oracle 公司出版的《Oracle VM Server for SPARC Administration Guide》。

以 root（根）权限登录根域。

```
root-dom0 console login: root
Password:
....
```

如需指定需要分配给虚拟磁盘服务 (vds) 的后端设备（物理磁盘），请执行 format 命令显示磁盘和其磁盘路径。

在格式 cXtXXXXXXXXXXXXXXXXXXdX 中的字符串表示磁盘，以 "/pci@" 开始的字符串表示设备路径。

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
```

```

0. c2t50000393A802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
   /pci@8100/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
1. c3t50000393D8285226d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
   /pci@8200/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393d8285226,0
2. c4t50000393A804B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
   /pci@3200/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
....

```

请参见显示在上述结果中的设备路径和《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "A.6 SPARC M10-4S 设备路径"，并检查磁盘的物理位置。然后，确定需要分配给虚拟磁盘服务的后端磁盘 (cXtXXXXXXXXXXXXXXXXdX)。

配置示例分配了两个磁盘 (c3t50000393D8285226d0 和 c4t50000393A804B13Ed0) 作为来宾域 (guest0, guest1) 的虚拟磁盘的后端。

接下来, 若需指定需要分配给虚拟交换服务 (vsw) 的网络接口, 请执行 `dladm show-phys` 命令显示与 PCIe 插槽的位置 (LOC) 相关联的网络接口名 (LINK)。从显示的结果, 检查网络接口的位置并确认需要分配给虚拟交换服务的网络接口。

配置示例将两个网络接口 (net1 和 net2) 分配给与各来宾域的虚拟网络相关联的虚拟交换机。

```

# dladm show-phys -L
LINK          DEVICE          LOC
net0          igb0            BB#0-PCI#0
net1          igb1            BB#0-PCI#1
net2          igb2            BB#0-PCI#2
net3          igb3            BB#0-PCI#3
...

```

执行 `ldm add-vdiskserver` 命令将虚拟磁盘服务添加到根域。

下面的示例使用 `ldm add-vdiskserver` 命令将虚拟磁盘服务 (vds0) 添加到根域 (root-dom0)。

```

# ldm add-vdiskserver vds0 root-dom0

```

执行 `ldm add-vdiskserverdevice` 命令从根域导出虚拟磁盘的后端。

下面的示例执行 `ldm add-vdiskserverdevice` 命令将两个后端磁盘分配给 vds0 以便将其导出。

```

# ldm add-vdiskserver vds0 root-dom0
# ldm add-vdiskserverdevice /dev/dsk/c3t50000393D8285226d0s2 vol0@vds0
# ldm add-vdiskserverdevice /dev/dsk/c4t50000393A804B13Ed0s2 vol1@vds0

```

执行 `ldm add-vswitch` 命令将虚拟交换机添加到根域。

下面的示例添加了 vsw0 和 vsw1，以便添加各来宾域的虚拟交换机，并将不同的物理网络接口 (net1, net2) 分配给各来宾域。

```
# ldm add-vswitch net-dev=net1 vsw0 root-dom0
# ldm add-vswitch net-dev=net2 vsw1 root-dom0
```

同样，将虚拟 I/O 服务添加到分配 BB1 侧上的 SPARC M10-4S 的 I/O 的根域 (root-dom1)。

分配以下虚拟磁盘服务名和虚拟交换机名给 root-dom1 当作示例。有关步骤的详情，请参见 11.e。

虚拟磁盘服务名：vds1（分配 vol10 和 vol11 作为两个后端卷的名称。）

虚拟交换机名：vsw10, vsw11

f. 配置来宾域。

本项目介绍配置来宾域的步骤。

执行 ldm add-domain 命令添加名为 guest0 的逻辑域。

```
# ldm add-domain guest0
```

执行 ldm set-variable 命令将被设计为自动启动 Oracle Solaris 的 OpenBoot PROM 环境变量 "auto-boot?" 变更为 "false"（禁用）。默认状态下，此设置为 "true"（启用）。因此，当未安装 Oracle Solaris 时，OpenBoot PROM 尝试自动启动 Oracle Solaris。安装 Oracle Solaris 前，变更此设置为禁用将有利于将要执行的操作。

```
# ldm set-variable auto-boot\?=false guest0
```

首先，使用 ldm set-core 命令分配 CPU 内核，然后使用 ldm set-memory 命令分配内存。

下面的示例根据配置示例，使用 ldm set-core 命令分配 16 个 CPU 内核，使用 ldm set-memory 命令分配 32 GB 内存。

```
# ldm set-core 16 root-dom0
# ldm set-memory 32G root-dom0
```

注 - 建议首先使用 ldm set-core 命令配置 CPU 内核，然后使用 ldm set-memory 命令配置内存。这有利于集合内存的连续区域的分配。这也易于可进行物理分区动态重新配置的 CPU 内核配置条件的满足，以及内存配置条件的满足。有关 CPU 内核和内存的配置条件的详情，请参见 ["2.5.2 动态重新配置的系统操作注意事项"](#) 中的 ["CPU 内核和内存的配置"](#)。

执行 ldm set-vconsole 命令分配虚拟控制台 (vcons)。

下面的示例执行 ldm set-vconsole 命令将控制域中虚拟控制台终端集中器的服务 (vcc0) 的端口号 5100 分配给虚拟控制台。

```
# ldm set-vconsole service=vcc0 port=5100 guest0
```

执行 `ldm add-vdisk` 命令分配虚拟磁盘 (vdisk)。

下面的示例按照配置示例分配了两个虚拟磁盘 (vdisk0, vdisk10)。各虚拟磁盘的后端指定添加到两个根域 (root-dom0, root-dom1) 的后端 (vds0 vol0, vds1 vol10)。

```
# ldm add-vdisk vdisk0 vol10@vds0 guest0
# ldm add-vdisk vdisk10 vol110@vds1 guest0
```

执行 `ldm add-vnet` 命令分配虚拟网络设备 (vnet)。

下面的示例按照配置示例分配了两个虚拟设备 (vnet0, vnet10)。连接到各虚拟网络设备的虚拟交换机指定添加到两个根域 (root-dom0, root-dom1) 的虚拟交换机 (vsw0, vsw10)。

```
# ldm add-vnet vnet0 vsw0 guest0
# ldm add-vnet vnet10 vsw10 guest0
```

同样，按照步骤 11.f 所述配置其他来宾域（本例中为 `guest1`）。

按照配置示例，将要分配以下设备：

虚拟磁盘： vdisk1（vol1@vds0 适用于后端），vdisk11（vol11@vds1 适用于后端）

虚拟网络： vnet1（vsw1 适用于虚拟交换机），vnet11（vsw11 适用于虚拟交换机）

注 - 如果分配多个虚拟磁盘或虚拟网络设备，请从 `ldm list -l` 命令的执行结果记录分配给虚拟设备的 ID 值。虚拟设备被动态删除后，如果再次添加虚拟设备并指定 ID，则需要该ID，以便虚拟设备路径保持不变。

g. 在来宾域中安装 **Oracle Solaris**。

本项目介绍在来宾域中安装 Oracle Solaris 的步骤。

执行 `ldm bind-domain` 命令绑定来宾域，然后通过执行 `start-domain` 命令将其启动。

下面的示例显示启动 `guest0` 的命令的执行。

```
# ldm bind-domain guest0
# ldm start-domain guest0
LDom guest0 started
```

执行 `ldm set-variable` 命令将被设计为自动启动 Oracle Solaris 的 OpenBoot PROM 环境变量 "auto-boot?" 变更为 "true"（启用）。安装 Oracle Solaris 后，执行 `ldm start-domain` 命令也可启动 Oracle Solaris。

```
# ldm set-variable auto-boot\?=true guest0
```

执行 `telnet` 命令连接到来宾域的控制台。

下面的示例通过执行 `ldm list-domain` 命令确认 `guest0` 控制台的端口号是 "5100"。通过使用 `telnet` 命令连接到 `localhost` 的端口号 "5100" 也可以确认 `guest0` 在 OpenBoot PROM 状态停止。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    16      14G       0.0%    8h 7m
guest0        active    -t----   5100    32      32G       0.0%    20s
root-dom0     active    -n--v-   5000    24      24G       0.0%    43s
root-dom1     active    -n--v-   5001    24      24G       0.0%    20s
guest1        inactive  ------   32      32G
# telnet localhost 5100
....
{0} ok
```

有关物理分区动态重新配置所要求的 Oracle Solaris 的版本和条件的详情，请参见表 1-3 和表 1-4。

下面的示例通过网络执行命令启动 Oracle Solaris 11 安装。

```
{0} ok boot net:dhcp
....
```

有关安装的详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com/>) 上的以下文档。

- Oracle Solaris 10

《Oracle Solaris 10 1/13 Installation Guide》

- Oracle Solaris 11

《Installing Oracle Solaris 11.x Systems》

根据步骤 11. g，采用相同的方式，安装其他来宾域（示例中为 `guest1`）。

h. 创建来宾域的虚拟 I/O 的冗余配置。

下面介绍使用 IPMP 创建分配给来宾域 `guest0` 的两个虚拟网络接口 (vnets) 的冗余配置步骤示例。有关其他冗余配置的步骤的详情，请参见各自冗余配置软件的文档。

登录来宾域 `guest0`。

在下面的示例中，使用 `ldm list-domain` 命令检查 `guest0` 的控制台的端口号，使用 `telnet` 命令连接到端口号 "5100"。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    16      14G       0.0%    8h 7m
guest0        active    -n----   5100    32      32G       0.0%    20s
guest1        active    -n----   5101    32      32G       0.0%    19s
root-dom0     active    -n--v-   5000    24      24G       0.0%    43s
root-dom1     active    -n--v-   5001    24      24G       0.0%    20s
# telnet localhost 5100
```

```
....
guest0 console login: root
Password:
...
guest0#
```

执行 `dladm` 命令确认虚拟网络设备可见。

在下面的示例中,可把虚拟网络设备称为网络接口 `net0` 和 `net1` 。

```
guest0# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	0	unknown	vnet0
net1	Ethernet	up	0	unknown	vnet1

执行 `ipadm show-if` 命令确认 `net0` 和 `net1` 未显示。

```
guest0# ipadm show-if
```

IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--

执行 `ipadm create-ip` 命令创建 IP 接口 `net0` 和 `net1`，然后使用 `ipadm show-if` 命令确认已经将其正常创建。

```
guest0# ipadm create-ip net0
guest0# ipadm create-ip net1
guest0# ipadm show-if
```

IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--
net0	ip	down	no	--
net1	ip	down	no	--

执行 `ipadm create-ipmp` 命令创建 IPMP 接口 `ipmp0`，然后执行 `ipadm add-ipmp` 命令将 IP 接口 `net0` 和 `net4` 添加到 IPMP 组。

```
guest0# ipadm create-ipmp ipmp0
guest0# ipadm add-ipmp -i net0 -i net1 ipmp0
```

执行 `ipadm create-addr` 命令将 IP 地址分配给 IPMP 接口 `ipmp0`，然后使用 `ipadm show-addr` 命令检查此设置。在下面的示例中，分配了固定 IP 地址。

```
guest0# ipadm create-addr -T static -a local=xx.xx.xx.xx/24 ipmp0/v4
guest0# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
ipmp0/v4	static	ok	xx.xx.xx.xx/24
lo0/v6	static	ok	::1/128

执行 `ipadm set-ifprop` 命令设置备用接口，然后使用 `ipmpstat -i` 命令检查

IPMP 配置。

```
guest0# ipadm set-ifprop -p standby=on -m ip net1
guest0# ipmpstat -i
INTERFACE    ACTIVE    GROUP      FLAGS      LINK      PROBE      STATE
net1         no       ipmp0      is-----  up        disabled   ok
net0         yes      ipmp0      --mbM--    up        disabled   ok
```

对其他来宾域（本示例中为 guest1）执行相同的步骤。

i. 检查内存资源的分配状况，并分组分散的内存块。

首先，请参见 "A.2.2 物理分区配置步骤示例" 中的步骤 11.h 和 11.i，然后分组分散的内存块。

下面的示例显示根据配置示例，在内存分散的情况下，执行 ldm list-devices 命令的结果。

您可以确认每个逻辑域仅存在一个内存块，并且不存在许多 256 至 512 MB 的分散内存块。

```
# ldm list-devices -a memory
MEMORY
    PA                SIZE                BOUND
0x70000000000000    24G                root-dom1
0x70060000000000    8G
0x72000000000000    32G                guest0
0x74000000000000    32G                guest1
0x76000080000000    1272M              _sys_
0x76005000000000    24G                root-dom0
0x76065000000000    6912M
0x78000000000000    32G
0x7a000000000000    32G
0x7c000000000000    32G
0x7e000080000000    1272M              _sys_
0x7e005000000000    512M                _sys_
0x7e007000000000    256M                _sys_
0x7e008000000000    14G                primary
0x7e040000000000    16G
```

j.检查是否存在足够的可用内存块，并进行相应调整。

此步骤介绍释放 SPARC M10-4S，并进行调整后，如何检查逻辑域内存块是否可被移动至可用内存块。

i. 检查内存的连续区域（内存块）的使用状况。

执行 prtdiag 命令检查内存物理地址和 SPARC M10-4S 之间的映射。

```
# prtdiag
(省略)
===== Physical Memory Configuration =====
Segment Table:
-----
Base                Segment  Interleave  Bank        Contains
```

Address	Size	Factor	Size	Modules
0x7e0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUL/CMP0/MEM00A
0x7c0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUL/CMP1/MEM10A
0x7a0000000000 (省略)	32 GB	4	8 GB	/BB0/CMUU/CMP0/MEM00A
0x780000000000 (省略)	32 GB	4	8 GB	/BB0/CMUU/CMP1/MEM10A
0x760000000000 (省略)	32 GB	4	8 GB	/BB1/CMUL/CMP0/MEM00A
0x740000000000 (省略)	32 GB	4	8 GB	/BB1/CMUL/CMP1/MEM10A
0x720000000000 (省略)	32 GB	4	8 GB	/BB1/CMUU/CMP0/MEM00A
0x700000000000 (省略)	32 GB	4	8 GB	/BB1/CMUU/CMP1/MEM10A

本示例的结果按内存中物理地址的升序重新排列。下表列出了物理地址和 SPARC M10-4S 之间的对应关系。

表 A-9 物理地址和 SPARC M10-4S 之间对应关系的示例

基地址（物理地址）	SPARC M10-4S 的模块化架构配置
0x700000000000 和之后	BB1
0x720000000000 和之后	BB1
0x740000000000 和之后	BB1
0x760000000000 和之后	BB1
0x780000000000 和之后	BB0
0x7a0000000000 和之后	BB0
0x7c0000000000 和之后	BB0
0x7e0000000000 和之后	BB0

将步骤 11. i 中的结果与使用 prtdiag 命令所确认的物理地址匹配后，内存块的使用状况如下所示。

表 A-10 内存块使用状况的示例

SPARC M10-4S	物理地址	容量	逻辑域
BB1（更换的目标）	0x700000000000	24 GB	root-dom1
	0x700600000000	8 GB	未分配
	0x720000000000	32 GB	guest0
	0x740000000000	32 GB	guest1
	0x760050000000	24 GB	root-dom0
	0x760650000000	6,912 MB	未分配

表 A-10 内存块使用状况的示例 (续)

SPARC M10-4S	物理地址	容量	逻辑域
BB0	0x780000000000	32 GB	未分配
	0x7a0000000000	32 GB	未分配
	0x7c0000000000	32 GB	未分配
	0x7e0080000000	14 GB	primary
	0x7e0400000000	16 GB	未分配

ii.检查移动源内存块的容量和数量。

参照内存块使用状况的检查结果时，检查分配给需要更换的 SPARC M10-4S 的内存块（在下文中称为“源内存块”）。

在“表 A-10 内存块使用状况的示例”中，您可以看到分配给 BB1 侧逻辑域的内存块的数量是 32 GB × 2（分配给 guest0 和 guest1）和 24 GB × 1（root-dom0）。

注 - 分配需要释放的 SPARC M10-4S 的 I/O 的根域（例如, 如果释放 BB1，则为 root-dom1）不被视为移动目标，由于该根域处于 unbound 状态，在释放 SPARC M10-4S 时处于 inactive 状态。

iii.检查空内存块。

接下来，根据在步骤 i 中获得的检查结果，检查未被分配给未断开的 SPARC M10-4S 上的逻辑域的内存块（在下文中称为“空内存块”）。

在“表 A-10 内存块使用状况的示例”中，您可以看到空内存块的数量是 32 GB × 3 和 16 GB × 1。

iv.检查内存块是否可移动。

使用在步骤 ii 和步骤 iii 中获得的检查结果，检查源内存块是否可以移动到空内存块。

如果空内存块的容量等于或大于源内存块的容量，则可以移动。

在“表 A-10 内存块使用状况的示例”示例中，有 32 GB × 3 空内存块，作为 guest0 (32 GB)、guest1 (32 GB) 和 root-dom0 (24 GB) 的目标。因此，可以看到需要配置内存，以便可断开 BB1。在“表 A-11 目标备选内存块”中进行汇总。

如果是这种情况，请跳转到步骤 12。

表 A-11 目标备选内存块

SPARC M10-4S	容量	逻辑域	目标备选
BB1（更换的目标）	24 GB	root-dom0	-
	8 GB	未分配	-
	32 GB	guest0	BB0 中的 32 GB

表 A-11 目标备选内存块 (续)

SPARC M10-4S	容量	逻辑域	目标备选
BB0	32 GB	guest1	BB0 中的 32 GB
	24 GB	root-dom0	BB0 中的 32 GB
	6,912 MB	未分配	由于在 BB0 上存在 32-GB guest1 内存，将其删除。
	32 GB	未分配	被移到此处
	32 GB	未分配	被移到此处
	32 GB	未分配	被移到此处
	14 GB	primary	-
	16 GB	未分配	由于容量低排除目标备选

v. 如果没有可用目标，减少逻辑域中的内存分配。

此步骤介绍在没有足够的可用内存块作为目标备选时，如何进行操作。

如果未满足 "2.5.2 动态重新配置的系统操作注意事项" 中的 "CPU 内核和内存的配置" 所述的条件，内存块移动将失败。

例如，在 "表 A-12 无目标的内存块配置示例" 的配置中，源内存块的数量是 32 GB x 1 和 24 GB x 3。同时，目标处的可用内存块是 32 GB x 3 和 16 GB x 1。

因此，您可移动 BB1 中的一个 32-GB 内存块 (guest0) 和两个 24-GB 内存块 (guest1 和 guest2 两者之一和 root-dom0)。

然而，剩下的空目标内存块的数量是 16 GBx1 和 8 GBx2，在 24-GB 内存块已经被移动到 32 GB 的内存块后而获取。因此，无法移动分配 24-GB 内存块的任何 guest1、guest2 和 root-dom0。

表 A-12 无目标的内存块配置示例

SPARC M10-4S	容量	逻辑域	目标备选
BB1（更换的目标）	24 GB	guest2	无法移动
	8 GB	root-dom1	-
	32 GB	guest0	BB0 中的 32 GB
	32 GB	guest1	无法移动
	24 GB	root-dom0	无法移动
	6,912 MB	未分配	-
BB0	32 GB	未分配	被移到此处
	32 GB	未分配	guest1、guest2 和 root-dom0 (24 GB) 中的任意一个移到此处，剩余 8 GB。
	32 GB	未分配	guest1、guest2 和 root-dom0 (24 GB) 中的任意一个移到此处，剩余 8 GB。
	14 GB	primary	-

表 A-12 无目标的内存块配置示例 (续)

SPARC M10-4S	容量	逻辑域	目标备选
	16 GB	未分配	由于容量低排除目标备选

在这种情况下，您需要减小逻辑域中可能不可移动的内存块的容量为等于或小于目标中内存块的容量。

在上述示例中，您必须将逻辑域 `guest1` 和 `guest2` 两者之一或 `root-dom0` 的内存容量从 24 GB 变更到不大于16 GB。

下面的示例显示如何将来宾域 `guest2` 的内存容量变更到不大于 16 GB。

执行 `ldm list-domain` 命令检查逻辑域状态。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.5%	40m
guest0	active	-n----	5100	32	32G	0.0%	26m
guest1	active	-n----	5101	32	24G	0.0%	26m
guest2	active	-n----	5102	12	24G	0.1%	1m
root-dom0	active	-n--v-	5000	24	24G	0.0%	26m
root-dom1	active	-n--v-	5001	16	8G	0.0%	26m

执行 `ldm stop-domain` 命令停止域，然后执行 `ldm unbind-domain` 命令将其设置为 `inactive` 状态。

下面的示例执行命令将来宾域 `guest2` 设置为 `inactive` 状态。

```
# ldm stop-domain guest2
Ldom guest2 stopped
# ldm unbind-domain guest2
```

注 - 停止具有虚拟 I/O 服务配置的域之前，您必须使用 `ldm stop-domain` 命令首先停止分配相应虚拟 I/O 设备的来宾域。

如果 Oracle VM Server for SPARC 的版本早于 3.2，执行 `ldm set-memory` 命令重新配置目标逻辑域内存。

由于将 6 个内核 (12vcpu) 分配给该域，设置该配置以便来宾域 `guest2` 的容量不大于 16 GB (16,384 MB) 并且是（CPU 内核数量 x 256 MB）的倍数。

16384 / (6 x 256) = 10.66，切舍到 10

因此，重新配置来宾域 `guest2` 的内存容量为 (6 x 256 MB) x 10 = 15,360 MB (15 GB)。

下面的示例执行 `ldm set-memory` 命令重新配置内存容量为 15 GB。

```
# ldm set-memory 15G guest2
```

执行 `ldm list-domain` 命令确认内存容量是否已经变更为重新配置的值。

下面的示例执行 `ldm list-domain` 命令显示来宾域 `guest2` 的内存容量被重新配置为 15 GB。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.5%	40m
guest0	active	-n----	5100	32	32G	0.0%	26m
guest1	active	-n----	5101	32	24G	0.0%	26m
root-dom0	active	-n--v-	5000	24	24G	0.0%	26m
root-dom1	active	-n--v-	5001	16	8G	0.0%	26m
guest2	inactive	-----		12	15G		

执行 `ldm bind-domain` 和 `ldm start-domain` 命令绑定和启动逻辑域。

下面的示例执行 `ldm bind-domain` 和 `ldm start-domain` 命令启动来宾域 `guest2`。

```
# ldm bind-domain guest2
# ldm start-domain guest2
LDom guest2 started
```

执行 `ldm list-domain` 命令，然后确认逻辑域是否已经启动。

下面的示例执行 `ldm list-domain` 命令表明来宾域 `guest2` 已经以重新配置的内存容量启动。

确认 [STATE] 显示 “active” 并且 [FLAGS] 的字符串从左起第二个字符是 “n”。

# ldm bind-domain guest2 root-dom0							
-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.5%	44m
guest0	active	-n----	5100	32	32G	0.0%	30m
guest1	active	-n----	5101	32	24G	0.0%	30m
guest2	active	-n----	5102	12	15G	0.1%	1m
root-dom0	active	-n----	5000	24	24G	0.0%	30m
root-dom1	active	-n----	5001	16	8G	0.0%	30m

执行 `ldm list-devices -a memory` 命令检查内存块配置是否满足 [2.5.2 动态重新配置的系统操作注意事项](#) 中 “CPU 内核和内存的配置” 所述的条件。

下面的示例执行了 `ldm list-devices -a memory` 命令。

# ldm ls-devices -a mem							
MEMORY							
PA		SIZE	BOUND				
(BB0)							
0x700000000000		32G	guest0	→移动到	0x780000000000	(32 GB)	
0x720000000000		24G	root-dom0	→移动到	0x7a0000000000	(32 GB)	
0x720600000000		8G					

0x740000000000	8G	root-dom1 (不需要移动)
0x740200000000	24G	
0x760000800000	1272M	_sys_
0x760050000000	24G	guest1 →移动到 0x7c0000000000 (32 GB)
0x760650000000	6912M	
(BB0)		
0x780000000000	32G	
0x7a0000000000	32G	
0x7c0000000000	32G	
0x7e0000800000	1272M	_sys_
0x7e0050000000	512M	_sys_
0x7e0070000000	256M	_sys_
0x7e0080000000	14G	primary
0x7e0400000000	15G	guest2
0x7e07c0000000	1G	

12. 将配置的逻辑域配置信息保存在 XSCF 中。

执行 `ldm set-spconfig` 命令保存配置的信息。

下面的示例检查保存的配置信息，然后使用与现有的配置信息相同的名称将其保存。

执行 `ldm list-spconfig` 命令检查当前配置信息。

```
# ldm list-spconfig
factory-default
ldm-set1 [next poweron]
```

执行 `ldm remove-spconfig` 命令删除需要覆盖的配置信息。

```
# ldm remove-spconfig ldm-set1
```

执行 `ldm add-spconfig` 命令重新保存配置信息。

```
# ldm add-spconfig ldm-set1
```

执行 `ldm list-spconfig` 命令检查保存的配置信息是否成为 [current]。

```
# ldm list-spconfig
factory-default
ldm-set1 [current]
```

13. 将配置的逻辑域配置信息保存到 XML 文件。

为了防止保存到 XSCF 的配置信息无法使用，请将配置信息保存到 XML 文件。建议将 XML 文件保存到不同的介质。

下面介绍示例的步骤。

执行 `ldm list-domain` 命令检查是否所有逻辑域都处于 active 状态。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.0%	10h 7m
guest0	active	-n----	5100	32	32G	0.0%	2h 20s
guest1	active	-n----	5101	32	32G	0.0%	2h 5s
root-dom0	active	-n----	5000	24	24G	0.0%	2h 43s
root-dom1	active	-n----	5001	24	24G	0.0%	2h 20s

执行 `ldm list-constraints` 命令将配置信息保存到 XML 文件。

ldm list-constraints -x > /ldm-set1.xml

A.3.3 活动状态更换的步骤示例（适用于 Oracle VM Server for SPARC 3.1.x）

本节介绍使用 "图 A-3 操作连续性（具有可用资源）的 2BB 配置示例" 中所述的 2BB 配置系统的 PPAR DR 活动状态更换 BB#01 的步骤示例。

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。
 在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

1. 登录主 XSCF。
 执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。
 如果您已经登录到备用 XSCF，注销后再次登录到主 XSCF。

XSCF> showbbstatus
BB#00 (Master)

2. 执行 `showhardconf` 命令检查需要更换的 SPARC M10-4S 中的 XSCF 的 [Status] 是否是 "Normal"。

XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Locked;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;


```

+ Freq:3.000 GHz; Type:0x10;
+ Core:16; Strand:2;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
:

```

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。
在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

3. 执行 **showbbstatus** 命令来确认需要更换的 **SPARC M10-4S** 中的 **XSCF** 不是主 **XSCF**。

```

XSCF> showbbstatus
BB#00 (Master)

```

如果需要更换的 SPARC M10-4S 是主 XSCF，则执行 **switchscf** 命令切换 XSCF。

```

XSCF> switchscf -t Standby
The XSCF unit switch between the Active and Standby states.
Continue? [y|n] :y

```

注 - 释放 SPARC M10-4S 前，请确认 XSCF 已经切换并重新启动。

4. 执行 **console** 命令连接到控制域的控制台，然后登录。

```

XSCF> console -p 0

```

5. 检查逻辑域的运行状态和资源使用状况。

a. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义：

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

下面的示例表明控制域、两个根域和两个来宾域正在运行。

检查所有的域是处于表明 Oracle Solaris 处于运行状态的 "active"，还是

"inactive" 状态。如果存在处于 OpenBoot PROM 状态或绑定状态的域，则物理分区的动态重新配置可能失败。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.0%	10h 7m
guest0	active	-n----	5100	32	32G	0.0%	2h 20s
guest1	active	-n----	5101	32	32G	0.0%	2h 5s
root-dom0	active	-n--v-	5000	24	24G	0.0%	2h 43s
root-dom1	active	-n--v-	5001	24	24G	0.0%	2h 20s

b. 执行指定 **-a** 选项的 **ldm list-devices** 命令来检查资源的使用状况。

在下面的示例中，指定 **-a** 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。

# ldm list-devices -a			
CORE			
	ID	%FREE	CPUSET
	0	0	(0, 1)
	4	0	(8, 9)
	8	0	(16, 17)
(省略)			
	944	0	(1888, 1889)
	948	0	(1896, 1897)
	952	0	(1904, 1905)
	956	0	(1912, 1913)
VCPU			
	PID	%FREE	PM
	0	0	no
	1	0	no
	8	0	no
	9	0	no
(省略)			
	1904	0	no
	1905	0	no
	1912	0	no
	1913	0	no
(省略)			

6. 释放控制域中系统卷和 I/O 设备的冗余配置。

本步骤介绍如何释放需要更换的 SPARC M10-4S 的 I/O 设备以及控制域中使用的 I/O 设备，以便启用 BB-ID#01 SPARC M10-4S 的释放，如果您正在使用其他冗余配置软件，请参见该冗余配置的软件文档，了解如何取消配置的详细信息。

a. 取消控制域中系统卷的冗余配置。

下面的示例介绍如何取消控制域中系统卷的 ZFS 镜像功能。

在控制域中执行 **zpool status** 命令检查镜像配置状态。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
    NAME                                STATE      READ WRITE CKSUM
    rpool                               ONLINE    0     0     0
        mirror-0                       ONLINE    0     0     0
            c2t50000393E802CCE2d0s0    ONLINE    0     0     0
            c3t50000393A803B13Ed0s0    ONLINE    0     0     0
errors: No known data errors
```

执行 `zpool detach` 命令从镜像配置中释放磁盘。

```
# zpool detach rpool c3t50000393A803B13Ed0
```

执行 `zpool status` 命令确认镜像配置已经取消。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
    NAME                                STATE      READ WRITE CKSUM
    rpool                               ONLINE    0     0     0
        mirror-0                       ONLINE    0     0     0
            c2t50000393E802CCE2d0s0    ONLINE    0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则移除冗余配置或停止使用那些设备。有关如何取消冗余配置或停止使用设备的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

b. 删除控制域的 I/O 配置。

通过延迟重新配置，从分配给控制域的物理 I/O 设备中，删除 BB#01 中的根合成体。

首先，将控制域设置到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

执行 `ldm list-io` 命令检查分配给 `primary` 的根合成体。

下面的示例表明 BB1 设备的根合成体是 PCIE8 和 PCIE12。

```
# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary    IOV
PCIE4          BUS      PCIE4      primary    IOV
PCIE8          BUS      PCIE8      primary    IOV
PCIE12         BUS      PCIE12     primary    IOV
/BB0/CMUL/NET0 PCIE     PCIE0      primary    OCC
/BB0/CMUL/SASHBA PCIE     PCIE0      primary    OCC
/BB0/CMUL/NET2 PCIE     PCIE4      primary    OCC
/BB1/CMUL/NET0 PCIE     PCIE8      primary    OCC
/BB1/CMUL/SASHBA PCIE     PCIE8      primary    OCC
/BB1/CMUL/NET2 PCIE     PCIE12     primary    OCC
```

执行 ldm remove-io 命令从 primary 删除 PCIE8 和 PCIE12，然后重新启动 Oracle Solaris。

```
# ldm remove-io PCIE8 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

# ldm remove-io PCIE12 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

# shutdown -i6 -g0 -y
....
```

Oracle Solaris 启动后，执行 ldm list-io 命令确认 BB#01 中的根合成体已经从控制域删除。

```
# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary    IOV
PCIE4          BUS      PCIE4      primary    IOV
/BB0/CMUL/NET0 PCIE     PCIE0      primary    OCC
/BB0/CMUL/SASHBA PCIE     PCIE0      primary    OCC
/BB0/CMUL/NET2 PCIE     PCIE4      primary    OCC
```

- c. 取消分配给来宾域的虚拟 I/O 设备的冗余配置。
- 若需首先关闭分配 BB#01 中的根合成体的根域 (root-dom1)，请登录各来宾域，然后从 root-dom1 中取消虚拟 I/O 设备的冗余配置。
- 有关如何使用冗余配置软件的详情，请参见该冗余配置软件的文档。
- 在下面的示例中，将虚拟网络设备 (vnet1) 从 IPMP 配置中取消。有关命令的详情，请参见 Oracle Solaris 的手册。
- 登录来宾域 (guest0)。

```
# ldm list-domain
NAME          STATE      FLAGS      CONS      VCPU      MEMORY      UTIL      UPTIME
primary       active    -n-cv-    UART      64        56G         0.0%     4h 17m
```

```
guest0          active      -n---- 5100    64    64G    0.0% 1h 13m
guest1          active      -n---- 5101    64    64G    0.0% 1h  4m
root-dom0       active      -n--v- 5000    32    32G    0.0% 1h 47m
root-dom1       active      -n--v- 5001    32    32G    0.0% 1h 19m
# telnet localhost 5100
....
guest0#
```

执行 `dladm show-phys` 命令检查虚拟网络接口 (`vnet1`) 和网络接口的名称 (`net1`) 之间的映射。

```
guest0# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      up       0      unknown  vnet0
net1          Ethernet      up       0      unknown  vnet1
```

执行 `ipmpstat -i` 命令检查配置 IPMP 的网络接口的配置信息。

```
guest0# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net0       yes    ipmp0      -smbM--    up        disabled   ok
net1       no     ipmp0      is-----  up        disabled   ok
```

执行 `if_mpadm -d` 命令从 IPMP 组释放 `net1`，然后执行 `ipmpstat -i` 命令确认已经将其释放。下面的示例确认 STATE 是 `offline`。

```
guest0# if_mpadm -d net1
guest0# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net0       yes    ipmp0      -smbM--    up        disabled   ok
net1       no     ipmp0      -s---d-    up        disabled   offline
```

执行 `ipadm delete-ip` 命令删除 `net1`。

```
guest0# ipadm delete-ip net1
```

同样，对来宾域 (`guest1`) 执行相同的释放处理。

d. 从根域移除已分配的虚拟 I/O 设备，以便将其停止。

按照下述步骤，执行 `ldm remove-vdisk` 和 `ldm remove-vnet` 命令从根域移除已分配的虚拟磁盘 (`vdisk`) 和虚拟网络设备 (`vnet`)，以便将其停止。

下面的示例显示移除使用 BB#01 根域 (`root-dom1`) 的虚拟 I/O 服务的虚拟磁盘 (`vdisk10`) 和虚拟网络设备 (`vnet10`) 的命令的执行。

对来宾域 (`guest1`) 执行相同的删除操作。

```
# ldm remove-vdisk vdisk10 guest0
# ldm remove-vnet vnet10 guest0
```

7. 检查 I/O 设备的资源使用状况，然后取消需要更换的 **SPARC M10-4S** 中的所有 I/O 设备。
- a. 检查分配需要释放的 **SPARC M10-4S** 中的根合成体的逻辑域。

执行 `ldm list-io` 命令，以检查分配 BB#01 中的根合成体的逻辑域。

下面的示例表明仅 `root-dom1` 具有以 `"/BB1/"` 开始的 PCIe 端点。您可以看到 PCIe 端点根合成体 (BUS) PCIe9、PCIe10、PCIe11、PCIe13、PCIe14 和 PCIe15 被分配给 `root-dom1`。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV
....				
/BB1/CMUL/NET0	PCIE	PCIE8		UNK
/BB1/CMUL/SASHBA	PCIE	PCIE8		UNK
/BB1/PCI0	PCIE	PCIE9	root-dom1OCC	
/BB1/PCI3	PCIE	PCIE10	root-dom1OCC	
/BB1/PCI4	PCIE	PCIE10	root-dom1OCC	
/BB1/PCI7	PCIE	PCIE11	root-dom1OCC	
/BB1/PCI8	PCIE	PCIE11	root-dom1OCC	
/BB1/CMUL/NET2	PCIE	PCIE12		UNK
/BB1/PCI1	PCIE	PCIE13	root-dom1OCC	
/BB1/PCI2	PCIE	PCIE13	root-dom1OCC	
/BB1/PCI5	PCIE	PCIE14	root-dom1OCC	
/BB1/PCI6	PCIE	PCIE14	root-dom1OCC	
/BB1/PCI9	PCIE	PCIE15	root-dom1OCC	
/BB1/PCI10	PCIE	PCIE15	root-dom1OCC	

- b. 停止分配需要释放的 **SPARC M10-4S** 中的根合成体的逻辑域。

下面的示例执行 `ldm stop-domain` 和 `ldm unbind-domain` 命令释放根域 (`root-dom1`)，并表明根域处于 `inactive` 状态。

```
# ldm stop-domain root-dom1
LDom root-dom1 stopped
# ldm unbind-domain root-dom1
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.2%	4h 59m
guest0	active	-n----	5100	32	32G	0.0%	1h 55m
guest1	active	-n----	5101	32	32G	0.0%	1h 46m
root-dom0	active	-n--v-	5000	24	24G	0.0%	2h 29m
root-dom1	inactive	-----		24	24G		

c. 确认需要更换的 **SPARC M10-4S** 中的所有 I/O 设备已经被取消。

执行 `ldm list-io` 命令确认已经释放所有 I/O 设备。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10		
PCIE11	BUS	PCIE11		
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13		
PCIE14	BUS	PCIE14		
PCIE15	BUS	PCIE15		
(省略)				

8. 从物理分区释放 **SPARC M10-4S** 的系统板 (PSB<BB>)。

a. 执行 `deleteboard -c disconnect` 命令将 PSB 从物理分区释放。

```
XSCF> deleteboard -c disconnect 01-0
PSB#01-0 will be unconfigured from PPAR immediately. Continue?
[y|n] :y
Start unconfigure preparation of PSB. [1200sec]
0end
Unconfigure preparation of PSB has completed.
Start unconfiguring PSB from PPAR. [7200sec]
0..... 30..... 60....end
Unconfigured PSB from PPAR.
PSB power off sequence started. [1200sec]
0..... 30..... 60..... 90.....120.....150.....end
Operation has completed.
```

b. 执行 `showresult` 命令检查刚刚执行的 `deleteboard` 命令的结束状态。

结束值 0 表明 `deleteboard` 命令正常终止。

如果结束值非 0，或如果在执行 `deleteboard` 命令时显示错误信息，则表明 `deleteboard` 命令异常终止。根据错误信息，通过参考 ["C.1.2 deleteboard"](#) 识

别错误，然后采取应对措施。

```
XSCF> showresult
0
```

c. 执行 **showboards** 命令检查 **PSB** 状态。

确认需要更换的 SPARC M10-4S 内的 PSB 是否处于 “Assigned” 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 “n”。

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB)  Assignment  Pwr  Conn  Conf  Test  Fault
-----
00-0  00(00)         Assigned    y    y    y    Passed Normal
01-0  00(01)         Assigned    n    n    n    Passed Normal
```

9. 执行 **replacefru** 命令来更换 **SPARC M10-4S**。

```
XSCF> replacefru
```

注 -关于使用 **replacefru** 命令更换 SPARC M10-4S 的详情，请参见《Fujitsu M10-4/Fujitsu M10-4S/SPARC M10-4/SPARC M10-4S Service Manual》中的 "5.8 Releasing a SPARC M10-4/M10-4S FRU from the System with the **replacefru** Command" 和 "6.2 Incorporating a SPARC M10-4/M10-4S FRU into the System with the **replacefru** Command"。

10. 将 **PSB** 整合到物理分区。

a. 执行 **showboards** 命令检查 **PSB** 状态。

确认需要更换的 SPARC M10-4S 内的 PSB 是否处于 Assigned 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 “n”。

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB)  Assignment  Pwr  Conn  Conf  Test  Fault
-----
00-0  00(00)         Assigned    y    y    y    Passed Normal
01-0  00(01)         Assigned    n    n    n    Passed Normal
```

b. 执行 **addboard -c configure** 命令将 **PSB** 整合到物理分区。

若需恢复初始逻辑域配置，请执行指定 **-m bind=resource** 选项的 **addboard -c configure** 命令。

```
XSCF> addboard -c configure -m bind=resource -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
810.....840.....870.....900.....930.....960.....end
```



```
Connected PSB to PPAR.  
Start configuring PSB to Logical Domains (LDoms) Manager.  
[1800sec] 0.....end  
Configured PSB to Logical Domains (LDoms) Manager.  
Operation has completed.
```

注 - 如果执行 **addboard** 命令过程中出现错误信息，请参见 ["C.1.1 addboard"](#)，然后识别错误并采取应对措施。

c. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。

结束值 0 表明 **addboard** 命令正常终止。

如果结束值非 0，或如果在执行 **addboard** 命令时显示错误信息，则表明 **addboard** 命令异常终止。根据错误信息，通过参考 ["C.1.1 addboard"](#) 识别错误，然后采取应对措施。

```
XSCF> showresult  
0
```

d. 执行 **showboards** 命令检查 **PSB** 状态。

在已更换的 SPARC M10-4S 中的 PSB 已经被成功整合后，确认 [Conn] 和 [Conf] 栏是否都显示 "y"。

```
XSCF> showboards -p 0
```

PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

11. 检查逻辑域的运行状态。

a. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

b. 执行 **ldm list-domain** 命令来确认 **SPARC M10-4S PSB (BB)** 添加后逻辑域的运行状态未发生变化。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义：

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.0%	10h 30m
guest0	active	-n----	5100	32	32G	0.0%	2h 50s

guest1	active	-n----	5101	32	32G	0.0%	2h 30s
root-dom0	active	-n--v-	5000	24	24G	0.0%	3h 20s
root-dom1	inactive	-----		24	24G		

12. 重新启动 I/O 设备的使用。

a. 重新分配根合成体。

执行 `ldm bind-domain` 和 `ldm start-domain` 命令，以便启动在 `unbind` 状态下分配已更换的 SPARC M10-4S 中的根合成体的根域。

下面的示例启动在 `unbind` 状态下的根域 (`root-dom1`)，然后确认是否已经启动。

```
# ldm bind-domain root-dom1
# ldm start-domain root-dom1
LDom root-dom1 started
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	16G	0.2%	3h 8m
guest0	active	-n----	5100	32	32G	0.0%	3h 8m
guest1	active	-n----	5101	32	32G	0.0%	3h 8m
root-dom0	active	-n--v-	5000	24	24G	0.0%	3h 8m
root-dom1	active	-n--v-	5001	24	24G	7.3%	8s

执行 `ldm list-io` 命令确认物理 I/O 设备是否已经分配给刚刚启动的根域。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV
....				
/BB1/CMUL/NET0	PCIE	PCIE8		UNK
/BB1/CMUL/SASHBA	PCIE	PCIE8		UNK
/BB1/PCI0	PCIE	PCIE9	root-dom1OCC	
/BB1/PCI3	PCIE	PCIE10	root-dom1OCC	
/BB1/PCI4	PCIE	PCIE10	root-dom1OCC	
/BB1/PCI7	PCIE	PCIE11	root-dom1OCC	
/BB1/PCI8	PCIE	PCIE11	root-dom1OCC	

/BB1/CMUL/NET2	PCIE	PCIE12	UNK
/BB1/PCI1	PCIE	PCIE13	root-dom1OCC
/BB1/PCI2	PCIE	PCIE13	root-dom1OCC
/BB1/PCI5	PCIE	PCIE14	root-dom1OCC
/BB1/PCI6	PCIE	PCIE14	root-dom1OCC
/BB1/PCI9	PCIE	PCIE15	root-dom1OCC
/BB1/PCI10	PCIE	PCIE15	root-dom1OCC

b. 将虚拟 I/O 设备从根域添加到来宾域。

执行 `ldm add-vdisk` 和 `ldm add-vnet` 命令，将支持已经启动的根域的虚拟 I/O 服务的虚拟磁盘 (vdisk) 和虚拟网络设备 (vnet) 添加到各个来宾域。

下面的示例显示添加使用 BB#01 根域 (root-dom1) 的虚拟 I/O 服务的虚拟磁盘 (vdisk10) 和虚拟网络设备 (vnet10) 的命令的执行。

对来宾域 (guest1) 执行相同的添加操作。

```
# ldm add-vdisk id=1 vdisk10 vol110@vds1 guest0
# ldm add-vnet id=1 vnet10 vsw10 guest0
```

注 - 若需再次添加虚拟 I/O 设备，有必要提前指定已分配的 ID。您可以在虚拟 I/O 设备被删除之前，在被使用的状态下，从 `ldm list -l` 命令的执行结果检查 ID。

c. 将分配给来宾域的虚拟 I/O 设备整合到冗余配置中。

启动分配 BB#1 中的根合成体的根域 (root-dom1) 后，对应各来宾域的虚拟 I/O 设备服务也被启动。

登录各来宾域，然后从以前被取消的 root-dom1 中将虚拟 I/O 设备整合到冗余配置。有关如何使用冗余配置软件的详情，请参见该冗余配置软件的文档。

下面的示例介绍将虚拟网络设备 (vnet1) 整合到 IPMP 配置的示例。有关命令的详情，请参见 Oracle Solaris 的手册。

首先登录来宾域 (guest0)。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    16      14G       0.0%   4h 17m
guest0        active    -n----   5100    32      32G       0.0%   1h 13m
guest1        active    -n----   5101    32      32G       0.0%   1h 4m
root-dom0     active    -n----   5000    24      24G       0.0%   1h 47m
root-dom1     active    -n----   5001    24      24G       0.0%   1h 19m
# telnet localhost 5100
....
guest0#
```

执行 `dladm show-phys` 命令检查虚拟网络接口 (vnet1) 和网络接口的名称 (net1) 之间的映射。

```
guest0# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	0	unknown	vnet0
net1	Ethernet	up	0	unknown	vnet1

执行 `ipadm create-ip` 命令、`ipadm set-ifprop` 命令以及 `ipadm add-ipmp` 命令将 `net1` 注册为 `ipmp0` 的备用设备。

```
guest0# ipadm create-ip net1
guest0# ipadm set-ifprop -p standby=on -m ip net1
guest0# ipadm add-ipmp -i net1 ipmp0
```

执行 `ipmpstat -i` 命令确认配置 IPMP 的网络接口的 `STATE` 是否显示 `ok`。
对另一个来宾域 (`guest1`)，执行相同的步骤。

```
guest0# ipmpstat -i
INTERFACE ACTIVE GROUP FLAGS LINK PROBE STATE
net0 yes ipmp0 -smbM-- up disabled ok
net1 no ipmp0 -s---d- up disabled ok
```

13. 将控制域上的系统卷和 I/O 设备恢复到冗余配置。
 - a. 添加控制域的根合成体配置。

通过延迟配置的方式，在以前从控制域中被移除的 `BB#01` 中添加根合成体。
首先，将控制域设置到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

执行 `ldm list-io` 命令检查未分配的根合成体。
下面的示例表明 `BB1` 设备的根合成体，`PCIE8` 和 `PCIE12` 未被分配。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV

PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
(省略)				

执行 `ldm add-io` 命令将 PCIE8 和 PCIE12 添加到 `primary` , 然后重新启动 Oracle Solaris。

```
# ldm add-io PCIE8 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

# ldm add-io PCIE12 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

# shutdown -i6 -g0 -y
....
```

Oracle Solaris 启动后，执行 `ldm list-io` 命令确认 BB#01 中的根合成体是否已经被添加到控制域。

```
# ldm list-io | grep primary
```

PCIE0	BUS	PCIE0	primary	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE8	BUS	PCIE8	primary	
PCIE12	BUS	PCIE12	root-dom1	
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/CMUL/NET2	PCIE	PCIE4	primary	OCC

- b. 将控制域中的系统卷设置为冗余配置。
 在控制域中执行 `zpool status` 命令检查镜像配置状态。
 下面的示例介绍如何配置控制域中系统卷的 ZFS 镜像功能。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 29.1M in 0h0m with 0 errors on Thu Jan 23 17:27:59 2014
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                                ONLINE    0     0     0
    c2t50000393E802CCE2d0s0          ONLINE    0     0     0
errors: No known data errors
```

执行 `zpool attach` 命令，以便将磁盘整合到镜像配置中。

```
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting.
#
```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。

使用 `zpool status` 命令确认同步处理 (resilver) 是否完成。

下面显示同步处理过程中显示的示例。

```
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
       continue to function in a degraded state.
action: Wait for the resilver to complete.
       Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Jan 27 15:55:47 2014
      21.1G scanned out of 70.6G at 120M/s, 0h7m to go
      21.0G resilvered, 29.84% done
config:
      NAME                                STATE      READ  WRITE CKSUM
      rpool                                DEGRADED   0     0     0
        mirror-0                          DEGRADED   0     0     0
          c2t50000393E802CCE2d0s0         ONLINE    0     0     0
          c3t50000393A803B13Ed0s0         DEGRADED   0     0     0 (resilvering)
errors: No known data errors
```

同步处理完成后，将显示如下画面：

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:
      NAME                                STATE      READ  WRITE CKSUM
      rpool                                ONLINE     0     0     0
        mirror-0                          ONLINE     0     0     0
          c2t50000393E802CCE2d0s0         ONLINE    0     0     0
          c3t50000393A803B13Ed0s0         ONLINE    0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则创建冗余配置或恢复使用那些设备。有关如何创建冗余配置或恢复设备使用的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

A.3.4 活动状态更换步骤示例（当 PCIe 总线的动态分配可用时）

本节介绍使用 "图 A-3 操作连续性（具有可用资源）的 2BB 配置示例" 所述的 2BB 配

置系统的 PPAR DR 活动状态更换 BB#01 的步骤示例。此示例适用于 PCIe 总线可动态分配的环境（Oracle VM Server for SPARC 3.2 或更高版本的 XCP 2240 或更高版本和 Oracle Solaris 11.2 SRU11.2.8.4.0 或更高版本的根域）。

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。

在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

1. 登录主 **XSCF**。

执行 **showbbstatus** 命令来检查您已经登录的 XSCF 是否是主 XSCF。

如果您已经登录到备用 XSCF，注销然后再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

2. 执行 **showhardconf** 命令检查需要更换的 **SPARC M10-4S** 中的 **XSCF** 的 **[Status]** 是否是 **"Normal"**。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Locked;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;
+ Freq:3.000 GHz; Type:0x10;
+ Core:16; Strand:2;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
:
```

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。

在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

3. 执行 **showbbstatus** 命令来确认需要更换的 **SPARC M10-4S** 中的 **XSCF** 不是主 **XSCF**。

```
XSCF> showbbstatus
BB#00 (Master)
```

如果需要更换的 SPARC M10-4S 是主 XSCF，则执行 **switchscf** 命令切换 XSCF。

```
XSCF> switchscf -t Standby
The XSCF unit switch between the Active and Standby states.
Continue? [y|n] :y
```

注 - 释放系统板前，请确认 XSCF 已经切换并重新启动。

4. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

5. 检查逻辑域的运行状态和资源使用状况。
- a. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。
- 若需检查逻辑域的运行状态，请确认 [STATE] 显示 "active"，并检查 FLAGS 内的字符串从左起第二个字符。显示的字符及其含义如下：

- "n": Oracle Solaris 正在运行
- "t": OpenBoot PROM 状态
- "-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

下面的示例表明控制域、两个根域和两个来宾域正在运行。

检查所有的域是处于表明 Oracle Solaris 处于运行状态的 "active"，还是 "inactive" 状态。如果存在处于 OpenBoot PROM 状态或绑定状态的域，则物理分区的动态重新配置可能失败。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.0%	10h 7m
guest0	active	-n----	5100	32	32G	0.0%	2h 20s
guest1	active	-n----	5101	32	32G	0.0%	2h 5s
root-dom0	active	-n--v-	5000	24	24G	0.0%	2h 43s
root-dom1	active	-n--v-	5001	24	24G	0.0%	2h 20s

- b. 执行指定 **-a** 选项的 **ldm list-devices** 命令来检查资源的使用状况。
- 在下面的示例中，指定 **-a** 选项来显示绑定到逻辑域的所有资源和未绑定的所有资源。


```
# ldm list-devices -a
CORE
  ID      %FREE    CPUSET
  0        0      (0, 1)
  4        0      (8, 9)
  8        0     (16, 17)
(省略)
  944      0     (1888, 1889)
  948      0     (1896, 1897)
  952      0     (1904, 1905)
  956      0     (1912, 1913)
VCPU
  PID      %FREE    PM
  0        0      no
  1        0      no
  8        0      no
  9        0      no
(省略)
  1904     0      no
  1905     0      no
  1912     0      no
  1913     0      no
(省略)
```

6. 释放控制域中系统卷和 I/O 设备的冗余配置。
若需释放模块化架构 BB-ID#01，此步骤介绍如何释放需要更换的 SPARC M10-4S 的 I/O 设备和控制域中使用的 I/O 设备。如果您正在使用其他的冗余配置软件，有关如何取消配置的详情，请参见该冗余配置软件的文档。

a. 取消控制域中系统卷的冗余配置。

下面的示例介绍如何取消控制域中系统卷的 ZFS 镜像功能。

在控制域中执行 `zpool status` 命令检查镜像配置状态。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                               ONLINE    0     0     0
    mirror-0                          ONLINE    0     0     0
      c2t50000393E802CCE2d0s0        ONLINE    0     0     0
      c3t50000393A803B13Ed0s0        ONLINE    0     0     0
errors: No known data errors
```

执行 `zpool detach` 命令从镜像配置中释放磁盘。

```
# zpool detach rpool c3t50000393A803B13Ed0
```

执行 `zpool status` 命令确认镜像配置已经取消。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
    NAME                                STATE      READ WRITE CKSUM
    rpool                               ONLINE    0     0     0
        mirror-0                       ONLINE    0     0     0
            c2t50000393E802CCE2d0s0   ONLINE    0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则移除冗余配置或停止使用那些设备。有关如何取消冗余配置或停止使用设备的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

b. 删除控制域的 I/O 配置。

从分配给控制域的物理 I/O 设备，删除 BB#01 中的根合成体。

执行 `ldm list-io` 命令检查分配给 `primary` 的根合成体。

下面的示例表明 BB1 设备的根合成体是 PCIE8 和 PCIE12。

```
# ldm list-io | grep primary
PCIE0                                BUS      PCIE0      primary  IOV
PCIE4                                BUS      PCIE4      primary  IOV
PCIE8                                BUS      PCIE8      primary  IOV
PCIE12                               BUS      PCIE12     primary  IOV
/BB0/CMUL/NET0                      PCIE     PCIE0      primary  OCC
/BB0/CMUL/SASHBA                    PCIE     PCIE0      primary  OCC
/BB0/CMUL/NET2                      PCIE     PCIE4      primary  OCC
/BB1/CMUL/NET0                      PCIE     PCIE8      primary  OCC
/BB1/CMUL/SASHBA                    PCIE     PCIE8      primary  OCC
/BB1/CMUL/NET2                      PCIE     PCIE12     primary  OCC
```

执行 `ldm remove-io` 命令从主域删除 PCIE8 和 PCIE12 。

```
# ldm remove-io PCIE8 primary
# ldm remove-io PCIE12 primary
```

执行 `ldm list-io` 命令确认 BB#01 中的根合成体已经从控制域删除。

```
# ldm list-io | grep primary
PCIE0                                BUS      PCIE0      primary  IOV
PCIE4                                BUS      PCIE4      primary  IOV
/BB0/CMUL/NET0                      PCIE     PCIE0      primary  OCC
/BB0/CMUL/SASHBA                    PCIE     PCIE0      primary  OCC
/BB0/CMUL/NET2                      PCIE     PCIE4      primary  OCC
```

c. 取消分配给来宾域的虚拟 I/O 设备的冗余配置。

若需首先关闭分配 BB#01 中的根合成体的根域 (`root-dom1`)，请登录各来宾域，然后从 `root-dom1` 中取消虚拟 I/O 设备的冗余配置。

有关如何使用冗余配置软件的详情，请参见该冗余配置软件的文档。

在下面的示例中，将虚拟网络设备 (vnet1) 从 IPMP 配置中取消。有关命令的详情，请参见 Oracle Solaris 的手册。

登录来宾域 (guest0)。

```
# ldm list-domain
NAME                STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary             active     -n-cv-   UART    64      56G       0.0%    4h 17m
guest0              active     -n----   5100    64      64G       0.0%    1h 13m
guest1              active     -n----   5101    64      64G       0.0%    1h 4m
root-dom0           active     -n--v-   5000    32      32G       0.0%    1h 47m
root-dom1           active     -n--v-   5001    32      32G       0.0%    1h 19m
# telnet localhost 5100
....
guest0#
```

执行 `dladm show-phys` 命令检查虚拟网络接口 (vnet1) 和网络接口的名称 (net1) 之间的映射。

```
guest0# dladm show-phys
LINK                MEDIA                STATE    SPEED    DUPLEX    DEVICE
net0                 Ethernet             up       0         unknown   vnet0
net1                 Ethernet             up       0         unknown   vnet1
```

执行 `ipmpstat -i` 命令检查配置 IPMP 的网络接口的配置信息。

```
guest0# ipmpstat -i
INTERFACE  ACTIVE  GROUP    FLAGS    LINK    PROBE    STATE
net0       yes    ipmp0    -smbM--  up      disabled ok
net1       no     ipmp0    is----- up      disabled ok
```

执行 `if_mpadm -d` 命令从 IPMP 组释放 net1，然后执行 `ipmpstat -i` 命令确认已经将其释放。下面的示例确认 STATE 是 offline。

```
guest0# if_mpadm -d net1
guest0# ipmpstat -i
INTERFACE  ACTIVE  GROUP    FLAGS    LINK    PROBE    STATE
net0       yes    ipmp0    -smbM--  up      disabled ok
net1       no     ipmp0    -s---d-  up      disabled offline
```

执行 `ipadm delete-ip` 命令删除 net1。

```
guest0# ipadm delete-ip net1
```

同样，对来宾域 (guest1) 执行相同的释放处理。

d. 从根域移除已分配的虚拟 I/O 设备，以便将其停止。

按照下述步骤，执行 `ldm remove-vdisk` 和 `ldm remove-vnet` 命令从根域移除

已分配的虚拟磁盘 (vdisk) 和虚拟网络设备 (vnet)，以便将其停止。

下面的示例显示移除使用 BB#01 根域 (root-dom1) 的虚拟 I/O 服务的虚拟磁盘 (vdisk10) 和虚拟网络设备 (vnet10) 的命令的执行。

```
# ldm remove-vdisk vdisk10 guest0
# ldm remove-vnet vnet10 guest0
```

对来宾域 (guest1) 执行相同的删除操作。

7. 检查 I/O 设备的资源使用状况，然后取消需要更换的 **SPARC M10-4S** 中的所有 I/O 设备。

a. 检查分配需要释放的 **SPARC M10-4S** 中的根合成体的逻辑域。

执行 ldm list-io 命令，以检查分配 BB#01 中的根合成体的逻辑域。

下面的示例表明仅 root-dom1 具有以 "/BB1/" 开始的 PCIe 端点。您可以看到 PCIe 端点根合成体 (BUS) PCIE9、PCIE10、PCIE11、PCIE13、PCIE14 和 PCIE15 被分配给 root-dom1。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV
....				
/BB1/CMUL/NET0	PCIE	PCIE8		UNK
/BB1/CMUL/SASHBA	PCIE	PCIE8		UNK
/BB1/PCI0	PCIE	PCIE9	root-dom1	OCC
/BB1/PCI3	PCIE	PCIE10	root-dom1	OCC
/BB1/PCI4	PCIE	PCIE10	root-dom1	OCC
/BB1/PCI7	PCIE	PCIE11	root-dom1	OCC
/BB1/PCI8	PCIE	PCIE11	root-dom1	OCC
/BB1/CMUL/NET2	PCIE	PCIE12		UNK
/BB1/PCI1	PCIE	PCIE13	root-dom1	OCC
/BB1/PCI2	PCIE	PCIE13	root-dom1	OCC
/BB1/PCI5	PCIE	PCIE14	root-dom1	OCC
/BB1/PCI6	PCIE	PCIE14	root-dom1	OCC
/BB1/PCI9	PCIE	PCIE15	root-dom1	OCC
/BB1/PCI10	PCIE	PCIE15	root-dom1	OCC

b. 停止分配需要释放的 **SPARC M10-4S** 中的根合成体的逻辑域。

下面的示例执行 `ldm stop-domain` 和 `ldm unbind-domain` 命令释放根域 (`root-dom1`), 并表明根域处于 `inactive` 状态。

```
# ldm stop-domain root-dom1
LDom root-dom1 stopped
# ldm unbind-domain root-dom1
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.2%	4h 59m
guest0	active	-n----	5100	32	32G	0.0%	1h 55m
guest1	active	-n----	5101	32	32G	0.0%	1h 46m
root-dom0	active	-n--v-	5000	24	24G	0.0%	2h 29m
root-dom1	inactive	-----		24	24G		

c. 确认需要更换的 **SPARC M10-4S** 中的所有 I/O 设备已经被取消。

执行 `ldm list-io` 命令确认已经释放所有 I/O 设备。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10		
PCIE11	BUS	PCIE11		
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13		
PCIE14	BUS	PCIE14		
PCIE15	BUS	PCIE15		
(省略)				

8. 从物理分区释放 **SPARC M10-4S** 的系统板 (PSB<BB>)。

a. 执行 `deleteboard -c disconnect` 命令将 **PSB** 从物理分区释放。

```
XSCF> deleteboard -c disconnect 01-0
PSB#01-0 will be unconfigured from PPAR immediately. Continue?
[y|n] :y
Start unconfigure preparation of PSB. [1200sec]
0end
Unconfigure preparation of PSB has completed.
Start unconfiguring PSB from PPAR. [7200sec]
0..... 30..... 60....end
Unconfigured PSB from PPAR.
```

```
PSB power off sequence started. [1200sec]
0..... 30..... 60..... 90.....120.....150.....end
Operation has completed.
```

- b. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。

结束值 0 表明 **deleteboard** 命令正常终止。

如果结束值非 0，或如果在执行 **deleteboard** 命令时显示错误信息，则表明 **deleteboard** 命令异常终止。根据错误信息，通过参考 "[C.1.2 deleteboard](#)" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

- c. 执行 **showboards** 命令检查 **PSB** 状态。

确认需要更换的 SPARC M10-4S 内的 PSB 是否处于 "Assigned" 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 "n"。

```
XSCF> showboards -p 0
PSB   PPAR-ID (LSB)  Assignment  Pwr   Conn  Conf  Test   Fault
-----
00-0  00(00)           Assigned    y     y     y     Passed Normal
01-0  00(01)           Assigned    n     n     n     Passed Normal
```

9. 执行 **replacefru** 命令来更换 **SPARC M10-4S**。

```
XSCF> replacefru
```

注 - 关于使用 **replacefru** 命令更换 SPARC M10-4S 的详情，请参见《Fujitsu M10-4/Fujitsu M10-4S/SPARC M10-4/SPARC M10-4S Service Manual》中的 "5.8 Releasing a SPARC M10-4/M10-4S FRU from the System with the **replacefru** Command" 和 "6.2 Incorporating a SPARC M10-4/M10-4S FRU into the System with the **replacefru** Command"。

10. 将 **PSB** 整合到物理分区。

- a. 执行 **showboards** 命令检查 **PSB** 状态。

确认已更换的 SPARC M10-4S 内的 PSB 是否处于 Assigned 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 "n"。

```
XSCF> showboards -p 0
PSB   PPAR-ID (LSB)  Assignment  Pwr   Conn  Conf  Test   Fault
-----
00-0  00(00)           Assigned    y     y     y     Passed Normal
01-0  00(01)           Assigned    n     n     n     Passed Normal
```

- b. 执行 **addboard -c configure** 命令将 **PSB** 整合到物理分区。

若需恢复初始逻辑域配置，请执行指定 **-m bind=resource** 选项的 **addboard -c**

configure 命令。

```
XSCF> addboard -c configure -m bind=resource -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
810.....840.....870.....900.....930.....960.....end
Connected PSB to PPAR.
Start configuring PSB to Logical Domains (LDoms) Manager.
[1800sec] 0.....end
Configured PSB to Logical Domains (LDoms) Manager.
Operation has completed.
```

注 - 如果执行 addboard 命令过程中出现错误信息，请参见 "C.1.1 addboard"，然后识别错误并采取应对措施。

- c. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。
结束值 0 表明 addboard 命令正常终止。
如果结束值非 0，或如果在执行 addboard 命令时显示错误信息，则表明 addboard 命令异常终止。根据错误信息，通过参考 "C.1.1 addboard" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

- d. 执行 **showboards** 命令检查 **PSB** 状态。
在已更换的 SPARC M10-4S 中的 PSB 已经被成功整合后，确认 [Conn] 和 [Conf] 栏是否都显示 "y"。

```
XSCF> showboards -p 0
```

PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

- 11. 检查逻辑域的运行状态。
 - a. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

- b. 执行 **ldm list-domain** 命令来确认 **SPARC M10-4S PSB (BB)** 整合后逻辑域的运行状态未发生变化。
若需检查逻辑域的运行状态，请确认 [STATE] 显示 "active"，并检查 [FLAGS] 内的字符串从左起第二个字符。显示的字符及其含义如下：

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 是 "active" 之外时的状态）

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	14G	0.0%	10h 30m
guest0	active	-n----	5100	32	32G	0.0%	2h 50s
guest1	active	-n----	5101	32	32G	0.0%	2h 30s
root-dom0	active	-n--v-	5000	24	24G	0.0%	3h 20s
root-dom1	inactive	-----		24	24G		

12. 重新启动 I/O 设备的使用。

a. 重新分配根合成体。

执行 `ldm bind-domain` 和 `ldm start-domain` 命令，以便启动在 `unbind` 状态下分配已更换的 SPARC M10-4S 中的根合成体的根域。

下面的示例启动在 `unbind` 状态下的根域 (`root-dom1`)，然后确认是否已经启动。

# ldm bind-domain root-dom1							
# ldm start-domain root-dom1							
LDom root-dom1 started							
# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	16	16G	0.2%	3h 8m
guest0	active	-n----	5100	32	32G	0.0%	3h 8m
guest1	active	-n----	5101	32	32G	0.0%	3h 8m
root-dom0	active	-n--v-	5000	24	24G	0.0%	3h 8m
root-dom1	active	-n--v-	5001	24	24G	7.3%	8s

执行 `ldm list-io` 命令确认物理 I/O 设备是否已经分配给刚刚启动的根域。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV

PCIE15	BUS	PCIE15	root-dom1	IOV
....				
/BB1/CMUL/NET0	PCIE	PCIE8		UNK
/BB1/CMUL/SASHBA	PCIE	PCIE8		UNK
/BB1/PCI0	PCIE	PCIE9	root-dom1	OCC
/BB1/PCI3	PCIE	PCIE10	root-dom1	OCC
/BB1/PCI4	PCIE	PCIE10	root-dom1	OCC
/BB1/PCI7	PCIE	PCIE11	root-dom1	OCC
/BB1/PCI8	PCIE	PCIE11	root-dom1	OCC
/BB1/CMUL/NET2	PCIE	PCIE12		UNK
/BB1/PCI1	PCIE	PCIE13	root-dom1	OCC
/BB1/PCI2	PCIE	PCIE13	root-dom1	OCC
/BB1/PCI5	PCIE	PCIE14	root-dom1	OCC
/BB1/PCI6	PCIE	PCIE14	root-dom1	OCC
/BB1/PCI9	PCIE	PCIE15	root-dom1	OCC
/BB1/PCI10	PCIE	PCIE15	root-dom1	OCC

b. 将虚拟 I/O 设备从根域添加到来宾域。

执行 `ldm add-vdisk` 和 `ldm add-vnet` 命令，将支持已经启动的根域的虚拟 I/O 服务的虚拟磁盘 (vdisk) 和虚拟网络设备 (vnet) 添加到各个来宾域。

下面的示例显示添加使用 BB#01 根域 (root-dom1) 的虚拟 I/O 服务的虚拟磁盘 (vdisk10) 和虚拟网络设备 (vnet10) 的命令的执行。

```
# ldm add-vdisk id=1 vdisk10 vol110@vds1 guest0
# ldm add-vnet id=1 vnet10 vsw10 guest0
```

注 - 若需再次添加虚拟 I/O 设备，有必要提前指定已分配的 ID。您可以在虚拟 I/O 设备被移除之前，在被使用的状态下，从 `ldm list -l` 命令的执行结果检查 ID。

对来宾域 (guest1) 执行相同的添加操作。

c. 将分配给来宾域的虚拟 I/O 设备整合到冗余配置中。

启动分配 BB#1 中的根合成体的根域 (root-dom1) 后，对应各来宾域的虚拟 I/O 设备服务也被启动。

登录各来宾域，然后从以前被取消的 root-dom1 中将虚拟 I/O 设备整合到冗余配置。有关如何使用冗余配置软件的详情，请参见该冗余配置软件的文档。

下面的示例介绍将虚拟网络设备 (vnet1) 整合到 IPMP 配置的示例。有关命令的详情，请参见 Oracle Solaris 的手册。

首先登录来宾域 (guest0)。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART     16      14G       0.0%    4h 17m
guest0        active    -n----   5100     32      32G       0.0%    1h 13m
guest1        active    -n----   5101     32      32G       0.0%    1h 4m
root-dom0     active    -n--v-   5000     24      24G       0.0%    1h 47m
root-dom1     active    -n--v-   5001     24      24G       0.0%    1h 19m
# telnet localhost 5100
```

```
....
guest0#
```

执行 `dladm show-phys` 命令检查虚拟网络接口 (vnet1) 和网络接口的名称 (net1) 之间的映射。

```
guest0# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      up        0      unknown  vnet0
net1          Ethernet      up        0      unknown  vnet1
```

执行 `ipadm create-ip` 命令、`ipadm set-ifprop` 命令以及 `ipadm add-ipmp` 命令将 net1 注册为 ipmp0 的备用设备。

```
guest0# ipadm create-ip net1
guest0# ipadm set-ifprop -p standby=on -m ip net1
guest0# ipadm add-ipmp -i net1 ipmp0
```

执行 `ipmpstat -i` 命令确认配置 IPMP 的网络接口的 STATE 是否显示 ok。

```
guest0# ipmpstat -i
INTERFACE ACTIVE GROUP FLAGS   LINK PROBE    STATE
net0      yes   ipmp0 -smbM-- up    disabled ok
net1      no    ipmp0 -s---d- up    disabled ok
```

对另一个来宾域 (guest1)，执行相同的步骤。

13. 将控制域上的系统卷和 I/O 设备恢复到冗余配置。
 - a. 添加控制域的根合成体配置。

在以前从控制域中被移除的 BB#01 中添加根合成体。

执行 `ldm list-io` 命令检查未分配的根合成体。

下面的示例表明 PCIE8 和 PCIE12 未被分配给 BB1 设备的根合成体。

```
# ldm list-io
NAME                                TYPE    BUS      DOMAIN    STATUS
----                                -
PCIE0                              BUS     PCIE0    primary   IOV
PCIE1                              BUS     PCIE1    root-dom0 IOV
PCIE2                              BUS     PCIE2    root-dom0 IOV
PCIE3                              BUS     PCIE3    root-dom0 IOV
PCIE4                              BUS     PCIE4    primary   IOV
PCIE5                              BUS     PCIE5    root-dom0 IOV
PCIE6                              BUS     PCIE6    root-dom0 IOV
PCIE7                              BUS     PCIE7    root-dom0 IOV
PCIE8                              BUS     PCIE8
PCIE9                              BUS     PCIE9    root-dom1 IOV
PCIE10                             BUS     PCIE10    root-dom1 IOV
PCIE11                             BUS     PCIE11    root-dom1 IOV
PCIE12                             BUS     PCIE12
```

PCIE13 (省略)	BUS	PCIE13	root-dom1	IOV
----------------	-----	--------	-----------	-----

执行 `ldm add-io` 命令将 PCIE8 和 PCIE12 添加到主域。

```
# ldm add-io PCIE8 primary
# ldm add-io PCIE12 primary
```

执行 `ldm list-io` 命令确认 BB#01 中的根合成体已经添加至控制域。

```
# ldm list-io | grep primary
PCIE0          BUS      PCIE0      primary  IOV
PCIE4          BUS      PCIE4      primary  IOV
PCIE8          BUS      PCIE8      primary  IOV
PCIE12         BUS      PCIE12     primary  IOV
/BB0/CMUL/NET0 PCIE    PCIE0      primary  OCC
/BB0/CMUL/SASHBA PCIE    PCIE0      primary  OCC
/BB0/CMUL/NET2  PCIE    PCIE4      primary  OCC
(省略)
```

b. 将控制域中的系统卷设置为冗余配置。

在控制域中执行 `zpool status` 命令检查镜像配置状态。

下面的示例介绍如何配置控制域中系统卷的 ZFS 镜像功能。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 29.1M in 0h0m with 0 errors on Thu Jan 23 17:27:59 2014
config:
   NAME                                STATE      READ WRITE CKSUM
   rpool                               ONLINE     0     0     0
       c2t50000393E802CCE2d0s0        ONLINE     0     0     0
errors: No known data errors
```

执行 `zpool attach` 命令，以便将磁盘整合到镜像配置中。

```
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting.
#
```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。

使用 `zpool status` 命令确认同步处理 (resilver) 是否完成。

下面显示同步处理过程中显示的示例。

```
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
```

```

        continue to function in a degraded state.
action: Wait for the resilver to complete.
        Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Jan 27 15:55:47 2014
      21.1G scanned out of 70.6G at 120M/s, 0h7m to go
      21.0G resilvered, 29.84% done
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                               DEGRADED   0     0     0
        mirror-0                          DEGRADED   0     0     0
          c2t50000393E802CCE2d0s0        ONLINE    0     0     0
          c3t50000393A803B13Ed0s0        DEGRADED   0     0     0 (resilvering)
errors: No known data errors

```

同步处理完成后，将显示如下画面：

```

# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:
      NAME                                STATE      READ WRITE CKSUM
      rpool                               ONLINE     0     0     0
        mirror-0                          ONLINE     0     0     0
          c2t50000393E802CCE2d0s0        ONLINE    0     0     0
          c3t50000393A803B13Ed0s0        ONLINE    0     0     0
errors: No known data errors

```

如果您正在使用 BB#01 中的其他设备，则创建冗余配置或恢复使用那些设备。有关如何创建冗余配置或恢复设备使用的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

A.4

采用 XCP 2220 或更高版本新安装的系统从 1BB 至 2BB 配置的扩展

本节介绍在物理分区工作状态下，通过物理分区动态重新配置，将构成 1BB 配置的 SPARC M10-4S 扩展到 2BB 配置的环境构建示例。也介绍环境构建相关的步骤。使用一个示例进行说明。

然而，具有 1BB 配置的 SPARC M10-4S 假定系统配置已经完成，包括 XCP 固件和 Oracle Solaris 相关的软件条件和 XSCF 设置。需要系统配置以便使用物理分区动态重新配置。

本介绍也适用于 SPARC M12。

A.4.1 配置示例

本节将构成 1BB 配置的 SPARC M10-4S 通过物理分区动态重新配置扩展为 2BB 配置，并添加来宾域 guest1 和已经分配扩展的 SPARC M10-4S (BB#1) 的根合成体的根域 root-dom1。本节也给出了将各来宾域的虚拟 I/O 设置为冗余配置和使用内置硬盘镜像控制域的系统卷的配置示例。

图 A-4 1BB 配置示例（扩展前）

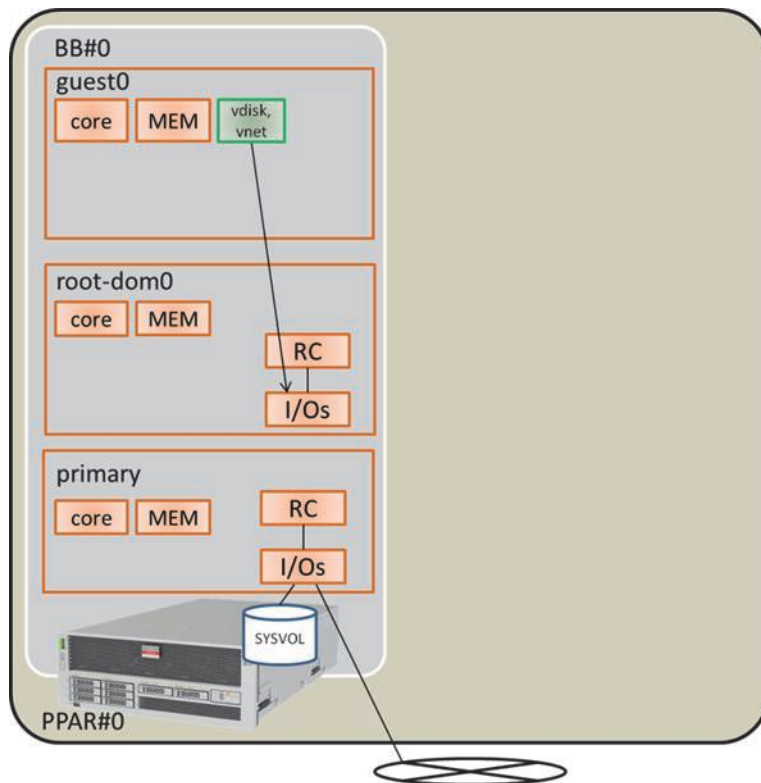


图 A-5 从 1BB 扩展到 2BB 后的配置示例

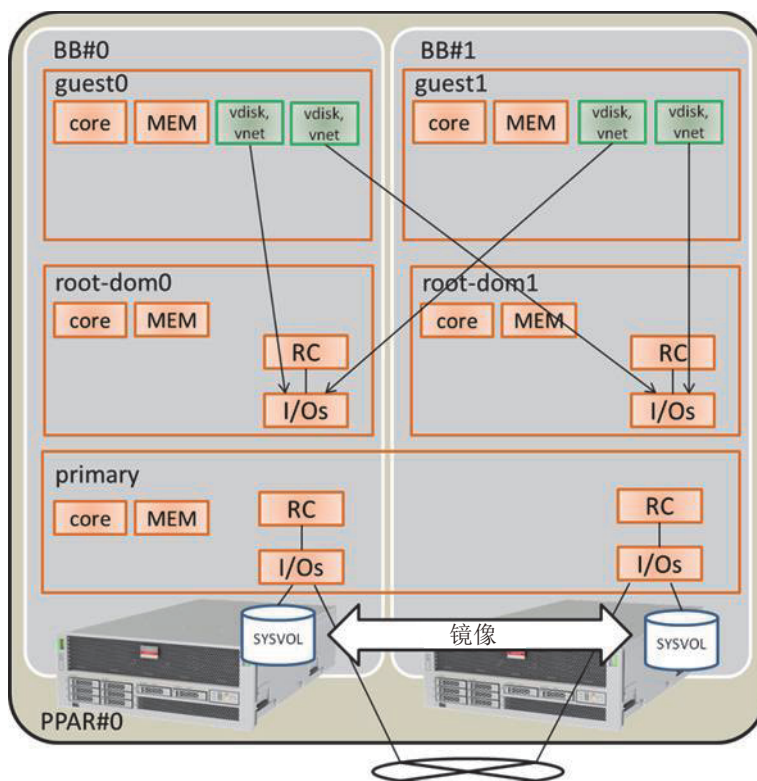


表 A-13 从 1BB 扩展到 2BB 的配置示例

逻辑域	扩展前			扩展后		
	CPU 内核	内存	I/O 配置	CPU 内核	内存	I/O 配置
控制域 (primary)	16	28 GB	On-board #0 (PCIE0, PCIE4)	32	56 GB	On-board #0 (PCIE0, PCIE4) On-board #1 (PCIE8, PCIE12)
guest0	32	64 GB	vdisk0, vnet0	32	64 GB	vdisk0, vdisk10 vnet0, vnet10
guest1	-	-	-	32	64 GB	vdisk1, vdisk11 vnet1, vnet11
root-dom0	16	32 GB	PCIE1, PCIE2, PCIE3, PCIE5, PCIE6, PCIE7, vds0, vsw0	16	32 GB	PCIE1, PCIE2, PCIE3, PCIE5, PCIE6, PCIE7, vds0, vsw0
root-dom1	-	-	-	16	32 GB	PCIE9, PCIE10, PCIE11, PCIE13, PCIE14, PCIE15, vds1, vsw1

表 A-13 从 1BB 扩展到 2BB 的配置示例 (续)

逻辑域	扩展前			扩展后		
	CPU 内核	内存	I/O 配置	CPU 内核	内存	I/O 配置
未分配的资源	0	约 2 GB(*1)	-	0	约 4.75 GB(*1)	-
总数	64	128 GB	-	128	256 GB	-

*1 在各个 SPARC M10-4S 上约 2 GB 或 1.25 GB 的内存区域被分配给虚拟机监控程序。因此，可以分配给逻辑域的内存资源比实际安装的内存容量小。

A.4.2 扩展步骤（适用于 Oracle VM Server for SPARC 3.1.x）

下面显示根据 "图 A-5 从 1BB 扩展到 2BB 后的配置示例" 中给出的配置示例，将具有 1BB 配置的 SPARC M10-4S 通过动态重新配置扩展为 2BB 配置。

1. 登录主 **XSCF**。
执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。
如果正在使用备用 XSCF，请再次登录主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

2. 执行 `showsscp` 命令检查内部服务处理器通信协议 (SSCP) 的 IP 地址是默认值，还是用户指定值。

```
XSCF> showsscp
```

当 IP 地址是默认值，并且您正在使用需要扩展的 SPARC M10-4S 的 IP 地址默认值时，请转到下一步。

若需设置用户值，请使用 `setsscp` 命令设置 IP 地址。然后，使用 `applynetwork` 命令来应用和检查需要扩展的 SPARC M10-4S 的 SSCP 的 IP 地址。随后，执行 `rebootxscf` 命令完成设置，然后转到下一步。有关步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.6 应用网络设置"。

3. 扩展 **SPARC M10-4S**。
 - a. 按照信息，执行 `addfru` 命令来扩展 **SPARC M10-4S**。

注 - 如果尚未设置 SSCP 的 IP 地址，执行 `addfru` 命令将导致错误。

注 - 如果执行 `addfru` 命令，需要扩展的 SPARC M10-4S 的固件版本将自动设置为主 XSCF 正在运行的 SPARC M10-4S 的版本。在此示例中，需要扩展的 SPARC M10-4S (BB-ID 1) 的固件版本将自动设置为主 XSCF 正在运行的 SPARC M10-4S (BB-ID 0) 的版本。

下面是扩展 BB#1 的示例。

```
XSCF> addfru
-----
Maintenance/Addition Menu
Please select the chassis including added FRU.
No. FRU                      Status
-----
 1  /BB#0                     Normal
 2  /BB#1                     Unmount
 3  /BB#2                     Unmount
 4  /BB#3                     Unmount
-----

Select [1-16|c:cancel] :2
Maintenance/Addition Menu
Please select the BB or a type of FRU to be added.
1. BB itself
2. PSU (Power Supply Unit)
-----

Select [1,2|c:cancel] :1
Maintenance/Addition Menu
Please select a FRU to be added.
No. FRU                      Status
-----
 1  /BB#1                     Unmount
Select [1|b:back] :1
```

- b. 将需要扩展的 **SPARC M10-4S** 安装到机架。
在执行上述提到的 addfru 命令时，如果出现 "After the added device is connected with the system, please turn on the breaker of the BB#1." 消息，则将需要扩展的 SPARC M10-4S 安装在机架上。
有关安装步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "3.4.1 在机架中安装 SPARC M10-4S"。
- c. 设置需要扩展的 **SPARC M10-4S** 的识别 ID (BB-ID)。
请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.1 设置标识机柜的 ID (BB-ID)"。
- d. 连接接线。
在现有的 SPARC M10-4S 和扩展的 BB 之间连接接线。关于连接线缆路由图和线缆列表，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "附录 B 模块化架构配置上的线缆连接信息"。有关机箱之间直接连接的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.2 连接线缆（机柜之间直接连接）"。
- e. 连接 **XSCF BB** 控制线。
通过 XSCF BB 控制线，连接现有的 SPARC M10-4S 和扩展的 SPARC M10-4S。关于 XSCF BB 控制线的路由图和线缆列表，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "附录 B 模块化架构配置上的线缆连接信息"。有关机箱之间直接连接的详情，请参见《Fujitsu M10-4S/SPARC

M10-4S 安装指南》中的 "4.2 连接线缆（机柜之间直接连接）"。

f. 连接 **XSCF DUAL** 控制线。

通过 XSCF DUAL 控制线，连接现有的 SPARC M10-4S 和扩展的 SPARC M10-4S。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.2 连接线缆（机柜之间直接连接）"。

g. 连接串行线或 **LAN** 线。

将串行线连接到需要扩展的 SPARC M10-4S 的 XSCF 串行端口。而且，将线缆连接到 XSCF-LAN 的各 LAN 端口、GbE 端口和 PCIe 卡。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "5.1 将线缆连接到 SPARC M10-4S"。

h. 连接输入电源到需要扩展的 **SPARC M10-4S**。

连接需要扩展的 SPARC M10-4S 的电源线到输入电源。

i. 在 **addfru** 命令的输入画面中输入 **"f"**。

在步骤 a. 所执行的 addfru 命令的输入画面中输入 "f"，以便执行 SPARC M10-4S 扩展。

```
2) Please select[f:finish] :f

Waiting for BB#1 to enter install state.
[This operation may take up to 20 minute(s)] (progress scale
  reported in seconds)
0..... 30..... done

Waiting for BB#1 to enter ready state.
[This operation may take up to 45 minute(s)] (progress scale
  reported in seconds)
0..... 30..... 60... done

Do you want to start to diagnose BB#1? [s:start|c:cancel] :
```

j. 跳过需要扩展的 **SPARC M10-4S** 的诊断，以便退出 **addfru** 命令。

在主 XSCF 的 addfru 命令的输入画面输入 "c"，然后跳过需要扩展的 SPARC M10-4S 的诊断处理。如果出现 "The addition of BB#1 has completed." 信息，则输入 "f"，然后输入 "c"，以便退出 addfru 命令。

```
Do you want to start to diagnose BB#1? [s:start|c:cancel] :c

Diagnostic tests are about to be skipped.
Running diagnostic tests are strongly recommended before using BB#1.
Are you sure you want to skip testing? [y:yes|n:no] :y

-----
Maintenance/Addition Menu Status of the added FRU.

FRU                               Status
-----
/BB#1                             Normal
```

```

-----
[Warning:007]
Running diagnostic tests on BB#1 is strongly recommended after addfru has
completed.
The addition of BB#1 has completed.[f:finish] :f
-----

Maintenance/Addition Menu
Please select the chassis including added FRU.

No.  FRU                      Status
-----
1    /BB#0                    Normal
2    /BB#1                    Normal
-----

Select [1,2|c:cancel] :c

```

4. 诊断已扩展的 **SPARC M10-4S**。

a. 执行 **testsb** 命令来实施已扩展的 **SPARC M10-4S** 的诊断测试。

执行 **testsb** 命令实施诊断测试。指定已扩展的 SPARC M10-4S 的物理系统板 (PSB) 编号，并检查初始诊断和连接 I/O。

```

XSCF> testsb -v -p -s -y 01-0
Initial diagnosis is about to start, Continue?[y|n] :y
PSB#01-0 power on sequence started.

```

如果显示错误，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "A.2.4 检查诊断结果"。

b. 执行 **diagxbu** 命令来实施接线诊断测试。

执行 **diagxbu** 命令实施目标物理分区和已扩展的 SPARC M10-4S 的接线诊断测试。指定已扩展的 SPARC M10-4S 的识别 ID (BB-ID) 和目标的物理分区 ID (PPAR-ID)。

```

XSCF> diagxbu -y -b 01 -p 00
XBU diagnosis is about to start, Continue?[y|n] :y
Power on sequence started. [7200sec]
0..... 30..... 60..end
XBU diagnosis started. [7200sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240...../
270.....300.....330.....360.....390.....420.....450.....480.....510.....\
540.....570.....600.....630.....660.....690.....720.....750.....780.....\
810.....840.....870.....900.....930..end
completed.
Power off sequence started. [1200sec]
0..... 30..... 60..... 90.....120.....150.....180.end
completed.

```

c. 执行 **showlogs error** 命令检查是否显示错误。

```
XSCF> showlogs error
```

如果显示错误，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "A.2.4 检查诊断结果"。

d. 执行 **showhardconf** 命令来检查已扩展的 **SPARC M10-4S** 的配置和状态。

执行 **showhardconf** 命令检查已扩展的 SPARC M10-4S 的硬件配置（CPU、内存或类似物），然后检查各单元的 Status 是否是 Normal。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Service;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;
+ Freq:3.000 GHz; Type:0x10;
+ Core:16; Strand:2;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
:
```

5. 设置已扩展的 **SPARC M10-4S** 的 **XSCF** 网络。

设置已扩展的 SPARC M10-4S 的 XSCF 网络。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.2 设置以太网 (XSCF-LAN) IP 地址" 和 "7.5.3 设置接管 IP 地址"。进行此设置后，执行 **applynetwork** 命令来应用和检查该设置。随后，执行 **rebootxscf** 命令完成设置，然后转到下一步。有关步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.6 应用网络设置"。

6. 设置已扩展的 **SPARC M10-4S** 的内存的镜像模式。

如果您未正在使用内存镜像模式，将不需要此步骤。

有关内存镜像模式的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "14.1 配置内存镜像"。

a. 如果您正在进行已扩展的 **SPARC M10-4S** 中内存的镜像配置，则使用 **setupfru** 命令来设置内存镜像模式。

以下示例将 SPARC M10-4S 物理系统板 (PSB 01-0) 下的所有 CPU 置于内存镜像模式。

```
XSCF> setupfru -m y sb 01-0
```

b. 执行 **showfru** 命令来检查内存的镜像模式设置。

```
XSCF> showfru sb 01-0
Device Location Memory Mirror Mode
sb      01-0
  cpu 01-0-0    yes
  cpu 01-0-1    yes
  cpu 01-0-2    yes
  cpu 01-0-3    yes
```

7. 在物理分区配置信息中登录已扩展的 **SPARC M10-4S** 的系统板。

a. 执行 **showpcl** 命令检查物理分区配置信息。

```
XSCF> showpcl -p 0
PPAR-ID      LSB      PSB      Status
00
              00      00-0      Running
```

b. 执行 **setpcl** 命令在物理分区配置信息中登录系统板。

执行 **setpcl** 命令在内置目标的物理分区配置信息中登录扩展的 SPARC M10-4S 系统板。

下面的示例中，将物理系统板 (PSB) 01-0 映射到物理分区 0 的逻辑系统板 (LSB) 01。

```
XSCF> setpcl -p 0 -a 01=01-0
```

c. 执行 **showpcl** 命令检查物理分区配置信息。

```
XSCF> showpcl -p 0
PPAR-ID      LSB      PSB      Status
00
              00      00-0      Running
              01      01-0
```

8. 登录 **CPU** 激活密钥以便分配 **CPU** 内核资源。

a. 执行 **showcodusage** 命令检查 **CPU** 激活信息。

执行 **showcodusage** 命令，检查物理分区是否包含可分配的 CPU 内核资源。

如此处所示，该系统已安装 128 个 CPU 内核资源，已注册 128 个 CPU 激活，64 个 CPU 内核资源正在使用中，当前未在使用的 CPU 激活的数量是 64。

```
XSCF> showcodusage -p resource
Resource In Use Installed CoD Permitted Status
-----
```

```
PROC          64          128          128 OK: 64 cores available
```

Note:

Please confirm the value of the "In Use" by the ldm command of Oracle VM Server for SPARC.

The XSCF may take up to 20 minutes to reflect the "In Use" of logical domains.

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。有关如何添加 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "5.3 添加 CPU 内核资源"。

b. 执行 `showcod` 命令检查物理分区的 CPU 内核资源信息。

执行 `showcod` 命令检查分配给物理分区的 CPU 内核资源的信息。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 64
```

如果已分配的资源不足，执行 `setcod` 命令将 CPU 内核资源分配给物理分区。

以下是将 64 个 CPU 内核资源添加到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c add 64
PROC Permits assigned for PPAR 0 : 64 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 - XCP 2250 和更早版本的 XSCF 固件不支持 `-c add`、`-c delete` 和 `-c set` 选项。按如下所示指定 `setcod` 命令的选项可交互式添加和删除 CPU 内核资源。

```
XSCF> setcod -s cpu
```

当 CPU 内核资源已经被分配时，再次执行 `showcod` 命令检查已分配给物理分区的 CPU 内核资源的信息。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 128
```

9. 检查逻辑域的运行状态。

a. 执行 `console` 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

b. 执行 `ldm list-domain` 命令检查逻辑域的运行状态。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义：

- "n": Oracle Solaris 正在运行
- "t": OpenBoot PROM 状态
- "-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

下面的示例表明控制域、一个根域和一个来宾域正在运行。

检查所有的域是处于表明 Oracle Solaris 处于运行状态的 "active"，还是 "inactive" 状态。如果存在处于 OpenBoot PROM 状态或绑定状态的域，则物理分区的动态重新配置可能失败。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	32	28G	0.0%	1h 33m
guest0	active	-n----	5100	64	64G	3.1%	2s
root-dom0	active	-n--v-	5000	32	32G	3.1%	17m

10. 将已扩展的 SPARC M10-4 的系统板 (PSB<BB>) 整合到物理分区。

a. 返回 XSCF shell，然后执行 showboards 命令检查 PSB 的状态。

执行 showboards 命令检查已扩展的 SPARC M10-4S 的 PSB (BB) 的状态是否是 "SP"（系统板池）。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal

b. 执行 addboard 命令将 PSB 整合到物理分区。

执行 addboard -c configure 命令将 PSB 整合到物理分区。

XSCF> addboard -c configure -p 0 01-0	
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y n] :y	
Start connecting PSB to PPAR. [3600sec]	
0.....30.....60.....90.....120.....150.....180.....210.....240.....	
270.....300.....330.....360.....390.....420.....450.....480.....510.....	
540.....570.....600.....630.....660.....690.....720.....750.....780.....	
810.....840.....870.....900.....930.....960.....end	
Connected PSB to PPAR.	
Start configuring PSB to Logical Domains (LDoms) Manager.	
[1800sec] 0.....end	
Configured PSB to Logical Domains (LDoms) Manager.	
Operation has completed.	

注 - 如果执行 addboard 命令过程中出现错误信息，请参见 "C.1.1 addboard"，然后识别错误并采取应对措施。

- c. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。

结束值 0 表明 **addboard** 命令正常终止。

如果结束值非 0，或如果在执行 **addboard** 命令时显示错误信息，则表明 **addboard** 命令异常终止。根据错误信息，通过参考 "[C.1.1 addboard](#)" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

- 执行 **showboards** 命令检查 **PSB** 状态。

确认已添加的 SPARC M10-4S 内的 PSB 是否处于 **Assigned** 状态，以及 **Pwr**、**Conn** 和 **Conf** 栏是否全显示 "y"。

```
XSCF> showboards -p 0
```

PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

11. 检查逻辑域的运行状态。

- a. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

- b. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。

如需查看逻辑域运行状态，请查看 **[STATE]** 和 **[FLAGS]** 的组合。如果 **[STATE]** 显示 "active"，**[FLAGS]** 中的字符串左起第二个字符具有以下含义：

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 **[STATE]** 中除 "active" 之外的状态）

下面的示例表明控制域、一个根域和一个来宾域正在运行。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	32	28G	0.0%	2h 3m
guest0	active	-n----	5100	64	64G	3.1%	33m
root-dom0	active	-n--v-	5000	32	32G	3.1%	47m

12. 变更现有逻辑域的配置，然后添加新逻辑域。

已添加的 SPARC M10-4S 的资源被分配给现有的逻辑域或新配置的逻辑域。下面的示例根据配置示例，创建镜像配置。换言之，添加控制域的 CPU 内核和内存，然后添加来宾域 **guest1** 和根域 **root-dom1**，以便创建各来宾域的虚拟 I/O 的冗余配置。此外，使用内置硬盘，创建控制域的系统卷的镜像配置。

- a. 执行 **ldm list-devices** 命令检查已添加的硬件资源的状态。

```
# ldm list-devices
CORE
  ID      %FREE   CPUSSET
  128     100    (256, 257)
  132     100    (264, 265)
  136     100    (272, 273)
  140     100    (280, 281)
....
MEMORY
  PA              SIZE          BOUND
  0x7000000000000  32G
  0x7200000000000  32G
  0x7400000000000  32G
  0x7600500000000  31488M
....
```

b. 执行 **ldm add-core** 命令和 **ldm add-memory** 命令来添加控制域的 **CPU** 内核和内存资源。

执行 **ldm add-core** 和 **ldm add-memory** 命令添加控制域的 CPU 内核和内存。通过执行 **ldm list-domain** 命令，检查是否已经添加 CPU 内核和内存。

```
# ldm add-core 16 primary
# ldm add-memory 38G primary
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    64      56G       0.0%    2h 13m
guest0        active    -n----   5100    64      64G       0.1%    43m
root-dom0     active    -n--v-   5000    32      32G       0.1%    57m
```

c. 创建可分配 **CPU** 内核和内存资源的新逻辑域。

本示例添加可分配 CPU 内核和内存的根域 (root-dom1) 和来宾域 (guest1)。然而，假设当配置域时，首先使用 **ldm set-core** 命令设置 CPU 内核数量，然后使用 **ldm set-memory** 命令设置需要分配的内存的容量。在这种情况下，分配给逻辑域的内存块不容易分散（碎片），在释放模块化架构时，考虑内存配置变得容易。

下面的示例显示了命令的执行。

```
# ldm add-domain root-dom1
# ldm add-domain guest1
# ldm set-core 16 root-dom1
# ldm set-core 32 guest1
# ldm set-memory 32G root-dom1
# ldm set-memory 64G guest1
```

注 - 若需在 Oracle VM Server for SPARC 的版本早于 3.2 时启用物理分区动态重新配置，请将需要分配给各逻辑域的内存容量设置为（CPU 内核数量 x 256 MB）的倍数。

d. 将根合成体分配给根域。

下面的示例将添加的 SPARC M10-4S 的根合成体分配给根域 (root-dom1)。
执行 `ldm list-io` 命令查看需要添加的根合成体。
NAME 以 `/BB1` 开始的设备在 BB-ID 1 上，该设备的 BUS 中显示的字符串表示根合成体。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10		
PCIE11	BUS	PCIE11		
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13		
PCIE14	BUS	PCIE14		
PCIE15	BUS	PCIE15		
....				
/BB1/CMUL/NET0	PCIE	PCIE8		
/BB1/CMUL/SASHBA	PCIE	PCIE8		
/BB1/PCI0	PCIE	PCIE9		
/BB1/PCI3	PCIE	PCIE10		
/BB1/PCI4	PCIE	PCIE10		
/BB1/PCI7	PCIE	PCIE11		
/BB1/PCI8	PCIE	PCIE11		
/BB1/CMUL/NET2	PCIE	PCIE12		
/BB1/PCI1	PCIE	PCIE13		
/BB1/PCI2	PCIE	PCIE13		
/BB1/PCI5	PCIE	PCIE14		
/BB1/PCI6	PCIE	PCIE14		
/BB1/PCI9	PCIE	PCIE15		
/BB1/PCI10	PCIE	PCIE15		

执行 `ldm add-io` 命令将根合成体添加到根域 (root-dom1)。然后，执行 `ldm list-io` 命令来实施检查。
下面的示例将除上述板载设备 (`/BB1/CMUL/NET0`、`/BB1/CMUL/SASHBA` 和 `/BB1/CMUL/NET2`) 的根合成体 (`PCIE8` 和 `PCIE12`) 之外的 BB1 根合成体 (`PCIE9`、`PCIE10`、`PCIE11`、`PCIE13`、`PCIE14` 和 `PCIE15`) 添加到 `root-dom1`。

```
# ldm add-io PCIE9 root-dom1
# ldm add-io PCIE10 root-dom1
....
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8	root-dom1	IOV
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV

e. 分配根域的控制台设备，然后将其启动。

使用 `ldm set-vconsole` 命令将控制台设备分配给根域 (root-dom1)，然后使用 `ldm bind-domain` 和 `ldm start-domain` 命令将其启动。

```
# ldm set-vconsole port=5001 vcc0 root-dom1
# ldm bind-domain root-dom1
# ldm start-domain root-dom1
LDom root-dom1 started
# telnet localhost 5001
....
```

f. 在根域中安装 **Oracle Solaris**。

在已启动的根域中安装 Oracle Solaris。有关如何安装 Oracle Solaris 的详情，请参见相关 Oracle Solaris 的文档。

g. 检查需要分配给根域的虚拟服务的物理设备。

安装 Oracle Solaris 后，登录根域，然后执行 `format` 命令来检查需要分配给虚拟磁盘服务的设备的名称。在这种情况下，执行 `ldm list-io -l` 命令显示已分配有 PCIe 端点的设备路径。因此，您可以通过将该设备路径的名称与使用 `format` 命令显示的磁盘设备路径的名称进行比较来检查该磁盘所连接的路径。

```
# ldm list-io -l
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
....
/BB1/PCI0                           PCIE  PCIE9  root-dom1OCC
[pci@8900/pci@4/pci@0/pci@0]
/BB1/PCI3                           PCIE  PCIE10 root-dom1OCC
[pci@8a00/pci@4/pci@0/pci@0]
/BB1/PCI4                           PCIE  PCIE10 root-dom1OCC
[pci@8a00/pci@4/pci@0/pci@8]....
# telnet localhost 5000
```

```

....
root-dom1# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c3t500000E01BDA70B2d0 <FUJITSU-MBB2147RC-3703 cyl 14087 alt 2 hd 24
    sec 848>
      / pci@8900/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w500000e01bda70b2,0
  1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
    sec 625>
      /pci@8a00/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
....
Specify disk (enter its number): ^C

```

随后，执行 `dladm show-phys -L` 命令查看分配给根域的以太网接口名和物理位置。

```

root-dom0# dladm show-phys -L
LINK          DEVICE          LOC
net0          igb0           /BB1/PCIO
net1          igb1           /BB1/PCIO
net2          igb2           /BB1/PCIO
net3          igb3           /BB1/PCIO

```

h. 将虚拟 I/O 服务分配给根域。

返回控制域，以便将虚拟 I/O 服务添加到根域 (root-dom1)。有关 I/O 设备设置的详情，请参见 Oracle 公司提供的《Oracle VM Server for SPARC Administration Guide》。

下面的示例将虚拟磁盘服务 (vds1) 和虚拟交换机网络接口 (vsw10和vsw11) 添加到已添加的根域 (root-dom1)。

```

# ldm add-vdiskserver vds1 root-dom1
# ldm add-vdiskserverdevice /dev/dsk/cXtXXXXXXXXXXXXXXXXdXsX vol0@vds1
# ldm add-vdiskserverdevice /dev/dsk/cXtXXXXXXXXXXXXXXXXdXsX vol1@vds1
# ldm add-vswitch net-dev=net1 vsw10 root-dom1
# ldm add-vswitch net-dev=net2 vsw11 root-dom1

```

i. 将虚拟 I/O 设备分配给来宾域。

将使用已添加虚拟 I/O 服务的虚拟 I/O 设备添加到各来宾域。

下面的示例将使用 root-dom1 的虚拟 I/O 服务的虚拟 I/O 设备 (vdisk10, vnet10) 添加到现有的来宾域 (guest0)。也将使用 root-dom0 和 root-dom1 的虚拟 I/O 服务的虚拟 I/O 设备 (vdisk1、vdisk11、vnet1 和 vnet11) 以及虚拟控制台 (vcons) 添加到需要新添加的来宾域 (guest1)。

```

# ldm add-vdisk vdisk10 vol0@vds1 guest0
# ldm add-vnet vnet10 vsw10 guest0
# ldm add-vdisk vdisk1 vol1@vds0 guest1
# ldm add-vdisk vdisk11 vol11@vds1 guest1
# ldm add-vnet vnet1 vsw1 guest1

```

```
# ldm add-vnet vnet11 vsw11 guest1
# ldm set-vconsole port=5101 guest1
```

j.启动添加的来宾域。

执行 `ldm bind-domain` 和 `ldm start-domain` 命令启动添加的来宾域 (guest1)。

```
# ldm bind-domain guest1
# ldm start-domain guest1
LDom guest1 started
```

k.在已添加的来宾域中安装 **Oracle Solaris**。

在已添加的来宾域中安装 Oracle Solaris。有关如何安装 Oracle Solaris 的详情，请参见相关 Oracle Solaris 的文档。

l.创建各来宾域的 I/O 的冗余配置。

变更已添加到各来宾域的虚拟 I/O 设备，以便创建冗余配置。有关如何设置冗余配置的详情，请参见各冗余配置软件相关的文档。

m. 创建控制域系统卷的冗余配置。

本示例将已添加的 SPARC M10-4S 的板载设备分配给控制域，以便创建冗余配置。将控制域切换到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

将已添加的 SPARC M10-4S 的根合成体 (PCIE8 和 PCIE12) 添加到控制域，然后重新启动 Oracle Solaris。

```
# ldm add-io PCIE8 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm add-io PCIE12 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# shutdown -i6 -g0 -y
....
```

重新启动 Oracle Solaris 后，创建系统卷的冗余配置。下面的示例使用 `format` 命令检查磁盘，以便使用 ZFS 创建镜像配置。有关如何设置冗余配置的详情，请参见各冗余配置软件相关的文档。

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
    0. c2t50000393E802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8000/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
        /dev/chassis/FUJITSU-BBEXP.500000e0e06d027f/0123_HDD00/disk
    1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8800/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
        /dev/chassis/FUJITSU-BBEXP.500000e0e06d243f/022U_HDD01/disk
Specify disk (enter its number): ^C
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:

        NAME                                STATE      READ WRITE CKSUM
        rpool                                ONLINE     0     0     0
            c2t50000393E802CCE2d0s0          ONLINE     0     0     0

errors: No known data errors
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting
```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。此时，执行 `zpool status` 命令来检查同步处理 (resilver) 是否已经完成。下面的示例表明，同步处理正在进行中。

```
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered. The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
        Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Jan 27 15:55:47 2014
      21.1G scanned out of 70.6G at 120M/s, 0h7m to go
      21.0G resilvered, 29.84% done
config:

        NAME                                STATE      READ WRITE CKSUM
        rpool                                DEGRADED   0     0     0
            mirror-0
                c2t50000393E802CCE2d0s0          ONLINE     0     0     0
                c3t50000393A803B13Ed0s0          DEGRADED   0     0     0 (resilvering)

errors: No known data errors
```

同步处理完成后，将显示如下画面：

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:

    NAME                                STATE      READ  WRITE CKSUM
    rpool                                ONLINE     0     0     0
      mirror-0
        c2t50000393E802CCE2d0s0        ONLINE     0     0     0
        c3t50000393A803B13Ed0s0        ONLINE     0     0     0

errors: No known data errors
```

n. 检查并调整各逻辑域的内存配置。

执行 `ldm list-devices -a memory` 命令检查内存块的配置。

```
# ldm list-devices -a memory
MEMORY
    PA                                SIZE                                BOUND
    0x7000000000000000                32G                                root-dom1
    0x7200000000000000                32G                                guest1
    0x7400000000000000                32G                                guest1
    0x7600008000000000                1272M                               _sys_
    0x7600500000000000                16G                                primary
    0x7604500000000000                10G                                primary
    0x7606d00000000000                4864M                               _sys_
    0x7800000000000000                32G                                guest0
    0x7a00000000000000                32G                                guest0
    0x7c00000000000000                256M                               primary
    0x7c00100000000000                3584M                               root-dom0
    0x7c00f00000000000                24832M                              primary
    0x7c07000000000000                4G                                 root-dom0
    0x7e00008000000000                1272M                               _sys_
    0x7e00500000000000                512M                               _sys_
    0x7e00700000000000                256M                               _sys_
    0x7e00800000000000                25088M                              root-dom0
    0x7e06a00000000000                3584M                               primary
    0x7e07800000000000                2G                                 primary
```

假设分配给逻辑域的内存块和未分配的内存块被分割成小碎片。您可以通过取消绑定逻辑域，然后再次将其绑定来将零散的内存块集成成大的连续区域。此操作可很容易地添加或删除 SPARC M10-4S。

有关步骤的详情，请参见 ["A.2 对于从 XCP 2220 或更高版本的新安装，逻辑域配置无可用的硬件资源"](#) 的 ["A.2.2 物理分区配置步骤示例"](#) 中的步骤 11。

o. 保存逻辑域配置信息。

执行 `ldm list-domain` 命令检查配置的逻辑域是否已经启动。

# ldm list-domain							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	64%	2h 54m
guest0	active	-n----	5100	64	64G	42%	2h 54m
guest1	active	-n----	5101	64	64G	11%	2h 54m
root-dom0	active	-n--v-	5000	32	32G	20%	2h 54m
root-dom1	active	-n--v-	5001	32	32G	18%	2h 54m

执行 `ldm add-spconfig` 命令将逻辑域配置信息保存到 XSCF 。

```
# ldm add-spconfig ldm-set4
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3
ldm-set4  [current]
```

A.4.3 扩展步骤（当 PCIe 总线的动态分配可用时）

下面显示根据 "图 A-5 从 1BB 扩展到 2BB 后的配置示例" 中给出的配置示例，将具有 1BB 配置的 SPARC M10-4S 通过动态重新配置扩展为 2 BB 配置。此步骤适用于 PCIe 总线可动态分配的环境（Oracle VM Server for SPARC 3.2 或更高版本的 XCP 2240 或更高版本和 Oracle Solaris 11.2 SRU11.2.8.4.0 或更高版本的根域）。

本介绍也适用于 SPARC M12。

1. 登录主 **XSCF**。
 执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。
 如果正在使用备用 XSCF，请再次登录主 XSCF 。

```
XSCF> showbbstatus
BB#00 (Master)
```

2. 执行 `showsscp` 命令检查内部服务处理器通信协议 (SSCP) 的 IP 地址是默认值，还是用户指定值。

```
XSCF> showsscp
```

当 IP 地址是默认值，并且您正在使用需要扩展的 SPARC M10-4S 的 IP 地址默认值时，请转到下一步。

若需设置用户值，请使用 `setsscp` 命令设置 IP 地址。然后，使用 `applynetwork` 命令来应用和检查需要扩展的 SPARC M10-4S 的 SSCP 的 IP 地址。随后，执行 `rebootxscf` 命令完成设置，然后转到下一步。有关步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.6 应用网络设置"。

3. 扩展 **SPARC M10-4S**。

a. 按照信息，执行 **addfru** 命令来扩展 **SPARC M10-4S**。

注 - 如果尚未设置 SSCP 的 IP 地址，执行 **addfru** 命令将导致错误。

注 - 如果执行 **addfru** 命令，需要扩展的 SPARC M10-4S 的固件版本将自动设置为主 XSCF 正在运行的 SPARC M10-4S 的版本。在此示例中，需要扩展的 SPARC M10-4S (BB-ID 1) 的固件版本将自动设置为主 XSCF 正在运行的 SPARC M10-4S (BB-ID 0) 的版本。

下面是扩展 **BB#1** 的示例。

```
XSCF> addfru
-----
Maintenance/Addition Menu
Please select the chassis including added FRU.
No. FRU                               Status
-----
 1  /BB#0                             Normal
 2  /BB#1                             Unmount
 3  /BB#2                             Unmount
 4  /BB#3                             Unmount
-----
Select [1-16|c:cancel] :2
Maintenance/Addition Menu
Please select the BB or a type of FRU to be added.
1. BB itself
2. PSU (Power Supply Unit)
-----
Select [1,2|c:cancel] :1
Maintenance/Addition Menu
Please select a FRU to be added.
No. FRU                               Status
-----
 1  /BB#1                             Unmount
Select [1|b:back] :1
```

b. 将需要扩展的 **SPARC M10-4S** 安装到机架。

在执行上述提到的 **addfru** 命令时，如果出现 "After the added device is connected with the system, please turn on the breaker of the BB#1." 消息，则将需要扩展的 **SPARC M10-4S** 安装在机架上。

有关安装步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "3.4.1 在机架中安装 SPARC M10-4S"。

c. 设置需要扩展的 **SPARC M10-4S** 的识别 ID (BB-ID)。

请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.1 设置标识机柜的 ID (BB-ID)"。

d. 连接接线。

在现有的 **SPARC M10-4S** 和扩展的 **SPARC M10-4S** 之间连接接线。关于连接线缆路由图和线缆列表，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "附录 B 模块化架构配置上的线缆连接信息"。有关机箱之间直接连接的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.2 连接线

缆（机柜之间直接连接）”。

e. 连接 **XSCF BB** 控制线。

通过 XSCF BB 控制线，连接现有的 SPARC M10-4S 和扩展的 SPARC M10-4S。关于 XSCF BB 控制线的路由图和线缆列表，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的“附录 B 模块化架构配置上的线缆连接信息”。有关机箱之间直接连接的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的“4.2 连接线缆（机柜之间直接连接）”。

f. 连接 **XSCF DUAL** 控制线。

通过 XSCF DUAL 控制线，连接现有的 SPARC M10-4S 和扩展的 SPARC M10-4S。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的“4.2 连接线缆（机柜之间直接连接）”。

g. 连接串行线或 **LAN** 线。

将串行线连接到需要扩展的 SPARC M10-4S 的 XSCF 串行端口。而且，将线缆连接到 XSCF-LAN 的各 LAN 端口、GbE 端口和 PCIe 卡。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的“5.1 将线缆连接到 SPARC M10-4S”。

h. 连接输入电源到需要扩展的 **SPARC M10-4S**。

连接需要扩展的 SPARC M10-4S 的电源线到输入电源。

i. 在 **addfru** 命令的输入画面中输入 **"f"**。

在步骤 a. 所执行的 addfru 命令的输入画面中输入 "f"，以便执行 SPARC M10-4S 扩展。

```
2) Please select[f:finish] :f

Waiting for BB#1 to enter install state.
[This operation may take up to 20 minute(s)] (progress scale
reported in seconds)
0..... 30.... done

Waiting for BB#1 to enter ready state.
[This operation may take up to 45 minute(s)] (progress scale
reported in seconds)
0..... 30..... 60... done

Do you want to start to diagnose BB#1? [s:start|c:cancel] :
```

j. 跳过需要扩展的 **SPARC M10-4S** 的诊断，以便退出 **addfru** 命令。

在主 XSCF 的 addfru 命令的输入画面输入 "c"，然后跳过需要扩展的 SPARC M10-4S 的诊断处理。如果出现 "The addition of BB#1 has completed." 信息，则输入 "f"，然后输入 "c"，以便退出 addfru 命令。

```
Do you want to start to diagnose BB#1? [s:start|c:cancel] :c

Diagnostic tests are about to be skipped.
Running diagnostic tests are strongly recommended before using BB#1.
Are you sure you want to skip testing? [y:yes|n:no] :y
```

```
-----
Maintenance/Addition Menu Status of the added FRU.

FRU                               Status
-----
/BB#1                             Normal
-----

[Warning:007]
Running diagnostic tests on BB#1 is strongly recommended after addfru has
completed.
The addition of BB#1 has completed.[f:finish] :f

-----

Maintenance/Addition Menu
Please select the chassis including added FRU.

No. FRU                               Status
-----
1   /BB#0                             Normal
2   /BB#1                             Normal
-----

Select [1,2|c:cancel] :c
```

- 4. 诊断已扩展的 **SPARC M10-4S**。
 - a. 执行 **testsb** 命令来实施已扩展的 **SPARC M10-4S** 的诊断测试。

执行 **testsb** 命令实施诊断测试。指定已扩展的 SPARC M10-4S 的物理系统板 (PSB) 编号，并检查初始诊断和连接 I/O。

```
XSCF> testsb -v -p -s -y 01-0
Initial diagnosis is about to start, Continue?[y|n] :y
PSB#01-0 power on sequence started.
```

- 如果显示错误，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "A.2.4 检查诊断结果"。
- b. 执行 **diagxbu** 命令来实施接线诊断测试。

执行 **diagxbu** 命令实施目标物理分区和已扩展的 SPARC M10-4S 的接线诊断测试。指定已扩展的 SPARC M10-4S 的识别 ID (BB-ID) 和目标的物理分区 ID (PPAR-ID)。

```
XSCF> diagxbu -y -b 01 -p 00
XBU diagnosis is about to start, Continue?[y|n] :y
Power on sequence started. [7200sec]
0..... 30..... 60..end
XBU diagnosis started. [7200sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240...../
270.....300.....330.....360.....390.....420.....450.....480.....510.....\
540.....570.....600.....630.....660.....690.....720.....750.....780.....\
810.....840.....870.....900.....930..end
```

```
completed.  
Power off sequence started. [1200sec]  
0..... 30..... 60..... 90.....120.....150.....180.end  
completed.
```

c. 执行 **showlogs error** 命令检查是否显示错误。

```
XSCF> showlogs error
```

如果显示错误，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "A.2.4 检查诊断结果"。

d. 执行 **showhardconf** 命令来检查已扩展的 **SPARC M10-4S** 的配置和状态。

执行 **showhardconf** 命令检查已扩展的 SPARC M10-4S 的硬件配置（CPU、内存或类似物），然后检查各单元的 Status 是否是 Normal。

```
XSCF> showhardconf  
SPARC M10-4S;  
+ Serial: 2081230011; Operator_Panel_Switch:Service;  
+ System_Power:On; System_Phase:Cabinet Power On;  
Partition#0 PPAR_Status:Running;  
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;  
+ FRU-Part-Number: CA07361-D202 A1 ;  
+ Power_Supply_System: ;  
+ Memory_Size:256 GB;  
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;  
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;  
+ Memory_Size:128 GB; Type: A ;  
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;  
+ Freq:3.000 GHz; Type:0x10;  
+ Core:16; Strand:2;  
:  
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;  
+ FRU-Part-Number: CA20393-B50X A2 ;  
+ Power_Supply_System: ;  
+ Memory_Size:256 GB;  
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;  
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;  
+ Memory_Size:128 GB; Type: A ;  
:
```

5. 设置已扩展的 **SPARC M10-4S** 的 **XSCF** 网络。

a. 执行 **testsb** 命令来实施已扩展的 **SPARC M10-4S** 的诊断测试。

设置已扩展的 SPARC M10-4S 的 XSCF 网络。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.2 设置以太网 (XSCF-LAN) IP 地址" 和 "7.5.3 设置接管 IP 地址"。进行此设置后，执行 **applynetwork** 命令来应用和检查该设置。随后，执行 **rebootxscf** 命令完成设置，然后转到下一步。有关步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.6 应用网络设置"。

6. 设置已扩展的 **SPARC M10-4S** 的内存的镜像模式。

- 如果您未正在使用内存镜像模式，将不需要此步骤。
- 有关内存镜像模式的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "14.1 配置内存镜像"。
- a. 如果您正在进行已扩展的 **SPARC M10-4S** 中内存的镜像配置，则使用 **setupfru** 命令来设置内存镜像模式。
以下示例将 SPARC M10-4S 物理系统板 (PSB 01-0) 下 CPU 的所有内存置于内存镜像模式。

```
XSCF> setupfru -m y sb 01-0
```

- b. 执行 **showfru** 命令来检查内存的镜像模式设置。

```
XSCF> showfru sb 01-0
Device Location Memory Mirror Mode
sb      01-0
  cpu 01-0-0      yes
  cpu 01-0-1      yes
  cpu 01-0-2      yes
  cpu 01-0-3      yes
```

- 7. 在物理分区配置信息中登录已扩展的 **SPARC M10-4S** 的系统板。
 - a. 执行 **showpctl** 命令检查物理分区配置信息。

```
XSCF> showpctl -p 0
PPAR-ID      LSB      PSB      Status
00
              00      00-0      Running
```

- b. 执行 **setpctl** 命令在物理分区配置信息中登录系统板。
执行 **setpctl** 命令在内置目标的物理分区配置信息中登录扩展的 SPARC M10-4S 系统板。
下面的示例中，将物理系统板 (PSB) 01-0 映射到物理分区 0 的逻辑系统板 (LSB) 01。

```
XSCF> setpctl -p 0 -a 01=01-0
```

- c. 执行 **showpctl** 命令检查物理分区配置信息。

```
XSCF> showpctl -p 0
PPAR-ID      LSB      PSB      Status
00
              00      00-0      Running
              01      01-0
```

- 8. 登录 **CPU** 激活密钥以便分配 **CPU** 内核资源。

a. 执行 **showcodusage** 命令查看 CPU 内核资源信息。

执行 **showcodusage** 命令，检查物理分区是否包含可分配的 CPU 内核资源。
如此处所示，该系统已安装 128 个 CPU 内核资源，已注册 128 个 CPU 激活，
64 个 CPU 内核资源正在使用中，当前未在使用的 CPU 激活的数量是 64。

```
XSCF> showcodusage -p resource
Resource In Use Installed CoD Permitted Status
-----
PROC          64      128          128 OK: 64 cores available

Note:
  Please confirm the value of the "In Use" by the ldm command of Oracle VM
  Server for SPARC.

  The XSCF may take up to 20 minutes to reflect the "In Use" of logical domains.
```

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。有关如何添加 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "5.3 添加 CPU 内核资源"。

b. 执行 **showcod** 命令检查物理分区的 CPU 内核资源信息。

执行 **showcod** 命令检查分配给物理分区的 CPU 内核资源的信息。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 64
```

如果已分配的资源不足，执行 **setcod** 命令将 CPU 内核资源分配给物理分区。
下面的示例将 64 个 CPU 内核资源添加到物理分区。

```
XSCF> setcod -p 0 -s cpu -c add 64
PROC Permits assigned for PPAR 0 : 64 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 - XCP 2250 和更早版本的 XSCF 固件不支持 **-c add**、**-c delete** 和 **-c set** 选项。按如下所示指定 **setcod** 命令的选项可交互式添加和删除 CPU 内核资源。

```
XSCF> setcod -s cpu
```

当 CPU 内核资源已经被分配时，再次执行 **showcod** 命令检查已分配给物理分区的 CPU 内核资源的信息。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 128
```

9. 检查逻辑域的运行状态。
 - a. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

- b. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义。

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

下面的示例表明控制域、一个根域和一个来宾域正在运行。

检查所有的域是处于表明 Oracle Solaris 处于运行状态的 "active"，还是 "inactive" 状态。如果存在处于 OpenBoot PROM 状态或绑定状态的域，则物理分区的动态重新配置可能失败。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   UART    32     28G     0.0%  1h 33m
guest0        active    -n----   5100    64     64G     3.1%  2s
root-dom0     active    -n--v-   5000    32     32G     3.1%  17m
```

10. 将已扩展的 **SPARC M10-4** 的系统板 (**PSB<BB>**) 整合到物理分区。
 - a. 返回 **XSCF shell**，然后执行 **showboards** 命令检查 **PSB** 的状态。

执行 **showboards** 命令检查已扩展的 SPARC M10-4S 的 PSB 状态是否是 "SP"（系统板池）。

```
XSCF> showboards -p 0
PSB  PPAR-ID (LSB) Assignment  Pwr  Conn Conf Test  Fault
-----
00-0 00(00)      Assigned   y    y    y    Passed Normal
01-0 SP          Available  n    n    n    Passed Normal
```

- b. 执行 **addboard** 命令将 **PSB** 整合到物理分区。

执行 **addboard -c configure** 命令将 PSB 整合到物理分区。

```
XSCF> addboard -c configure -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
```

```
810.....840.....870.....900.....930.....960.....end
Connected PSB to PPAR.
Start configuring PSB to Logical Domains (LDoms) Manager.
[1800sec] 0.....end
Configured PSB to Logical Domains (LDoms) Manager.
Operation has completed.
```

注 - 如果执行 **addboard** 命令过程中出现错误信息，请参见 ["C.1.1 addboard"](#)，然后识别错误并采取应对措施。

c. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。

结束值 0 表明 **addboard** 命令正常终止。

如果结束值非 0，或如果在执行 **addboard** 命令时显示错误信息，则表明 **addboard** 命令异常终止。根据错误信息，通过参考 ["C.1.1 addboard"](#) 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

执行 **showboards** 命令检查 PSB 状态。

确认已添加的 SPARC M10-4S 内的 PSB 是否处于 Assigned 状态，以及 Pwr、Conn 和 Conf 栏是否全显示 "y"。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

11. 检查逻辑域的运行状态。

a. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p 0
```

b. 执行 **ldm list-domain** 命令检查逻辑域的运行状态。

如需查看逻辑域运行状态，请查看 [STATE] 和 [FLAGS] 的组合。如果 [STATE] 显示 "active"，[FLAGS] 中的字符串左起第二个字符具有以下含义。

"n": Oracle Solaris 正在运行

"t": OpenBoot PROM 状态

"-": 其他状态（包含 [STATE] 中除 "active" 之外的状态）

下面的示例表明控制域、一个根域和一个来宾域正在运行。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	32	28G	0.0%	2h 3m
guest0	active	-n----	5100	64	64G	3.1%	33m
root-dom0	active	-n--v-	5000	32	32G	3.1%	47m

12. 变更现有逻辑域的配置，然后添加新逻辑域。
- 已添加的 SPARC M10-4S 的资源被分配给现有的逻辑域或新配置的逻辑域。下面的示例根据配置示例，创建镜像配置。换言之，添加控制域的 CPU 内核和内存，然后添加来宾域 guest1 和根域 root-dom1，以便创建各来宾域的虚拟 I/O 的冗余配置。此外，使用内置硬盘，创建控制域的系统卷的镜像配置。

a. 执行 `ldm list-devices` 命令检查已添加的硬件资源的状态。

```
# ldm list-devices
```

CORE		
ID	%FREE	CPUSET
128	100	(256, 257)
132	100	(264, 265)
136	100	(272, 273)
140	100	(280, 281)
....		
MEMORY		
PA	SIZE	BOUND
0x70000000000000	32G	
0x72000000000000	32G	
0x74000000000000	32G	
0x76005000000000	31488M	
....		

- b. 执行 `ldm add-core` 命令和 `ldm add-memory` 命令来添加控制域的 CPU 内核和内存资源。

执行 `ldm add-core` 和 `ldm add-memory` 命令添加控制域的 CPU 内核和内存。通过执行 `ldm list-domain` 命令，检查是否已经添加 CPU 内核和内存。

```
# ldm add-core 16 primary
# ldm add-memory 56G primary
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	0.0%	2h 13m
guest0	active	-n----	5100	64	64G	0.1%	43m
root-dom0	active	-n--v-	5000	32	32G	0.1%	57m

- c. 创建可分配 CPU 内核和内存资源的新逻辑域。

本示例添加可分配 CPU 内核和内存的根域 (root-dom1) 和来宾域 (guest1)。然而，假设当配置域时，首先使用 `ldm set-core` 命令设置 CPU 内核数量，然后使用 `ldm set-memory` 命令设置需要分配的内存的容量。在这种情况下，分配给逻辑域的内存块不容易分散（碎片），在释放模块化架构时，考虑内存配置变得容易。

下面的示例显示了命令的执行。


```
# ldm add-domain root-dom1
# ldm add-domain guest1
# ldm set-core 16 root-dom1
# ldm set-core 32 guest1
# ldm set-memory 32G root-dom1
# ldm set-memory 64G guest1
```

注 - 若需在 Oracle VM Server for SPARC 的版本早于 3.2 时启用物理分区动态重新配置，请将需要分配给各逻辑域的内存容量设置为（CPU 内核数量 x 256 MB）的倍数。

d. 将根合成体分配给根域。

下面的示例将添加的 SPARC M10-4S 的根合成体分配给根域 (root-dom1)。

执行 `ldm list-io` 命令来检查需要添加到根域 (root-dom1) 的根合成体。

NAME 以 "/BB1" 开始的设备在 BB-ID 1 上，该设备的 BUS 中显示的字符串表示根合成体。

```
# ldm list-io
NAME                                TYPE    BUS      DOMAIN    STATUS
----                                -
PCIE0                              BUS     PCIE0    primary   IOV
PCIE1                              BUS     PCIE1    root-dom0 IOV
PCIE2                              BUS     PCIE2    root-dom0 IOV
PCIE3                              BUS     PCIE3    root-dom0 IOV
PCIE4                              BUS     PCIE4    primary   IOV
PCIE5                              BUS     PCIE5    root-dom0 IOV
PCIE6                              BUS     PCIE6    root-dom0 IOV
PCIE7                              BUS     PCIE7    root-dom0 IOV
PCIE8                              BUS     PCIE8
PCIE9                              BUS     PCIE9
PCIE10                             BUS     PCIE10
PCIE11                             BUS     PCIE11
PCIE12                             BUS     PCIE12
PCIE13                             BUS     PCIE13
PCIE14                             BUS     PCIE14
PCIE15                             BUS     PCIE15
...
/BB1/CMUL/NET0                     PCIE    PCIE8
/BB1/CMUL/SASHBA                    PCIE    PCIE8
/BB1/PCI0                           PCIE    PCIE9
/BB1/PCI3                           PCIE    PCIE10
/BB1/PCI4                           PCIE    PCIE10
/BB1/PCI7                           PCIE    PCIE11
/BB1/PCI8                           PCIE    PCIE11
/BB1/CMUL/NET2                      PCIE    PCIE12
/BB1/PCI1                           PCIE    PCIE13
/BB1/PCI2                           PCIE    PCIE13
/BB1/PCI5                           PCIE    PCIE14
/BB1/PCI6                           PCIE    PCIE14
/BB1/PCI9                           PCIE    PCIE15
/BB1/PCI10                          PCIE    PCIE15
```

执行 `ldm add-io` 命令将根合成体添加到根域 (`root-dom1`)。然后, 执行 `ldm list-io` 命令来实施检查。

下面的示例将除上述板载设备 (`/BB1/CMUL/NET0`、`/BB1/CMUL/SASHBA` 和 `/BB1/CMUL/NET2`) 的根合成体 (`PCIE8` 和 `PCIE12`) 之外的 `BB1` 根合成体 (`PCIE9`、`PCIE10`、`PCIE11`、`PCIE13`、`PCIE14` 和 `PCIE15`) 添加到 `root-dom1`。

```
# ldm add-io PCIE9 root-dom1
# ldm add-io PCIE10 root-dom1
....
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8		
PCIE9	BUS	PCIE9	root-dom1	IOV
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV

e. 分配根域的控制台设备, 然后将其启动。

使用 `ldm set-vconsole` 命令将控制台设备分配给根域 (`root-dom1`), 然后使用 `ldm bind-domain` 和 `ldm start-domain` 命令将其启动。

```
# ldm set-vconsole port=5001 vcc0 root-dom1
# ldm bind-domain root-dom1
# ldm start-domain root-dom1
LDom root-dom1 started
# telnet localhost 5001
....
```

f. 在根域中安装 **Oracle Solaris**。

在已启动的根域中安装 Oracle Solaris。有关如何安装 Oracle Solaris 的详情, 请参见相关 Oracle Solaris 的文档。

g. 检查需要分配给根域的虚拟服务的物理设备。

安装 Oracle Solaris 后, 登录根域, 然后执行 `format` 命令来检查需要分配给虚拟磁盘服务的设备的名称。

在登录根域显示与 `PCIe` 端点关联的设备路径之前, 在控制域上执行 `ldm list-io -l` 命令。因此, 您可以通过将该设备路径的名称与使用 `format` 命令显示的磁盘设备路径的名称进行比较来检查该磁盘所连接的路径。

```
# ldm list-io -l
NAME                                TYPE    BUS      DOMAIN    STATUS
----                                -
....
/BB1/PCIO                          PCIE    PCIE9    root-dom1OCC
[pci@8900/pci@4/pci@0/pci@0]
/BB1/PCI3                          PCIE    PCIE10   root-dom1OCC
[pci@8a00/pci@4/pci@0/pci@0]
/BB1/PCI4                          PCIE    PCIE10   root-dom1OCC
[pci@8a00/pci@4/pci@0/pci@8]....
# telnet localhost 5001
....
root-dom1# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
    0. c3t500000E01BDA70B2d0 <FUJITSU-MBB2147RC-3703 cyl 14087 alt 2 hd 24
sec 848>
        / pci@8900/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w500000e01bda70b2,0
    1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
        /pci@8a00/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
....
Specify disk (enter its number): ^C
```

随后，执行 `ldadm show-phys -L` 命令查看分配给根域的以太网接口名和物理位置。

```
root-dom1# dladm show-phys -L
LINK          DEVICE          LOC
net0          igb0            /BB1/PCIO
net1          igb1            /BB1/PCIO
net2          igb2            /BB1/PCIO
net3          igb3            /BB1/PCIO
```

h. 将虚拟 I/O 服务分配给根域。

返回控制域，以便将虚拟 I/O 服务添加到根域 (root-dom1)。有关 I/O 设备设置的详情，请参见 Oracle 公司提供的《Oracle VM Server for SPARC Administration Guide》。

下面的示例将虚拟磁盘服务 (vds1) 和虚拟交换机网络接口 (vsw10和vsw11) 添加到已添加的根域 (root-dom1)。

```
# ldm add-vdiskserver vds1 root-dom1
# ldm add-vdiskserverdevice /dev/dsk/cXtXXXXXXXXXXXXdXsX vol0@vds1
# ldm add-vdiskserverdevice /dev/dsk/cXtXXXXXXXXXXXXdXsX vol1@vds1
# ldm add-vswitch net-dev=net1 vsw10 root-dom1
# ldm add-vswitch net-dev=net2 vsw11 root-dom1
```

i. 将虚拟 I/O 设备分配给来宾域。

将使用已添加虚拟 I/O 服务的虚拟 I/O 设备添加到各来宾域。

下面的示例将使用 root-dom1 的虚拟 I/O 服务的虚拟 I/O 设备 (vdisk10, vnet10) 添加到现有的来宾域 (guest0)。也将使用 root-dom0 和 root-dom1 的虚拟 I/O 服务的虚拟 I/O 设备 (vdisk1、vdisk11、vnet1 和 vnet11) 以及虚拟控制台 (vcons) 添加到需要新添加的来宾域 (guest1)。

```
# ldm add-vdisk vdisk10 vol10@vds1 guest0
# ldm add-vnet vnet10 vsw10 guest0
# ldm add-vdisk vdisk1 vol11@vds0 guest1
# ldm add-vdisk vdisk11 vol11@vds1 guest1
# ldm add-vnet vnet1 vsw1 guest1
# ldm add-vnet vnet11 vsw11 guest1
# ldm set-vconsole port=5101 guest1
```

j. 启动添加的来宾域。

执行 ldm bind-domain 和 ldm start-domain 命令启动添加的来宾域 (guest1)。

```
# ldm bind-domain guest1
# ldm start-domain guest1
LDom guest1 started
```

k. 在已添加的来宾域中安装 **Oracle Solaris**。

在已添加的来宾域中安装 Oracle Solaris。有关如何安装 Oracle Solaris 的详情，请参见相关 Oracle Solaris 的文档。

l. 创建各来宾域的 I/O 的冗余配置。

变更已添加到各来宾域的虚拟 I/O 设备，以便创建冗余配置。有关如何设置冗余配置的详情，请参见各冗余配置软件相关的文档。

m. 创建控制域系统卷的冗余配置。

本示例将已添加的 SPARC M10-4S 的板载设备分配给控制域，以便创建冗余配置。

将已添加的 SPARC M10-4S 的根合成体 (PCIE8 和 PCIE12) 添加到控制域。

```
# ldm add-io PCIE8 primary
# ldm add-io PCIE12 primary
```

然后，创建系统卷的冗余配置。下面的示例使用 format 命令检查磁盘，以便使用 ZFS 创建镜像配置。有关如何设置冗余配置的详情，请参见各冗余配置软件相关的文档。

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
    0. c2t50000393E802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
    sec 625>
        /pci@8000/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
        /dev/chassis/FUJITSU-BBEXP.500000e0e06d027f/0123_HDD00/disk
```

```

1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
    /pci@8800/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
    /dev/chassis/FUJITSU-BBEXP.500000e0e06d243f/022U_HDD01/disk
Specify disk (enter its number): ^C
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:

    NAME                                STATE      READ WRITE CKSUM
    rpool                                ONLINE    0     0     0
        c2t50000393E802CCE2d0s0        ONLINE    0     0     0

errors: No known data errors
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting

```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。此时，执行 `zpool status` 命令来检查同步处理 (resilver) 是否已经完成。下面的示例表明，同步处理正在进行中。

```

# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered. The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Jan 27 15:55:47 2014
    21.1G scanned out of 70.6G at 120M/s, 0h7m to go
    21.0G resilvered, 29.84% done
config:

    NAME                                STATE      READ WRITE CKSUM
    rpool                                DEGRADED    0     0     0
        mirror-0
            c2t50000393E802CCE2d0s0        ONLINE    0     0     0
            c3t50000393A803B13Ed0s0        DEGRADED    0     0     0 (resilvering)

errors: No known data errors

```

同步处理完成后，将显示如下画面：

```

# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:

    NAME                                STATE      READ WRITE CKSUM
    rpool                                ONLINE    0     0     0

```

mirror-0	ONLINE	0	0	0
c2t50000393E802CCE2d0s0	ONLINE	0	0	0
c3t50000393A803B13Ed0s0	ONLINE	0	0	0

errors: No known data errors

- n. 检查并调整各逻辑域的内存配置。
- 执行 `ldm list-devices -a memory` 命令检查内存块的配置。

```
# ldm list-devices -a memory
```

MEMORY	PA	SIZE	BOUND
	0x7000000000000	32G	root-dom1
	0x7200000000000	32G	guest1
	0x7400000000000	32G	guest1
	0x7600008000000	1272M	_sys_
	0x7600500000000	16G	primary
	0x7604500000000	10G	primary
	0x7606d00000000	4864M	
	0x7800000000000	32G	guest0
	0x7a00000000000	32G	guest0
	0x7c00000000000	256M	primary
	0x7c00100000000	3584M	root-dom0
	0x7c00f00000000	24832M	primary
	0x7c07000000000	4G	root-dom0
	0x7e00008000000	1272M	_sys_
	0x7e00500000000	512M	_sys_
	0x7e00700000000	256M	_sys_
	0x7e00800000000	25088M	root-dom0
	0x7e06a00000000	3584M	primary
	0x7e07800000000	2G	primary

- 假设分配给逻辑域的内存块和未分配的内存块被分割成小碎片。您可以通过取消绑定逻辑域，然后再次将其绑定来将零散的内存块集成成大的连续区域。此操作可很容易地添加或删除 SPARC M10-4S。
- 有关步骤的详情，请参见 ["A.2 对于从 XCP 2220 或更高版本的新安装，逻辑域配置无可用的硬件资源"](#) 的 ["A.2.2 物理分区配置步骤示例"](#) 中的步骤 11。

- o. 保存逻辑域配置信息。
- 执行 `ldm list-domain` 命令检查配置的逻辑域是否已经启动。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	56G	64%	2h 54m
guest0	active	-n----	5100	64	64G	42%	2h 54m
guest1	active	-n----	5101	64	64G	11%	2h 54m
root-dom0	active	-n--v-	5000	32	32G	20%	2h 54m
root-dom1	active	-n--v-	5001	32	32G	18%	2h 54m

- 执行 `ldm add-sponconfig` 命令将逻辑域配置信息保存到 XSCF。

```
# ldm add-spconfig ldm-set4
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3
ldm-set4 [current]
```

A.5 向仅由 SPARC64 X 处理器配置的物理分区安装由 SPARC64X+ 处理器配置的系统板

本节介绍将搭载 SPARC64 X+ 处理器的 SPARC M10-4S 添加到 SPARC M10 中的物理分区的环境创建示例。该物理分区搭载着应用 XCP 2210 或早期版本的 XCP 固件的 SPARC64 X 处理器。本节也将介绍该步骤示例。

A.5.1 配置示例

本节介绍将搭载 SPARC64 X+ 处理器的 SPARC M10-4S 安装到配置有搭载着 SPARC64 X 处理器的一个 SPARC M10-4S 的物理分区的配置示例。这将启用各 SPARC M10-4S 的活动状态更换。

若需采用 PPAR DR 启用 SPARC M10-4S 的活动状态更换，配置必须满足以下条件。

- 有必要通过将在各 SPARC M10-4S 上根合成体下的 I/O 设备连接到系统卷 I/O 设备和控制域的网络来创建冗余配置。这是由于以下原因：即使重新启动控制域，I/O 配置发生了变更，当移除一个 SPARC M10-4S 时，剩余 SPARC M10-4S 的 I/O 设备可被用于继续操作。
- 有必要为各 SPARC M10-4S 分割根域。这是由于以下原因：当移除 SPARC M10-4S 时，必须避免发生无法使用其他 SPARC M10-4S 的物理 I/O 设备提供的服务的这种状态。
- 有必要使用由虚拟 I/O 设备 (vdisk, vnet) 组成的来宾域来操作任务。这是由于以下原因：在 SPARC M10-4S 移除时，根域被停止，物理 I/O 设备可以被断开。此外，这些操作将不会影响正在执行操作的来宾域。

图 A-6 系统配置示例（1BB 配置: 搭载 SPARC64 X 处理器）（扩展前）

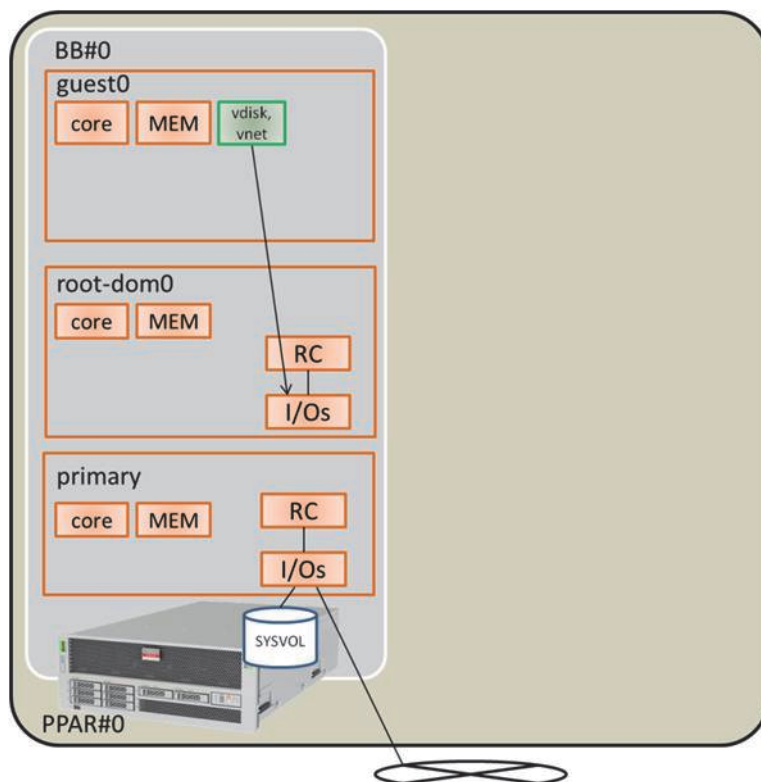


图 A-7 可进行 PPAR DR 的系统配置示例（2BB 配置: 搭载 SPARC64 X+ 处理器的系统板安装后）

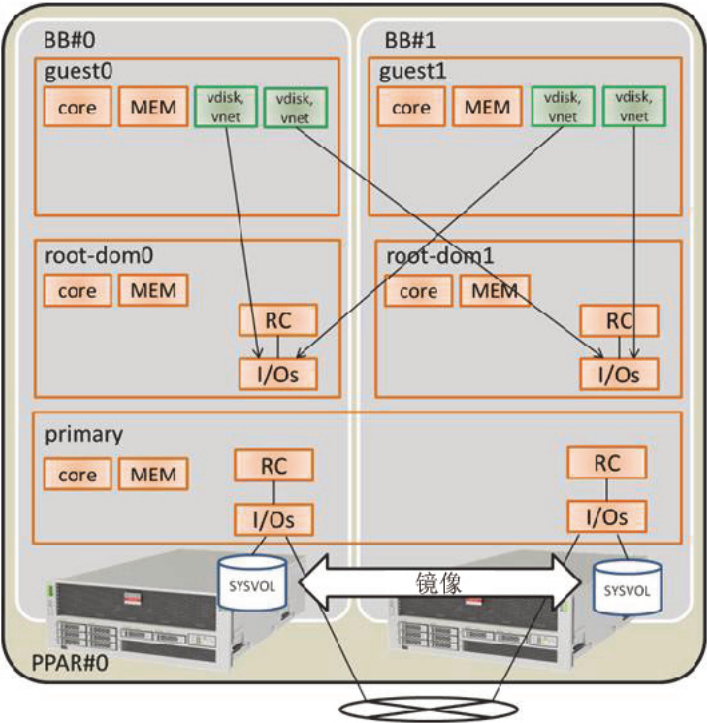


表 A-14 从 1BB 扩展到 2BB 的配置示例

逻辑域	扩展前			扩展后		
	CPU 内核	内存	I/O 配置	CPU 内核	内存	I/O 配置
控制域 (primary)	16	28 GB	On-board #0 (PCIE0, PCIE4)	32	56 GB	On-board #0 (PCIE0, PCIE4) On-board #1 (PCIE8, PCIE12)
guest0	32	64 GB	vdisk0, vnet0	32	64 GB	vdisk0, vdisk10 vnet0, vnet10
guest1	-	-	-	32	64 GB	vdisk1, vdisk11 vnet1, vnet11
root-dom0	16	32 GB	PCIE1, PCIE2, PCIE3, PCIE5, PCIE6, PCIE7, vds0, vsw0	16	32 GB	PCIE1, PCIE2, PCIE3, PCIE5, PCIE6, PCIE7, vds0, vsw0
root-dom1	-	-	-	16	32 GB	PCIE9, PCIE10, PCIE11, PCIE13, PCIE14, PCIE15, vds1, vsw1

表 A-14 从 1BB 扩展到 2BB 的配置示例 (续)

逻辑域	扩展前			扩展后		
	CPU 内核	内存	I/O 配置	CPU 内核	内存	I/O 配置
未分配的资源	0	约 2 GB(*1)	-	0	约 4.75 GB(*1)	-
总数	64	128 GB	-	128	256 GB	-

*1 在各个 SPARC M10-4S 上约 2 GB 或 1.25 GB 的内存区域被分配给虚拟机监控程序。因此，可以分配给逻辑域的内存资源比实际安装的内存容量小。

A.5.2 配置步骤

本示例将说明将搭载 SPARC64 X+ 处理器的 SPARC M10-4S 安装到搭载着应用 XCP 2210 或早期版本的 XCP 固件的 SPARC64 X 处理器的物理分区的步骤示例。

1. 将逻辑域配置信息保存到 XML 文件。

如果 XCP 固件更新后启用 PPAR DR 功能，逻辑域配置将恢复到出厂设置状态 (factory-default)。因此，无法使用保存到 XSCF 的逻辑域配置信息。出于上述原因，将当前逻辑域配置信息保存到 XML 文件，然后更新 XCP 固件。更新后，将逻辑域配置信息从 XML 文件恢复，以便启用简易逻辑域重新配置。

下面介绍将保存在 XSCF 中的逻辑域配置信息保存到 XML 文件的步骤。

但是，有关保存到 XML 文件中的配置信息的详情，请参见《Oracle VM Server for SPARC Administration Guide》。

- a. 登录控制域。
- b. 切换到需要保存的配置信息。

在控制域上执行 `ldm list-spconfig` 命令来显示逻辑域配置信息。`[current]` 行显示当前应用的配置信息。下面的示例中被应用的是 `ldm-set3`。

```
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3 [current]
```

如果仅显示 `[next poweron]`，则存储在 XSCF 中的逻辑域配置信息和存储在控制域中的逻辑域配置信息之间存在差异。因此，请执行 `ldm add-spconfig` 命令，以便使用不同的名称保存当前逻辑域配置信息。

```
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3 [next poweron]
# ldm add-spconfig ldm-set4
# ldm list-spconfig
factory-default
```

```
ldm-set1
ldm-set2
ldm-set3
ldm-set4 [current]
```

当显示在 [current] 的逻辑域配置信息与您需要保存的配置信息相匹配，请执行步骤 1-c。

如果不同，请执行 `ldm set-spconfig` 命令切换到您需要保存的配置信息。

在下面的示例中，使用 `ldm set-spconfig` 命令切换配置信息，然后保存 `ldm-set1`。

```
# ldm set-spconfig ldm-set1
# ldm list-spconfig
factory-default
ldm-set1 [next poweron]
ldm-set2
ldm-set3 [current]
```

在 XSCF 上执行 `poweroff` 和 `poweron` 命令，以便关闭物理分区 (PPAR) 的电源，然后再将其开启电源。在这种情况下，将逻辑域状态变更到 Oracle Solaris 正在运行的状态或 `inactive` 状态后，执行 `poweroff` 命令。

下面的示例关闭 PPAR-ID 0 的电源，然后再将其开启电源。

```
XSCF> poweroff -p 0
XSCF> poweron -p 0
```

执行 `ldm list-spconfig` 命令检查指定的逻辑域配置信息是否已设置。下面的示例显示将 `ldm-set1` 设置为当前配置信息。

```
# ldm list-spconfig
factory-default
ldm-set1 [current]
ldm-set2
ldm-set3
```

- c. 执行 **ldm list-constraints** 命令将当前逻辑域配置信息保存到 XML 文件。

执行 `ldm list-constraints` 命令将当前逻辑域配置信息保存到 XML 文件。

下面的示例将当前逻辑域配置信息保存到 `ldm-set1.xml`。如需防止已保存的 XML 文件丢失，请将其备份到其他介质或类似物。

```
# ldm list-constraints -x > /ldm-set1.xml
```

- d. 检查是否将配置信息保存到 XML 文件。

执行 Oracle Solaris 的 `more` 命令或相似命令，检查信息是否已经保存到 XML 文件。

```
# more /ldm-set1.xml
<?xml version="1.0"?>
<LDM_interfaceversion="1.3" xmlns:xsi=http://www.w3.org/2001/
XMLSchema-instance
```

e. 如需保存其他逻辑域配置信息，请重复步骤 1.b 至 1.e。

如果存在其他您需要保存的配置信息，请重复步骤 1.b 至 1.e。

2. 保存 **XSCF** 设置信息。

可将 XSCF 设置信息保存到 USB 设备或通过网络保存到外部服务器。下面说明各种方法。

- 将 **XSCF** 设置信息保存到 **USB** 设备

a. 登录主 **XSCF**。

执行 showbbstatus 命令来检查您已经登录的 XSCF 是否是主 XSCF。

注 - 在模块化架构配置的系统中，如果您登录到了备用 XSCF，则需再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

b. 将设置信息保存到主 **XSCF** 的 **USB** 设备。

i. 将 **USB** 设备连接到主 **XSCF** 的 **XSCF** 单元面板（后面板）上的 **USB** 端口。

ii. 指定 **XSCF** 上本地 **USB** 设备的输出文件名，然后执行 **dumpconfig** 命令。

设置信息以 base64 编码文本格式采用指定的文件名被保存。

```
XSCF> dumpconfig file:///media/usb_msd/backup-file.txt
operation completed
```

当 "operation completed" 信息出现时，数据传输已经正常结束。

iii. 数据传输完成后，从 **USB** 端口拔下 **USB** 设备。

c. 使用编辑器或个人计算机上的类似软件，检查所保存的设置文件信息。

检查以下信息。

- User-Comments: 在 dumpconfig 命令中指定 -c 选项时所编制的注释
- Created: 保存信息的日期和时间
- Platform: 型号名
- Serial-No: 系统序列号

注 - 保存到 USB 设备的任何文件都无法在 XSCF 上打开。有必要将文件下载进行检查。

```

XSCF Configuration File
User-Comments:
Encrypted: No
Created: Mon Jan 27 13:47:38 2014
Platform: M10-4S
Serial-No: 2111234001
Chassis-serial80:
Chassis-serial81:
Chassis-serial82:
Chassis-serial83:
Chassis-serial00:2111234001
Chassis-serial01:2111234003
Chassis-serial02:
Chassis-serial03:
Chassis-serial04:
Chassis-serial05:
Chassis-serial06:
Chassis-serial07:
Chassis-serial08:
Chassis-serial09:
Chassis-serial10:
Chassis-serial11:
Chassis-serial12:
Chassis-serial13:
Chassis-serial14:
Chassis-serial15:
Version: 0001
begin-base64_common
U1VOVyxTUEFSQylFbnRlcnByaXNlAAAAAFAFLmZ6gAAPrfADhbdAAAAAIyMTExMjM0MDAzAAAA
...

```

- 通过网络将设置信息保存到外部服务器

a. 登录主 **XSCF**。

执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。

注 - 在模块化架构配置的系统中，如果您登录到了备用 XSCF，则需再次登录到主 XSCF。

```

XSCF> showbbstatus
BB#00 (Master)

```

b. 通过网络指定目标目录，以便保存 **XSCF** 设置信息。

指定目标目录和输出文件名，然后执行 `dumpconfig` 命令。设置信息以 base64 编码文本格式采用指定的文件名被保存。

`user_name` 指定保存目标服务器的用户名。

```

XSCF> dumpconfig -u user-name ftp://server/backup/backup-sca-ff2-16.txt
operation completed

```

当 "operation completed" 信息出现时，数据传输已经正常结束。

- c. 使用编辑器或个人计算机上的类似软件，检查所保存的设置文件信息。

检查以下信息。

- User-Comments: 在 dumpconfig 命令中指定 -c 选项时所编制的注释
- Created: 保存信息的日期和时间
- Platform: 型号名
- Serial-No: 系统序列号

```
XSCF Configuration File
User-Comments:
Encrypted: No
Created: Mon Jan 27 13:47:38 2014
Platform: M10-4S
Serial-No: 2111234001
Chassis-serial80:
Chassis-serial81:
Chassis-serial82:
Chassis-serial83:
Chassis-serial00:2111234001
Chassis-serial01:2111234003
Chassis-serial02:
Chassis-serial03:
Chassis-serial04:
Chassis-serial05:
Chassis-serial06:
Chassis-serial07:
Chassis-serial08:
Chassis-serial09:
Chassis-serial10:
Chassis-serial11:
Chassis-serial12:
Chassis-serial13:
Chassis-serial14:
Chassis-serial15:
Version: 0001
begin-base64_common
U1VOVyxTUEFSQy1FbnRlcnByaXNlAAAAAAAFmZ6gAAPrfADhbdAAAAAIyMTExMjMOMDAzAAAA
...
```

3. 更新 **Oracle Solaris** 和 **Oracle VM Server for SPARC**。

获取支持物理分区动态重新配置的 Oracle Solaris 和 Oracle VM Server for SPARC，并将其应用到系统。

- a. 更新控制域的 **Oracle VM Server for SPARC**。

请参见您的服务器最新的《Product Notes》，检查最新版的 Oracle Solaris 和 Oracle VM Server for SPARC。然后使用以下步骤更新 Oracle VM Server for SPARC。

- i. 获取用于控制域的 **Oracle VM Server for SPARC**。

- 如果控制域是 **Oracle Solaris 10**

获取支持物理分区动态重新配置的 Oracle VM Server for SPARC。有关如何获取的详细信息，请参见您的服务器的最新《Product Notes》。

- 如果控制域是 **Oracle Solaris 11**

获取 SRU11.1.14.0 或更高版本。

ii. 无论何时完成更新 **Oracle VM Server for SPARC** 或控制域的 **Oracle Solaris**，都请重新启动控制域。

重新启动控制域可能导致 I/O 域崩溃或来宾域 I/O 停止。如果存在这样的逻辑域，请提前使用 shutdown 命令停止，或者从控制域执行 ldm stop-domain 命令将其停止。

使用下面的方法，可以检查逻辑域是否将要被停止。

- 从控制域分配 **PCIe** 端点的 **I/O 域**

执行 ldm list-io 命令确定分配给逻辑域的 PCIe 端点的根合成体 (BUS) 是否被分配给 primary。下面的示例表明分配给 iodom0 的 PCIe 端点 "/BB0/PCI3" 和 "/BB0/PCI4" 的总线 "PCIE2" 被分配给 primary（控制域）。请注意，由于这里阐述的配置与 "A.5.1 配置示例" 中的配置不同，所以对逻辑域和一些其他信息进行了变更。

# ldm list-io				
NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	primary	IOV
PCIE2	BUS	PCIE2	primary	IOV
PCIE3	BUS	PCIE3	primary	IOV
....				
/BB0/CMUL/NET0	PCIE	PCIE0	primary	OCC
/BB0/CMUL/SASHBA	PCIE	PCIE0	primary	OCC
/BB0/PCI0	PCIE	PCIE1	primary	OCC
/BB0/PCI3	PCIE	PCIE2	iodom0	OCC
/BB0/PCI4	PCIE	PCIE2	iodom0	OCC
/BB0/PCI7	PCIE	PCIE3	primary	OCC
/BB0/PCI8	PCIE	PCIE3	primary	EMP
....				

- 分配控制域虚拟服务的来宾域

执行 ldm list-bindings primary 检查虚拟网络交换机 (VSW) 和连接目标 (PEER) 之间的对应关系，以及虚拟磁盘服务 (VDS) 和使用虚拟磁盘的逻辑域 (CLIENT) 之间的对应关系。在下面的示例中，可以确认 VSW PEER 和 VDS CLIENT 设置为 "guestdom0"。

# ldm list-bindings primary						
....						
VSW						
NAME	MAC	NET-DEV	ID	DEVICE	LINKPROP	
DEFAULT-VLAN-ID	PVID					
VID	MTU	MODE	INTER-VNET-LINK			
vsw0	00:14:4f:f9:88:ca	net0	0	switch@0		1
1		1500	on			
PEER	MAC			PVID	VID	

MTU	MAXBW	LINKPROP			
INTERVNETLINK					
	vnet0@guestdom0		00:14:4f:fa:64:dd	1	
1500					
VDS					
	NAME	VOLUME	OPTIONS	MPGROUP	DEVICE
....					
	CLIENT	VOLUME			
	vdisk0@guestdom0	vol0			

执行 `ldm stop-domain` 命令停止上述已检查的逻辑域。在下面的示例中，执行 `ldm stop-domain` 命令来停止 `iodom0` 和 `guestdom0`，然后执行 `ldm list-domain` 命令来检查是否设置为 `bound` 状态。

```
# ldm stop-domain guestdom0
LDom guestdom0 stopped
# ldm stop-domain iodom0
LDom iodom0 stopped
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	58G	0.0%	6h 3m
guestdom0	bound	-----	5100	64	64G		
iodom0	bound	-----	5000	32	32G		

iii.更新控制域的 **Oracle VM Server for SPARC** 。

- **Oracle Solaris 10** 的情形

卸载旧版 Oracle VM Server for SPARC，然后安装新版 Oracle VM Server for SPARC。有关详情，请参见所获得的 Oracle VM Server for SPARC 随附的 README 文件。

- **Oracle Solaris 11** 的情形

应用 SRU。有关详情，请参见所获得的 SRU 的安装手册。

iv.执行 **shutdown** 命令重启控制域。

```
# shutdown -i6 -g0 -y
....
```

v. 执行 **ldm start-domain** 命令启动在上述 ii 中停止的逻辑域。

在下面的示例中, 执行 `ldm start-domain` 命令启动 "`iodom0`" 和 "`guestdom0`"，然后执行 `ldm list-domain` 命令检查是否设置为 `active` 状态。

```
# ldm start-domain guestdom0
# ldm start-domain iodom0
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	64	58G	0.0%	6h 3m
guestdom0	active	-n----	5100	64	64G		
iodom0	active	-n----	5000	32	32G		

b. 更新除控制域之外的逻辑域的 **Oracle Solaris**。

请参见您的服务器的最新《Product Notes》，并为除控制域之外的逻辑域更新 Oracle Solaris。有关更新步骤的详情，请参见与各更新有关的信息。

4. 更新 **XCP** 固件。

将搭载 SPARC64 X+ 处理器的 SPARC M10-4S 安装在配置有 SPARC64 X 处理器的物理分区。若需进行此操作，请将配置有 SPARC64 X 处理器的物理分区更新为 XCP 2220 或更高版本的 XCP 固件。

a. 获取最新版 **XCP** 固件。

i. 下载 **XCP** 固件的程序文件。

从网站上将 XCP 固件的程序文件（XCPxxxx.tar.gz 或 XCPxxxx.exe）下载到与本系统相连接的 PC 上的任意一个文件夹。

使用以下其中一种方法获取您正在使用的服务器的固件。

- 日文网站

曾经登录过 SupportDesk 的用户可从 SupportDesk-Web 获取固件。

- 全球网站

有关获取固件最新文件方法的详情，请联系本公司销售人员。提供以下文件。

- 固件程序文件

（XSCF Control Package (XCP) 文件）

- XSCF 扩展 MIB (XSCF-SP-MIB) 定义文件

ii. 检查所下载程序文件的 **XCP** 版本。

检查所下载 XCP 固件程序文件的版本。对于 XCP 版本，请参考固件程序（tar.gz 格式）的文件名中的 4 位数字，以便检查是否需要应用的 XCP 固件版本更新。例如，如果程序文件的名称是 "XCP2220.tar.gz"，则 XCP 版本是 2220。

iii. 解压缩下载的程序文件。

解压缩所下载 XCP 固件的程序文件。导入到系统中的 XCP 映像文件被展开。例如，如果解压缩 "XCP2220.tar.gz"，则 "BBXCP2220.tar.gz" 被展开。

b. 检查当前 **XCP** 固件版本。

i. 登录主 **XSCF**。

执行 showbbstatus 命令来检查您已经登录的 XSCF 是否是主 XSCF。

注 - 在模块化架构配置的系统中，如果您登录到了备用 XSCF，则需再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

ii. 执行 **version** 命令检查当前系统的 **XCP** 版本。

更新固件前，请检查当前系统的 XCP 版本。在下面的示例中，执行带有 -c xcp 选项的 version 命令，以便检查 XCP 版本是否是不支持物理分区动态重新配置的 XCP 2041。

```
XSCF> version -c xcp
BB#00-XSCF#0 (Master)
XCP0 (Current): 2041
XCP1 (Reserve): 2041
```

c. 更新 XCP 固件。

i. 执行 **poweroff** 命令关闭所有物理分区的电源。

```
XSCF> poweroff -a
```

执行 **showpparstatus** 命令检查是否所有物理分区电源已关闭。

```
XSCF> showpparstatus -a
PPAR-ID          PPAR Status
0                Powered Off
```

ii. 执行 **showhardconf** 命令检查主、备 XSCF 的 [Status] 是否是 "Normal"。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Service;
+ System_Power:Off; System_Phase:Cabinet Power Off;
Partition#0 PPAR_Status:Powered Off;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
:
```

iii. 执行 **getflashimage** 命令导入 XCP 镜像文件。

在下面的示例中，将 USB 设备连接到主 XSCF 的 XSCF 单元面板（后面板）上的 USB 端口（印刷 MAINTENANCE ONLY），并导入 XCP 映像文件。

```
XSCF> getflashimage file:///mnt/share/scf-firm/xscf/user/scfadmin/BBXCP2220.tar.gz
0MB received
1MB received
2MB received
...
86MB received
87MB received
88MB received
Download successful: 90539 Kbytes in 58 secs (1562.668 Kbytes/sec) Checking
file...
MD5: 2b89c06548205ce35a8ecb6c2321d999
```

当正常结束消息"Download successful: ..." 和 "MD5: ..." 出现时，表示 XCP 镜像文件的导入已经结束。

注 - 当 XCP 镜像文件导入时，可能出现消息 "Warning: About to delete existing old versions"。此信息要求您确认是否要删除已经导入的旧版 XCP 映像文件。如果出现 "Continue?"，请输入 "y" 继续导入处理。

注 - 如果导入 XCP 镜像文件后出现 "Error:File is invalid or corrupt" 消息，这表明导入的 XCP 镜像文件不合适。此 XCP 镜像文件可能已损坏。因此，请获取正确的 XCP 镜像文件，然后将其导入。

iv.执行 **getflashimage -l** 命令检查导入的 **XCP** 镜像文件的版本。

```
XSCF> getflashimage -l
Existing versions:
      Version              Size  Date
      BBXCP2220.tar.gz    92712351  Thu May 23 15:01:42 JST 2014
```

v. 执行 **flashupdate -c check** 命令检查导入的 **XCP** 镜像文件是否可用于更新。
执行 **flashupdate** 命令后，立即执行 **showresult** 命令。如果结束值是 0，可进行更新。

```
XSCF> flashupdate -c check -m xcp -s 2220
XCP update is started. [3600sec]
0XSCF>
XSCF> showresult
0
XSCF>
```

vi.执行 **flashupdate** 命令更新固件。

```
XSCF> flashupdate -c update -m xcp -s 2220
The XSCF will be reset. Continue? [y|n]: y
XCP update is started. [3600sec]
0.....30.....60.....90.....120.....150.....180.....210.....240.....-
270.....300.....330.....360.....390.....420.....450.....480.....510.....|
540.....570.....600.....630.....660.....690.....720.....750.....780.....-
810.....840.....870.....900.....930
:
```

此时，XSCF 被重新启动，XSCF 会话被断开。此时，XCP 固件更新尚未完成。

注 - 在固件工作时间内，更新操作将历时 45 分钟。在模块化架构的系统上，更新时间约 60 分钟，完成更新后 XSCF 的自动切换大约需要 10 分钟。

注 - 如需安全地进行更新，请勿操作物理分区的电源，直到出现 "XCP update has been completed" 信息，表明 XCP 固件已更新。

vii.再次连接到主 **XSCF**。

注 -在模块化架构配置的系统中，XSCF 重新启动之后，主 XSCF 和备用 XSCF 将与原来的状态恰好相反。例如，如果 BB-ID 0 的主 XSCF 执行固件更新，然后再次进行 XSCF 连接，则 BB-ID 1 进入主状态，BB-ID 0 进入备用状态。

注 - 如果设置了继承的 IP 地址，然后用于连接，将自动连接到主 XSCF。

viii.执行 **showbbstatus** 命令来检查是否已经登录主 XSCF。

注 - 在模块化架构配置的系统中，如果您登录到了备用 XSCF，则需再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

ix.执行 **showlogs monitor** 命令来检查 XCP 固件更新是否已完成。

如果出现 "XCP update has been completed" 信息，则 XCP 固件更新已经完成。

```
XSCF> showlogs monitor
Jan 23 16:42:57 5 M10-4S-0 Event: SCF:XCP update is started (XCP version=2220:
last version=2041)
Jan 23 16:44:52 M10-4S-0 Event: SCF:XSCF update is started (BBID=0, bank=0)
Jan 23 16:45:11 M10-4S-0 Event: SCF:XSCF writing is performed (BBID=0, XSCF
version=02220000)
:
Jan 23 16:54:05 01:50 M10-4S-0 Event: SCF:XSCF bank apply has been completed
(BBID=0, bank=0, XCP version=2220: last version=2041)
:
Jan 23 17:19:57 32:38 M10-4S-1 Event: SCF:CMU update has been completed (BBID=0)
Jan 23 17:19:59 32:41 M10-4S-1 Even SCF:XCP update has been completed (XCP
version=2220: last version=2041)
```

注 - 如果没有出现 "XCP update has been completed" 信息，则更新尚未完成。再次执行 showlogs monitor 命令来检查更新是否已完成。

x.执行 **showhardconf** 命令检查 XSCF 的 [Status] 是否是 "Normal"。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Service;
+ System_Power:Off; System_Phase:Cabinet Power Off;
Partition#0 PPAR_Status:Powered Off;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
:
```

xi.执行 **version** 命令检查固件是否是新版本的固件。

```
XSCF> version -c xcp
BB#00-XSCF#0 (Master)
XCP0 (Current): 2220
XCP1 (Reserve): 2220
```

注 -如果开启物理分区的电源并更新固件，则 CMU 固件的当前排列将是新排列。如果关闭物理分区的电源并更新固件，则 CMU 固件的预约排列和当前排列将是新排列。有关 XCP 版本相关的 CMU 固件的版本详情，请参见最新产品说明中的 "现有 XCP 固件版本和支持信息"。

5. 启用物理分区动态重新配置功能（PPAR DR 功能）。

注 -为了使 SPARC64 X 处理器的各 SPARC M10-4S 和 SPARC64 X+ 处理器的 SPARC M10-4S 共存于一个物理分区内，请执行 **setpparmode** 命令将 CPU 运行模式设置为 **compatible** 模式（SPARC64 X 兼容模式）。有关详情，请参见 ["2.6.3 PPAR 内 SPARC64 X+ 处理器和 SPARC64 X 处理器的混合配置条件"](#)。

注 -如需使用物理分区动态重新配置，有必要使用 **setpparmode** 命令来启用 PPAR DR 模式。但是，如果启用了 PPAR DR 模式，逻辑域配置信息将变更为出厂默认值。因此，无法使用现有的逻辑域配置信息。在这种情况下，您必须新建逻辑域配置信息。若需使用现有的逻辑域配置信息，请禁用 PPAR DR 模式。在使用现有的逻辑域配置信息时，假设启用了 PPAR DR 模式，并且开启了物理分区的电源。将发生虚拟机监控程序中止和未能正常启动 Oracle VM Server for SPARC 等问题。

a. 执行 **showpparmode** 命令查看 PPAR DR 模式设置。

下面的示例指定物理分区编号 0 (PPAR ID 0)。

```
XSCF> showpparmode -p 0
Host-ID :9007002b
Diagnostic Level :min
Message Level :normal
Alive Check :on
Watchdog Reaction :reset
Break Signal :on
Autoboot (Guest Domain) :on
Elastic Mode :off
IOreconfigure :false
PPAR DR (Current) :-
PPAR DR (Next) :off
```

b. 如果 CPU 运行模式是 **auto**，执行 **setpparmode** 命令将该模式设置为 **compatible**。

```
XSCF> setpparmode -p 0 -m cpumode=compatible
```

c. 如果 PPAR DR 模式被禁用, 执行 **setpparmode** 命令来启用该模式。

```
XSCF> setpparmode -p 0 -m ppar_dr=on
```

d. 执行 **showpparmode** 命令来检查 **PPAR DR** 模式是否已启用。

```
XSCF> showpparmode -p 0
Host-ID                :9007002b
Diagnostic Level       :min
Message Level          :normal
Alive Check            :on
Watchdog Reaction      :reset
Break Signal           :on
Autoboot(Guest Domain) :on
Elastic Mode           :off
IOreconfigure          :false
PPAR DR(Current)       :-
PPAR DR(Next)          :on
```

6. 执行 **showsscp** 命令检查内部服务处理器通信协议 (**SSCP**) 的 IP 地址是默认值，还是用户指定值。

```
XSCF> showsscp
```

当 IP 地址是默认值，并且您正在使用需要扩展的 SPARC M10-4S 的 IP 地址默认值时，请转到下一步。

若需设置用户值，请使用 **setsscp** 命令设置 IP 地址。然后，使用 **applynetwork** 命令来应用和检查需要扩展的 SPARC M10-4S 的 SSCP 的 IP 地址。随后，执行 **rebootxscf** 命令完成设置，然后转到下一步。有关步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.6 应用网络设置"。

7. 扩展 **SPARC M10-4S**。
a. 按照信息，执行 **addfru** 命令来扩展 **SPARC M10-4S**。

注 - 如果尚未设置 SSCP 的 IP 地址，执行 **addfru** 命令将导致错误。

注 - 如果执行 **addfru** 命令，需要扩展的 SPARC M10-4S 的固件版本将自动设置为主 XSCF 正在运行的 SPARC M10-4S 的版本。在此示例中，需要扩展的 SPARC M10-4S (BB-ID 1) 的固件版本将自动设置为主 XSCF 正在运行的 SPARC M10-4S (BB-ID 0) 的版本。

下面是扩展 BB#1 的示例。

```
XSCF> addfru
-----
Maintenance/Addition Menu
Please select the chassis including added FRU.
No. FRU                Status
-----
 1 /BB#0                Normal
 2 /BB#1                Unmount
 3 /BB#2                Unmount
```

```

-----
Select [1-16|c:cancel] :2
Maintenance/Addition Menu
Please select the BB or a type of FRU to be added.
1. BB itself
2. PSU (Power Supply Unit)
-----

Select [1,2|c:cancel] :1
Maintenance/Addition Menu
Please select a FRU to be added.
No. FRU                      Status
-----
1 /BB#1                      Unmount
Select [1|b:back] :1

```

b. 将需要扩展的 **SPARC M10-4S** 安装到机架。

在执行上述提到的 `addfru` 命令时，如果出现 "After the added device is connected with the system, please turn on the breaker of the BB#1." 消息，则将需要扩展的 SPARC M10-4S 安装在机架上。

有关安装步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "3.4.1 在机架中安装 SPARC M10-4S"。

c. 设置需要扩展的 **SPARC M10-4S** 的识别 ID (BB-ID)。

请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.1 设置标识机柜的 ID (BB-ID)"。

d. 连接接线。

在现有的 SPARC M10-4S 和扩展的 SPARC M10-4S 之间连接接线。关于连接线缆路由图和线缆列表，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "附录 B 模块化架构配置上的线缆连接信息"。有关机箱之间直接连接的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.2 连接线缆（机柜之间直接连接）"。

e. 连接 **XSCF BB** 控制线。

通过 XSCF BB 控制线，连接现有的 SPARC M10-4S 和扩展的 SPARC M10-4S。关于 XSCF BB 控制线的路由图和线缆列表，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "附录 B 模块化架构配置上的线缆连接信息"。有关机箱之间直接连接的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.2 连接线缆（机柜之间直接连接）"。

f. 连接 **XSCF DUAL** 控制线。

通过 XSCF DUAL 控制线，连接现有的 SPARC M10-4S 和扩展的 SPARC M10-4S。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "4.2 连接线缆（机柜之间直接连接）"。

g. 连接串行线或 **LAN** 线。

将串行线连接到需要扩展的 SPARC M10-4S 的 XSCF 串行端口。而且，将线缆连接到 XSCF-LAN 的各 LAN 端口、GbE 端口和 PCIe 卡。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "5.1 将线缆连接到 SPARC M10-4S"。

h. 连接输入电源到需要扩展的 **SPARC M10-4S**。

连接需要扩展的 SPARC M10-4S 的电源线到输入电源。

i. 在 **addfru** 命令的输入画面中输入 **"f"**。

在步骤 a. 所执行的 **addfru** 命令的输入画面中输入 **"f"**，以便执行 SPARC M10-4S 扩展。

```
2) Please select[f:finish] :f

Waiting for BB#1 to enter install state.
[This operation may take up to 20 minute(s)] (progress scale
  reported in seconds)
0..... 30..... done

Waiting for BB#1 to enter ready state.
[This operation may take up to 45 minute(s)] (progress scale
  reported in seconds)
0..... 30..... 60... done

Do you want to start to diagnose BB#1? [s:start|c:cancel] :
```

j. 跳过需要扩展的 **SPARC M10-4S** 的诊断，以便退出 **addfru** 命令。

在主 XSCF 的 **addfru** 命令的输入画面输入 **"c"**，然后跳过需要扩展的 SPARC M10-4S 的诊断处理。如果出现 **"The addition of BB#1 has completed."** 信息，则输入 **"f"**，然后输入 **"c"**，以便退出 **addfru** 命令。

```
Do you want to start to diagnose BB#1? [s:start|c:cancel] :c

Diagnostic tests are about to be skipped.
Running diagnostic tests are strongly recommended before using BB#1.
Are you sure you want to skip testing? [y:yes|n:no] :y

-----
Maintenance/Addition Menu Status of the added FRU.

FRU                Status
-----
/BB#1              Normal
-----

[Warning:007]
Running diagnostic tests on BB#1 is strongly recommended after addfru has
completed.
The addition of BB#1 has completed.[f:finish] :f

-----
Maintenance/Addition Menu
Please select the chassis including added FRU.

No.  FRU                Status
---  -
1    /BB#0              Normal
2    /BB#1              Normal
```



```
-----  
Select [1,2|c:cancel] :c
```

8. 诊断已扩展的 **SPARC M10-4S**。

a. 执行 **testsb** 命令来实施已扩展的 **SPARC M10-4S** 的诊断测试。

执行 **testsb** 命令实施诊断测试。指定已扩展的 SPARC M10-4S 的物理系统板 (PSB) 编号，并检查初始诊断和连接 I/O。

```
XSCF> testsb -v -p -s -y 01-0  
Initial diagnosis is about to start, Continue?[y|n] :y  
PSB#01-0 power on sequence started.
```

如果显示错误，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "A.2.4 检查诊断结果"。

b. 指定 **diagxbu** 命令的 **-t** 选项，执行现有 **SPARC M10-4S** 和已扩展的 **SPARC M10-4S** 的接线诊断测试。

下面的示例指定现有 SPARC M10-4S 的识别 ID "00" (BB-ID 00) 和已扩展 SPARC M10-4S 的识别 ID "01" (BB-ID 01)，以诊断接线连接状况。

```
XSCF> diagxbu -y -b 00 -t 01  
XBU diagnosis is about to start, Continue?[y|n] :y  
Power on sequence started. [7200sec]  
0..... 30..... 60..end  
XBU diagnosis started. [7200sec]  
0..... 30..... 60..... 90.....120.....150.....180.....210.....240...../  
270.....300.....330.....360.....390.....420.....450.....480.....510.....\  
540.....570.....600.....630.....660.....690.....720.....750.....780.....\  
810.....840.....870.....900.....930..end  
completed.  
Power off sequence started. [1200sec]  
0..... 30..... 60..... 90.....120.....150.....180.end  
completed.
```

c. 执行 **showlogs error** 命令检查是否显示错误。

```
XSCF> showlogs error
```

如果显示错误，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "A.2.4 检查诊断结果"。

d. 执行 **showhardconf** 命令来检查已扩展的 **SPARC M10-4S** 的配置和状态。

执行 **showhardconf** 命令检查已扩展的 SPARC M10-4S 的硬件配置 (CPU、内存或类似物)，然后检查各单元的 Status 是否是 Normal。

```
XSCF> showhardconf  
SPARC M10-4S;  
+ Serial: 2081230011; Operator_Panel_Switch:Service;
```

```

+ System_Power:Off; System_Phase:Cabinet Power Off;
Partition#0 PPAR_Status:Powered Off;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
  + FRU-Part-Number: CA07361-D202 A1 ;
  + Power_Supply_System: ;
  + Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;
  + FRU-Part-Number:CA07361-D941 C4 /7060911 ;
  + Memory_Size:128 GB; Type: A ;
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;
  + Freq:3.000 GHz; Type:0x10;
  + Core:16; Strand:2;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
  + FRU-Part-Number: CA20393-B50X A2 ;
  + Power_Supply_System: ;
  + Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
  + FRU-Part-Number:CA07361-D941 C4 /7060911 ;
  + Memory_Size:128 GB; Type: A ;
:

```

9. 设置已扩展的 **SPARC M10-4S** 的 **XSCF** 网络。
 设置已扩展的 SPARC M10-4S 的 XSCF 网络。有关详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.2 设置以太网 (XSCF-LAN) IP 地址" 和 "7.5.3 设置接管 IP 地址"。进行此设置后，执行 **applynetwork** 命令来应用和检查该设置。随后，执行 **rebootxscf** 命令完成设置，然后转到下一步。有关步骤的详情，请参见《Fujitsu M10-4S/SPARC M10-4S 安装指南》中的 "7.5.6 应用网络设置"。
10. 设置已扩展的 **SPARC M10-4S** 的内存的镜像模式。
 如果您未正在使用内存镜像模式，将不需要此步骤。
 有关内存镜像模式的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "14.1 配置内存镜像"。
 - a. 如果您正在进行已扩展的 **SPARC M10-4S** 中内存的镜像配置，则使用 **setupfru** 命令来设置内存镜像模式。
 以下示例将 SPARC M10-4S 物理系统板 (PSB 01-0) 下 CPU 的所有内存置于内存镜像模式。

```
XSCF> setupfru -m y sb 01-0
```

- b. 执行 **showfru** 命令来检查内存的镜像模式设置。

```

XSCF> showfru sb 01-0
Device Location Memory Mirror Mode
sb      01-0
      cpu 01-0-0    yes
      cpu 01-0-1    yes
      cpu 01-0-2    yes
      cpu 01-0-3    yes

```

11. 在物理分区配置信息中登录已扩展的 **SPARC M10-4S** 的系统板。
- a. 执行 **showpcl** 命令检查物理分区配置信息。

```
XSCF> showpcl -p 0
PPAR-ID      LSB      PSB      Status
00
              00      00-0     Powered Off
```

- b. 执行 **setpcl** 命令在物理分区配置信息中登录系统板。
- 执行 **setpcl** 命令在内置目标的物理分区配置信息中登录扩展的 SPARC M10-4S 系统板。
- 下面的示例中，将物理系统板 (PSB) 01-0 映射到物理分区 0 的逻辑系统板 (LSB) 01。

```
XSCF> setpcl -p 0 -a 01=01-0
```

- c. 执行 **showpcl** 命令检查物理分区配置信息。

```
XSCF> showpcl -p 0
PPAR-ID      LSB      PSB      Status
00
              00      00-0     Powered Off
              01      01-0
```

12. 登录 **CPU** 激活密钥以便分配 **CPU** 内核资源。
- a. 执行 **showcodusage** 命令查看 **CPU** 内核资源信息。

执行 **showcodusage** 命令，检查物理分区是否包含可分配的 CPU 内核资源。

如此处所示，该系统已安装 128 个 CPU 内核资源，已注册 128 个 CPU 激活，64 个 CPU 内核资源正在使用中，当前未在使用的 CPU 激活的数量是 64。

```
XSCF> showcodusage -p resource
Resource In Use Installed CoD Permitted Status
-----
PROC      64      128      128 OK: 64 cores available

Note:
  Please confirm the value of the "In Use" by the ldm command of Oracle VM
  Server for SPARC.

  The XSCF may take up to 20 minutes to reflect the "In Use" of logical domains.
```

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。有关如何添加 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "5.3 添加 CPU 内核资源"。

b. 执行 **showcod** 命令检查物理分区的 CPU 内核资源信息。

执行 **showcod** 命令检查分配给物理分区的 CPU 内核资源的信息。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 64
```

如果已分配的资源不足，执行 **setcod** 命令将 CPU 内核资源分配给物理分区。
以下是将 64 个 CPU 内核资源添加到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c add 64
PROC Permits assigned for PPAR 0 : 64 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 -XCP 2250 和更早版本的 XSCF 固件不支持 -c add、-c delete 和 -c set 选项。按如下所示指定 **setcod** 命令的选项可交互式添加和删除 CPU 内核资源。
XSCF> setcod -s cpu

当 CPU 内核资源已经被分配时，再次执行 **showcod** 命令检查已分配给物理分区的 CPU 内核资源的信息。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 128
```

13. 将已扩展的 **SPARC M10-4** 的系统板 (**PSB<BB>**) 整合到物理分区。

a. 返回 **XSCF shell**，然后执行 **showboards** 命令检查 **PSB** 的状态。

执行 **showboards** 命令检查已扩展的 SPARC M10-4S 的 PSB 状态是否是 "SP" (系统板池)。

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB)  Assignment  Pwr  Conn  Conf  Test  Fault
-----
00-0  00(00)         Assigned   n    n     n     Passed Normal
01-0  SP              Available  n    n     n     Passed Normal
```

b. 执行 **addboard** 命令，将 **PSB** 分配给物理分区。

执行 **addboard -c assign** 命令，将 PSB 分配给物理分区。

```
XSCF> addboard -c assign -p 0 01-0
PSB#01-0 will be assigned into PPAR-ID 0. Continue?[y|n] :y
XSCF>
```

注 - 如果执行 `addboard` 命令过程中出现错误信息，请参见 "C.1.1 `addboard`"，然后识别错误并采取应对措施。

c. 执行 `showresult` 命令检查刚刚执行的 `addboard` 命令的结束状态。

结束值 0 表明 `addboard` 命令正常终止。

如果结束值非 0，或如果在执行 `addboard` 命令时显示错误信息，则表明 `addboard` 命令异常终止。根据错误信息，通过参考 "C.1.1 `addboard`" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

执行 `showboards` 命令检查 PSB 状态。

确认已添加的 SPARC M10-4S 内的 PSB 是否处于 Assigned 状态，以及 Pwr、Conn 和 Conf 栏是否全显示 "n"。

```
XSCF> showboards -p 0
PSB   PPAR-ID(LSB)  Assignment  Pwr   Conn  Conf  Test   Fault
-----
00-0  00(00)         Assigned    n     n     n     Passed Normal
01-0  00(01)         Assigned    n     n     n     Passed Normal
```

14. 开启物理分区的电源。

执行 `showdomainconfig` 命令来检查将要启动的逻辑域的配置信息是否设置为出厂默认值。下面的示例指定物理分区编号 0 (PPAR ID 0)。

```
XSCF> showdomainconfig -p 0
PPAR-ID :0
Booting config
(Current) : factory-default
(Next) : factory-default
-----
Index :1
config_name :factory-default
domains :1
date_created:-
-----
Index :2
config_name :ldm-set1
domains :8
date_created:'2012-08-0811:34:56'
-----
Index :3
config_name :ldm-set2
domains :20
date_created:'2012-08-0912:43:56'
:
```

执行 `poweron` 命令来开启物理分区的电源。

```
XSCF> poweron -p 0
```

15. 使用 XML 文件，重新配置逻辑域。

完成物理分区的启动后，从步骤 1 中保存的 XML 文件重新配置逻辑域。

a. 检查系统是否已经采用控制域的 **factory-default** 设置启动。

执行 `ldm list-spconfig` 命令检查 `[current]` 是否在 `factory-default` 行显示。

下面显示命令执行的示例。

```
# ldm list-spconfig
factory-default [current]
ldm-set1
ldm-set2
ldm-set3
```

b. 使用 `ldm init-system` 命令，从 XML 文件应用逻辑域配置信息。

执行 `ldm init-system` 命令应用所保存的 XML 文件设置。然后，执行 `shutdown` 命令重新启动控制域。

```
# ldm init-system -i /ldm-set1.xml
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
# shutdown -y -g0 -i6
```

c. 当 **Oracle VM Server for SPARC** 的版本早于 **3.2** 时，为了启用物理分区动态重新配置，需调整现有逻辑域的内存容量。

执行 `ldm list-domain` 命令或相似命令，检查是否已经从 XML 文件重新配置逻辑域。

```
# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	32	28G	0.0%	6m
guest0	inactive	-----		64	64G		
root-dom0	inactive	-----		32	32G		

为了能够使用物理分区动态重新配置，请将各逻辑域的内存容量设置为“逻辑域的 CPU 内核数量 x 256 MB 的倍数”。

如需重新设置除控制域之外的逻辑域的内存容量，请使用 `ldm set-memory` 命令。

如需重置控制域的内存容量，进行如下操作：首先使用 `ldm start-reconf` 命令首先进入延迟重新配置模式，使用 `ldm set-core` 命令设置相同的内核数量，然后使用 `ldm set-memory` 命令重新设置内存容量。最后，重新启动 Oracle Solaris。

在此示例中，由于控制域的 CPU 内核数量是 32，设置 $32 \times 256 \text{ MB} = 8192 \text{ MB}$

的倍数。

首先，获得 8192 MB 的倍数，使该值更接近初始设定值 (58 GB = 59392 MB)。其结果是 $59392/8192 = 7.25$ 。舍位至 7。因此，需要重新分配给控制域的内存容量为 $8192\text{ MB} \times 7 = 56\text{ GB}$ 。

下面的示例显示重新设置控制域内存为 56 GB 的命令的执行。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
# ldm set-core 32 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm set-memory 56G primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# shutdown -i6 -g0 -y
```

执行 `ldm list-domain` 命令检查内存容量 (MEMORY) 是否已经正确设置。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS   VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-  UART   64    56G     0.0%  6m
guest0        inactive -----    64    64G
root-dom0     inactive -----    32    32G
```

执行 `ldm bind-domain` 命令绑定各逻辑域，然后通过执行 `ldm start-domain` 命令启动各逻辑域。

下面的示例显示了命令的执行。

```
# ldm bind-domain root-dom0
# ldm bind-domain guest0
# ldm start-domain root-dom0
LDom root-dom0 started.
# ldm start-domain guest0
LDom guest0 started.
```

执行 `ldm list-domain` 命令，然后确认各逻辑域是否已经启动。检查 [STATE] 是否是 "active" 并且 [FLAGS] 字符串从左起第二个字符是否是 "n"。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS   VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-  UART   32    28G     0.0%  18m
guest0        active    -n----  5100   64    64G     0.1%  2m
```

16. 添加新逻辑域。

构建一个新逻辑域，并分配已添加的 SPARC M10-4S 的资源。下面的示例根据配置示例，创建镜像配置。换言之，添加来宾域 `guest1` 和根域 `root-dom1`，以便创建各来宾域的虚拟 I/O 的冗余配置。此外，使用内置硬盘，创建控制域的系统卷的镜像配置。

- a. 执行 `ldm list-devices` 命令检查已添加的硬件资源的状态。
- 当执行 `ldm list-devices` 命令后，将列出所有未分配的硬件资源。检查是否已添加硬件资源。下面的示例通过数 `ldm list-devices -a -p core` 命令的输出结果中 "free = 100" 的行数检查未分配的 CPU 内核数量。

```
# ldm list-devices -a -p core | grep "free=100" | wc -l
64
```

对于内存而言，检查已安装的 SPARC M10-4S 位置和使用 `prtdiag` 命令输出的内存信息的 [Contains Modules and Base Address] 中的内存物理地址之间的映射。接下来，使用 `ldm-list-devices memory` 命令检查未分配的内存的物理地址和容量。

下面的示例使您能够确定未分配的内存已添加。这是明确的事实，即 BB-ID 1 内存的物理地址的开头是 0x700000000000 至 0x760000000000，使用 `ldm list-devices memory` 命令将显示此范围内的地址。

```
# prtdiag
....
===== Physical Memory Configuration =====
Segment Table:
-----
Base      Segment  Interleave  Bank      Contains
Address   Size      Factor      Size      Modules
-----
....
0x760000000000 32 GB    4           8 GB     /BB1/CMUL/CMP0/MEM00A
....
0x740000000000 32 GB    4           8 GB     /BB1/CMUL/CMP1/MEM10A
....
0x720000000000 32 GB    4           8 GB     /BB1/CMUU/CMP0/MEM00A
....
0x700000000000 32 GB    4           8 GB     /BB1/CMUU/CMP1/MEM10A
....
# ldm list-devices memory
MEMORY
  PA      SIZE      BOUND
  0x700000000000 32G
  0x720000000000 32G
  0x740000000000 32G
  0x760050000000 31488M
....
```


b. 创建可分配 CPU 内核和内存资源的新逻辑域。

但是，配置域时，如果首先使用 `ldm set-core` 命令设置 CPU 内核数，然后使用 `ldm set-memory` 命令设置需要分配的内存大小，则分配给逻辑域的内存块会变得难以分散（碎片化），而在 SPARC M10-4S 释放时内存配置变得容易考虑。

下面的示例添加分配 CPU 内核和内存的根域 (`root-dom1`) 和来宾域 (`guest1`)。

```
# ldm add-domain root-dom1
# ldm add-domain guest1
# ldm set-core 16 root-dom1
# ldm set-core 32 guest1
# ldm set-memory 32G root-dom1
# ldm set-memory 64G guest1
```

注 - 若需在 Oracle VM Server for SPARC 的版本早于 3.2 时执行物理分区动态重新配置，请将需要分配给各逻辑域的内存容量设置为“CPU 内核数量 x 256 MB 的倍数”。

c. 将根合成体分配给根域。

将添加的 SPARC M10-4S 的根合成体分配给添加的根域 (`root-dom1`)。执行 `ldm list-io` 命令来检查需要添加到根域 (`root-dom1`) 的根合成体。NAME 以 “/BB1” 开始的设备在 BB-ID 1 上，该设备的 BUS 中显示的字符串表示根合成体。

```
# ldm list-io
NAME                                     TYPE    BUS      DOMAIN    STATUS
----                                     -
PCIE0                                  BUS     PCIE0    primary   IOV
PCIE1                                  BUS     PCIE1    root-dom0 IOV
PCIE2                                  BUS     PCIE2    root-dom0 IOV
PCIE3                                  BUS     PCIE3    root-dom0 IOV
PCIE4                                  BUS     PCIE4    primary   IOV
PCIE5                                  BUS     PCIE5    root-dom0 IOV
PCIE6                                  BUS     PCIE6    root-dom0 IOV
PCIE7                                  BUS     PCIE7    root-dom0 IOV
PCIE8                                  BUS     PCIE8
PCIE9                                  BUS     PCIE9
PCIE10                                 BUS     PCIE10
PCIE11                                 BUS     PCIE11
PCIE12                                 BUS     PCIE12
PCIE13                                 BUS     PCIE13
PCIE14                                 BUS     PCIE14
PCIE15                                 BUS     PCIE15
....
/BB1/CMUL/NET0                         PCIE    PCIE8
/BB1/CMUL/SASHBA                       PCIE    PCIE8
/BB1/PCI0                              PCIE    PCIE9
/BB1/PCI3                              PCIE    PCIE10
/BB1/PCI4                              PCIE    PCIE10
/BB1/PCI7                              PCIE    PCIE11
/BB1/PCI8                              PCIE    PCIE11
/BB1/CMUL/NET2                         PCIE    PCIE12
```

/BB1/PCI1	PCIE	PCIE13
/BB1/PCI2	PCIE	PCIE13
/BB1/PCI5	PCIE	PCIE14
/BB1/PCI6	PCIE	PCIE14
/BB1/PCI9	PCIE	PCIE15
/BB1/PCI10	PCIE	PCIE15

执行 `ldm add-io` 命令来添加需要添加到根域 (root-dom1) 的根合成体。然后，执行 `ldm list-io` 命令来实施检查。

下面的示例将除上述板载设备（/BB1/CMUL/NET0、/BB1/CMUL/SASHBA 和 /BB1/CMUL/NET2）的根合成体（PCIE8 和 PCIE12）之外的 BB1 根合成体（PCIE9、PCIE10、PCIE11、PCIE13、PCIE14 和 PCIE15）添加到 root-dom1。

```
# ldm add-io PCIE9 root-dom1
# ldm add-io PCIE10 root-dom1
....
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
PCIE0	BUS	PCIE0	primary	IOV
PCIE1	BUS	PCIE1	root-dom0	IOV
PCIE2	BUS	PCIE2	root-dom0	IOV
PCIE3	BUS	PCIE3	root-dom0	IOV
PCIE4	BUS	PCIE4	primary	IOV
PCIE5	BUS	PCIE5	root-dom0	IOV
PCIE6	BUS	PCIE6	root-dom0	IOV
PCIE7	BUS	PCIE7	root-dom0	IOV
PCIE8	BUS	PCIE8	root-dom1	IOV
PCIE9	BUS	PCIE9		
PCIE10	BUS	PCIE10	root-dom1	IOV
PCIE11	BUS	PCIE11	root-dom1	IOV
PCIE12	BUS	PCIE12		
PCIE13	BUS	PCIE13	root-dom1	IOV
PCIE14	BUS	PCIE14	root-dom1	IOV
PCIE15	BUS	PCIE15	root-dom1	IOV

- d. 将根合成体分配给根域。
 执行 `ldm set-vconsole` 命令将控制台设备分配给根域 (root-dom1)。

```
# ldm set-vconsole port=5001 vcc0 root-dom1
```

执行 `ldm bind-domain` 和 `ldm start-domain` 命令，将已配置的根合成体等信息再次绑定到根域 (root-dom1)，然后启动根域。

```
# ldm bind-domain root-dom1
# ldm start-domain root-dom1
LDom root-dom1 started
```

执行 `telnet` 命令连接到根域的控制台。

```
# telnet localhost 5001
....
```

e. 在根域中安装 **Oracle Solaris**。

在已启动的根域中安装 Oracle Solaris。有关如何安装 Oracle Solaris 的详情，请参见相关 Oracle Solaris 的文档。

f. 检查需要分配给根域的虚拟服务的物理设备。

安装 Oracle Solaris 后，登录根域，然后执行 `format` 命令来检查需要分配给虚拟磁盘服务的设备的名称。在登录根域显示与 PCIe 端点关联的设备路径之前，在控制域上执行 `ldm list-io -l` 命令。因此，您可以通过将该设备路径的名称与使用 `format` 命令显示的磁盘设备路径的名称进行比较来检查该磁盘所连接的路径。

```
# ldm list-io -l
NAME                                     TYPE      BUS      DOMAIN    STATUS
----
....
/BB1/PCI0                               PCIE      PCIE9    root-doml0CC
[pci@8900/pci@4/pci@0/pci@0]
/BB1/PCI3                               PCIE      PCIE10   root-doml0CC
[pci@8a00/pci@4/pci@0/pci@0]
/BB1/PCI4                               PCIE      PCIE10   root-doml0CC
[pci@8a00/pci@4/pci@0/pci@8]....
# telnet localhost 5000
....
root-dom1# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c3t500000E01BDA70B2d0 <FUJITSU-MBB2147RC-3703 cyl 14087 alt 2 hd 24
sec 848>
    / pci@8900/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w500000e01bda70b2,0
  1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
    /pci@8a00/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
....
Specify disk (enter its number): ^C
```

随后，执行 `dladm show-phys -L` 命令来检查从根域的顶层查看的以太网接口名称和物理位置。

```
root-dom0# dladm show-phys -L
LINK      DEVICE      LOC
net0      igb0        /BB1/PCI0
net1      igb1        /BB1/PCI0
net2      igb2        /BB1/PCI0
net3      igb3        /BB1/PCI0
```

g. 将虚拟 I/O 服务分配给根域。

返回控制域，以便将虚拟 I/O 服务添加到根域 (root-dom1)。有关 I/O 设备设

置的详情，请参见 Oracle 公司提供的《Oracle VM Server for SPARC Administration Guide》。

下面的示例将虚拟磁盘服务 (vds1) 和虚拟交换机网络接口 (vsw10和vsw11) 添加到已添加的根域 (root-dom1)。

```
# ldm add-vdiskserver vds1 root-dom1
# ldm add-vdiskserverdevice /dev/dsk/cXtXXXXXXXXXXXXdXsX vol0@vds1
# ldm add-vdiskserverdevice /dev/dsk/cXtXXXXXXXXXXXXdXsX vol1@vds1
# ldm add-vswitch net-dev=net1 vsw10 root-dom1
# ldm add-vswitch net-dev=net2 vsw11 root-dom1
```

h. 将虚拟 I/O 设备分配给来宾域。

将使用已添加虚拟 I/O 服务的虚拟 I/O 设备添加到各来宾域。

下面的示例将使用 root-dom1 的虚拟 I/O 服务的虚拟 I/O 设备 (vdisk10, vnet10) 添加到现有的来宾域 (guest0)。也将使用 root-dom0 和 root-dom1 的虚拟 I/O 服务的虚拟 I/O 设备 (vdisk1、vdisk11、vnet1 和 vnet11) 以及虚拟控制台 (vcons) 添加到需要新添加的来宾域 (guest1)。

```
# ldm add-vdisk vdisk10 vol0@vds1 guest0
# ldm add-vnet vnet10 vsw10 guest0
# ldm add-vdisk vdisk1 vol1@vds0 guest1
# ldm add-vdisk vdisk11 vol11@vds1 guest1
# ldm add-vnet vnet1 vsw1 guest1
# ldm add-vnet vnet11 vsw11 guest1
# ldm set-vconsole port=5101 guest1
```

i. 启动添加的来宾域。

执行 ldm bind-domain 和 ldm start-domain 命令启动添加的来宾域 (guest1)。

```
# ldm bind-domain guest1
# ldm start-domain guest1
LDom guest1 started
```

j. 在已添加的来宾域中安装 Oracle Solaris。

在已添加的来宾域中安装 Oracle Solaris。有关如何安装 Oracle Solaris 的详情，请参见相关 Oracle Solaris 的文档。

k. 创建各来宾域的虚拟 I/O 的冗余配置。

下面介绍使用 IPMP 创建具有分配给来宾域 guest0 的两个虚拟网络接口 (vnets) 的冗余配置步骤示例。有关其他冗余配置的步骤的详情，请参见各冗余配置软件的文档。

登录来宾域 guest0。在下面的示例中，使用 ldm list-domain 命令检查 guest0 的控制台的端口号，然后使用 telnet 命令连接到端口号 5100。

```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   UART    16    14G     0.0%  8h 7m
guest0        active    -n----   5100    32    32G     0.0%  20s
guest1        active    -n----   5101    32    32G     0.0%  19s
root-dom0     active    -n--v-   5000    24    24G     0.0%  43s
root-dom1     active    -n--v-   5001    24    24G     0.0%  20s

# telnet localhost 5100
....
guest0 console login: root
Password:
...
guest0#
```

执行 `dladm` 命令确认虚拟网络设备可见。在下面的示例中，作为网络接口（`net0` 和 `net1`）可以参照虚拟网络设备。

```
guest0# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      up        0      unknown  vnet0
net1          Ethernet      up        0      unknown  vnet1
```

执行 `ipadm show-if` 命令确认 `net0` 和 `net1` 未显示。

```
guest0# ipadm show-if
IFNAME      CLASS    STATE    ACTIVE OVER
lo0         loopback ok      yes    --
```

执行 `ipadm create-ip` 命令创建 IP 接口 `net0` 和 `net1`，然后使用 `ipadm show-if` 命令确认已经将其正常创建。

```
guest0# ipadm create-ip net0
guest0# ipadm create-ip net1
guest0# ipadm show-if
IFNAME      CLASS    STATE    ACTIVE OVER
lo0         loopback ok      yes    --
net0        ip       down     no     --
net1        ip       down     no     --
```

执行 `ipadm create-ipmp` 命令创建 IPMP 接口 `ipmp0`，然后执行 `ipadm add-ipmp` 命令将 IP 接口 `net0` 和 `net1` 添加到 IPMP 组。

```
guest0# ipadm create-ipmp ipmp0
guest0# ipadm add-ipmp -i net0 -i net1 ipmp0
```

执行 `ipadm create-addr` 命令将 IP 地址分配给 IPMP 接口 `ipmp0`，然后执行 `ipadm show-addr` 命令检查此设置。下面的示例分配固定 IP 地址。

```
guest0# ipadm create-addr -T static -a local=xx.xx.xx.xx/24 ipmp0/v4
guest0# ipadm show-addr
ADDROBJ          TYPE      STATE      ADDR
lo0/v4           static    ok         127.0.0.1/8
ipmp0/v4         static    ok         xxx.xxx.xxx.xxx/24
lo0/v6           static    ok         ::1/128
```

执行 ipadm set-ifprop 命令设置备用接口，然后执行 ipmpstat -i 命令检查 IPMP 配置。

```
guest0# ipadm set-ifprop -p standby=on -m ip net1
guest0 # ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net1       no      ipmp0      is-----  up        disabled   ok
net0       yes     ipmp0      --mbM--    up        disable
```

对其他来宾域（本示例中为 guest1）执行相同的步骤。

1.创建控制域系统卷的冗余配置。

添加的 SPARC M10-4S 的板载设备被分配给控制域，以便创建冗余配置。将控制域切换到延迟重新配置模式。

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

将已添加的 SPARC M10-4S 的根合成体（PCIE8 和 PCIE12）添加到控制域，然后重新启动 Oracle Solaris。

```
# ldm add-io PCIE8 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm add-io PCIE12 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# shutdown -i6 -g0 -y
....
```

重新启动 Oracle Solaris 后，创建系统卷的冗余配置。有关如何设置冗余配置的详情，请参见各冗余配置软件相关的文档。下面的示例使用 format 命令检查磁盘驱动器，然后创建具有 ZFS 的镜像配置。

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c2t50000393E802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
    /pci@8000/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
    /dev/chassis/FUJITSU-BBEXP.500000e0e06d027f/0123_HDD00/disk
  1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
    /pci@8800/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
    /dev/chassis/FUJITSU-BBEXP.500000e0e06d243f/022U_HDD01/disk
Specify disk (enter its number): ^C
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:

        NAME                                STATE      READ WRITE CKSUM
        rpool                                ONLINE      0     0     0
          c2t50000393E802CCE2d0s0            ONLINE      0     0     0

errors: No known data errors
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting.
```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。使用 `zpool status` 命令来确认同步处理 (resilver) 是否已经完成。下面的示例表明，同步处理正在进行中。

```
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
        Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Jan 27 15:55:47 2014
      21.1G scanned out of 70.6G at 120M/s, 0h7m to go
      21.0G resilvered, 29.84% done
config:

        NAME                                STATE      READ WRITE CKSUM
        rpool                                DEGRADED      0     0     0
          mirror-0                            DEGRADED      0     0     0
            c2t50000393E802CCE2d0s0            ONLINE      0     0     0
            c3t50000393A803B13Ed0s0            DEGRADED      0     0     0 (resilvering)

errors: No known data errors
```

同步处理完成后，将显示如下画面：

```
# zpool status rpool
pool: rpool
state: ONLINE
  scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:

    NAME                                STATE        READ  WRITE CKSUM
    rpool                                ONLINE       0     0     0
      mirror-0
        c2t50000393E802CCE2d0s0        ONLINE       0     0     0
        c3t50000393A803B13Ed0s0        ONLINE       0     0     0

errors: No known data errors
```

m. 检查并调整各逻辑域的内存配置。

执行 `ldm list-devices -a memory` 命令检查内存块的配置。

```
# ldm list-devices -a memory
MEMORY
    PA                                SIZE        BOUND
    0x7000000000000000                32G         root-dom1
    0x7200000000000000                32G         guest1
    0x7400000000000000                32G         guest1
    0x7600008000000000                1272M       _sys_
    0x7600500000000000                16G         primary
    0x7604500000000000                10G         primary
    0x7606d00000000000                4864M
    0x7800000000000000                32G         guest0
    0x7a00000000000000                32G         guest0
    0x7c00000000000000                256M        primary
    0x7c00100000000000                3584M       root-dom0
    0x7c00f00000000000                24832M      primary
    0x7c07000000000000                4G          root-dom0
    0x7e00008000000000                1272M       _sys_
    0x7e00500000000000                512M        _sys_
    0x7e00700000000000                256M        _sys_
    0x7e00800000000000                25088M      root-dom0
    0x7e06a00000000000                3584M       primary
    0x7e07800000000000                2G          primary
```

假设分配给逻辑域的内存块和未分配的内存块被分割成小碎片。您可以通过取消绑定逻辑域，然后再次将其绑定来将零散的内存块集合成大的连续区域。这将有利于以后添加或删除 SPARC M10-4S。有关步骤的详情，请参见 "[A.2.2 物理分区配置步骤示例](#)" 中的步骤 11-j。

n. 保存逻辑域配置信息。

执行 `ldm list-domain` 命令检查配置的逻辑域是否已经启动。


```
# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   UART    64      56G       64%     2h 54m
guest0        active    -n----   5100    64      64G       42%     2h 54m
guest1        active    -n----   5101    64      64G       11%     2h 54m
root-dom0     active    -n--v-   5000    32      32G       20%     2h 54m
root-dom1     active    -n--v-   5001    32      32G       18%     2h 54m
```

执行 `ldm add-spconfig` 命令将逻辑域配置信息保存到 XSCF。

```
# ldm add-spconfig ldm-set4
# ldm list-spconfig
factory-default
ldm-set1
ldm-set2
ldm-set3
ldm-set4 [current]
```

- o. 如果存在多项逻辑域配置信息需要恢复，请将当前配置恢复到出厂默认配置，然后重复步骤 15 和以后的步骤。

此项目将说明将当前配置恢复到出厂默认配置的步骤示例。从 XSCF 执行 `showdomainconfig` 命令检查需要启动的逻辑域的配置信息。

```
XSCF> showdomainconfig -p 0
PPAR-ID :0
Booting config
(Current) :ldm-set1
(Next) :ldm-set1
-----
Index :1
config_name :factory-default
domains :1
date_created:-
-----
Index :2
config_name :ldm-set1
domains :8
date_created:'2012-08-0811:34:56'
-----
....
```

执行 `setdomainconfig` 命令在物理分区编号 0 (PPAR ID 0) 中指定出厂默认 Index1。

```
XSCF> setdomainconfig -p 0 -i 1
```

执行 `poweroff` 命令，然后执行 `poweron` 命令重新启动物理分区。启动完成后，重复步骤 15 和以后的步骤。

```
XSCF> poweroff -p 0
...
XSCF> poweron -p 0
...
```

A.6 仅配置控制域的系统板的活动状态更换 （适用于 Oracle VM Server for SPARC 3.2 或更高版本）

在 Oracle VM Server for SPARC 3.2 或更高版本上可进行动态 PCIe 总线分配。因此，您无需延迟重新配置即可在控制域和根域上添加或删除 PCIe 根合成体。

因此，如果仅配置控制域（出厂默认配置），您无需停止或重新启动 Oracle Solaris 即可添加或删除系统板。

本节介绍释放设置为出厂默认配置的具有两个 SPARC M10-4S 的物理分区内的其中一个 SPARC M10-4S 的步骤示例。请注意：如果被设置为出厂默认设置，则所有的 CPU 内核和内存都被分配给控制域。本介绍也适用于 SPARC M12。

注 - 如果控制域或根域是 Oracle Solaris 10，由于禁用动态 PCIe 总线添加和删除，则此配置不可用。

A.6.1 配置示例

本节介绍由两个 SPARC M10-4S 即两个系统板 (PSB<BB>)（2BB 配置）组成控制域的配置示例。

如需启用物理分区动态重新配置，必须通过连接各 SPARC M10-4S 的物理 I/O，为系统卷磁盘和控制域的网络接口建立冗余配置。这样操作的目的是：当一个 SPARC M10-4S 被移除时，使用剩余的 SPARC M10-4S 的物理 I/O 可继续操作。

下面是满足上述条件的配置的示意图。而且，简化了各个逻辑域的 I/O 配置。

图 A-8 分配所有资源的 2BB 配置的配置示例

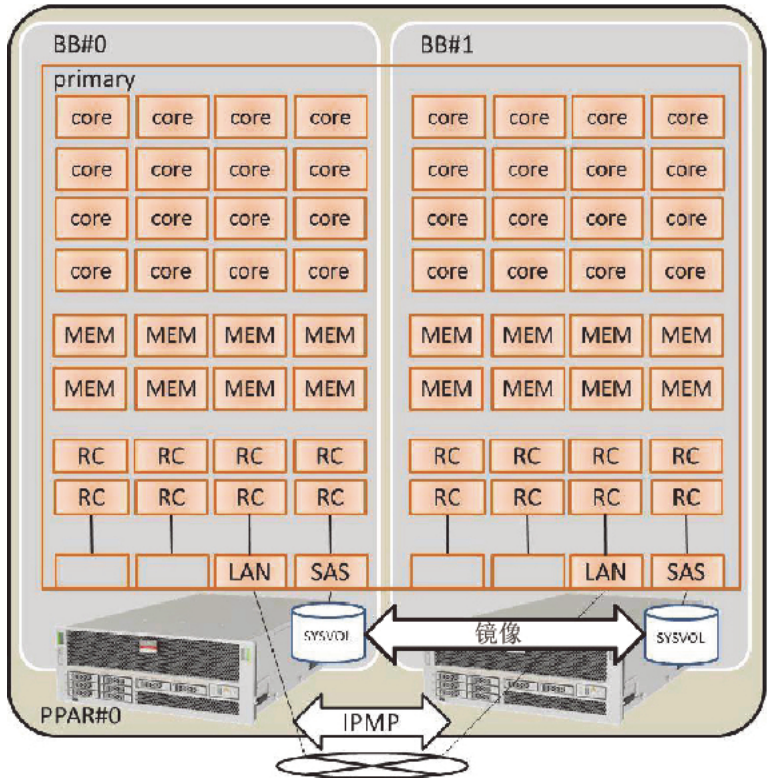


图 A-8 显示了将所有 CPU 内核、内存和物理 I/O 分配给控制域的 2BB 配置的配置示例。因此，搭载在通过物理分区动态重新配置将被断开的 SPARC M10-4S 上的 CPU 内核和内存将要被删除，无法在 SPARC M10-4S 释放时被移动。

A.6.2 物理分区配置步骤示例

1. 登录主 **XSCF**。
执行 `showbbstatus` 命令来检查您已经登录的 XSCF 是否是主 XSCF。
如果您已经登录到备用 XSCF，注销后再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

2. 设置内存的镜像模式。
 - a. 执行 `showfru` 命令来检查 **SPARC M10-4S** 系统板 (PSB<BB>) 上内存的镜像模式。
下面是显示物理系统板 (PSB 00-0) 下设备的设置信息示例。

```
XSCF> showfru sb 00-0
Device Location Memory Mirror Mode
sb 00-0
  cpu 00-0-0 no
  cpu 00-0-1 no
  cpu 00-0-2 no
  cpu 00-0-3 no
```

- b. 若需使用内存镜像模式，请执行 **setupfru** 命令来设置该模式。

如果您未正在使用内存镜像模式，将不需要此步骤。

有关内存镜像模式的详细信息，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "14.1.1 内存镜像概述"。

以下示例将系统板 (PSB 00-0) 下 CPU 的所有内存置于内存镜像模式。

```
XSCF> setupfru -m y sb 00-0
```

执行 **showfru** 命令检查内存镜像模式的设置。

```
XSCF> showfru sb 00-0
Device Location Memory Mirror Mode
sb 00-0
  cpu 00-0-0 yes
  cpu 00-0-1 yes
  cpu 00-0-2 yes
  cpu 00-0-3 yes
```

3. 创建物理分区配置信息。

- a. 执行 **showpctl** 命令检查物理分区配置信息。

```
XSCF> showpctl -p 0
PPAR-ID LSB PSB Status
```

- b. 执行 **setpctl** 命令在物理分区配置信息中登录系统板。

执行 **setpctl** 命令在内置目标的物理分区配置信息中登录 SPARC M10-4S 系统板。

在下面的示例中，物理系统板 (PSBs) 00-0 和 01-0 映射到物理分区 0 的逻辑系统板 (LSBs) 00 和 01。

```
XSCF> setpctl -p 0 -a 00=00-0 01=01-0
```

- c. 执行 **showpctl** 命令检查物理分区配置信息。

检查物理分区配置信息。

XSCF> showpcl -p 0			
PPAR-ID	LSB	PSB	Status
00			Running
	00	00-0	
	01	01-0	

指定 -v 选项显示配置策略、IO 无效选项 (No-IO) 和内存无效选项 (No-Mem) 的详细信息。

XSCF> showpcl -v -p 0						
PPAR-ID	LSB	PSB	Status	No-Mem	No-IO	Cfg-policy
00			Running			
						System
	00	00-0		False	False	
	01	01-0		False	False	

使用 setpcl 命令变更配置策略、I/O 无效选项 (No-IO) 和内存无效选项 (No-Mem) 的设置。

有关 setpcl 命令的详情，请参见《Fujitsu SPARC M12 and Fujitsu M10/ SPARC M10 XSCF Reference Manual》。

4. 将系统板 (**PSB<BB>**) 分配给物理分区。
 - a. 执行 **showboards-a** 命令检查 **PSB** 状态。

执行 showboards -a 命令检查各 PSB 的状态是否是 "SP"（系统板池）。

XSCF> showboards -a							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	SP	Available	n	n	n	Passed	Normal
01-0	SP	Available	n	n	n	Passed	Normal

- b. 执行 **addboard -c assign** 命令分配 **PSB**。

XSCF> addboard -c assign -p 0 00-0 01-0	
--	--

- c. 执行 **showboards-p** 命令检查 **PSB** 状态。

执行 showboards -p 命令检查分配给物理分区的各 PSB 的状态。

由于各 PSB 已经被正常分配给物理分区 0，此示例检查各 PSB 的 [Assignment] 字段是否成为 "Assigned"。

XSCF> showboards -p 0							
PSB	PPAR-ID(LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00(00)	Assigned	n	n	n	Passed	Normal
01-0	00(01)	Assigned	n	n	n	Passed	Normal

5. 登录 **CPU** 激活密钥以便分配 **CPU** 内核资源。

a. 执行 **showcodactivation** 命令，检查 CPU 激活密钥上的信息。

执行 **showcodactivation** 命令，检查物理分区是否包含可分配的 CPU 激活密钥。

如果仅显示标题，则 CPU 激活密钥未在 XSCF 中登录。

```
XSCF> showcodactivation
Index  Description  Count
-----  -
```

注 - 如果注册的 CPU 激活的数量对于将要使用的 CPU 的数量而言不足，则需购买 CPU 激活并添加 CPU 激活密钥。

b. 执行 **addcodactivation** 命令添加 CPU 激活密钥。

有关如何添加 CPU 激活密钥的详情，请参见《Fujitsu SPARC M12 和 Fujitsu M10/SPARC M10 系统 操作和管理指南》中的 "5.3 添加 CPU 内核资源"。

```
XSCF> addcodactivation "Product: SPARC M10-4S
SequenceNumber:10005
Cpu: noExpiration 2
Text-Signature-SHA256-RSA2048:
PSSrElBrse/r69AVSVFd38sT6AZm2bxUDdPQHKbtXgvZPsrtYguqiNUieB+mTDC
:
:
b1GCKFx1RH27FdVHiB2H0A=="
AboveKeywillbeadded,Continue?[y|n]:y
```

c. 执行 **showcodactivation** 命令，检查 CPU 激活密钥上的信息。

执行 **showcodactivation** 命令，检查物理分区是否包含可分配的 CPU 激活密钥。

```
XSCF> showcodactivation
Index  Description  Count
-----  -
      0  PROC           2
      1  PROC           2
      2  PROC           2
      3  PROC           2

---  省略  ---

     62  PROC           2
     63  PROC           2
```

d. 执行 **setcod** 命令分配 CPU 内核资源。

执行 **setcod** 命令将 CPU 内核资源分配给物理分区。

以下是将 128 个 CPU 内核资源分配到物理分区 0 的示例。

```
XSCF> setcod -p 0 -s cpu -c set 128
PROC Permits assigned for PPAR 0 : 0 -> 128

PROC Permits assigned for PPAR will be changed.
Continue? [y|n] :y

Completed.
```

注 -XCP 2250 和更早版本的 XSCF 固件不支持 **-c add**、**-c delete** 和 **-c set** 选项。按如下所示指定 **setcod** 命令的选项可交互式添加和删除 CPU 内核资源。

```
XSCF> setcod -s cpu
```

执行 **showcod** 命令检查分配给物理分区的 CPU 内核资源的信息。

下面的示例，使用刚刚执行的 **setcod** 命令，证实 128 个 CPU 内核资源被分配给物理分区 0。

```
XSCF> showcod -p 0
PROC Permits assigned for PPAR 0: 128
```

6. 执行 **resetdateoffset** 命令以 XSCF 管理的时间重置差异。

执行 **resetdateoffset** 命令来重置 XSCF 管理的时间和物理分区管理的时间之间的差异。

```
XSCF> resetdateoffset -p 0
```

7. 执行 **showpparmode** 命令来检查诊断消息的详细级别的设置以及 PPAR DR 模式的设置。

执行 **showpparmode** 命令来检查诊断信息的详细级别 (Message Level) 是否是 "normal" (标准) 以及 PPAR DR 模式的 Next 是否设置为 "on" (启用)。

```
XSCF> showpparmode -p 0
Host-ID                      :9007002b
Diagnostic Level              :min
Message Level                 :normal
Alive Check                  :on
Watchdog Reaction            :reset
Break Signal                  :on
Autoboot (Guest Domain)      :on
Elastic Mode                  :off
IOreconfigure                 :false
PPAR DR (Current)            :-
PPAR DR (Next)               :on
```

如果诊断信息的详细级别是除 "normal" 之外的级别, 执行 **setpparmode** 命令可将其设置为 "normal"。

有关 **setpparmode** 命令的详情, 请参见《Fujitsu SPARC M12 and Fujitsu M10/SPARC M10 XSCF Reference Manual》。

```
XSCF> setpparmode -p 0 -m message=normal
```

如果 PPAR DR 模式设置为 "off"（禁用），执行 setpparmode 命令可将其设置为 "on"。

```
XSCF> setpparmode -p 0 -m ppar_dr=on
```

8. 执行 **poweron** 命令来开启物理分区的电源。

```
XSCF> poweron -p 0
```

9. 执行 **console** 命令将控制台连接到物理分区。

```
XSCF> console -p 0
```

10. 安装 **Oracle Solaris** 和 **Oracle VM Server for SPARC**。

在控制域上安装 Oracle Solaris 和 Oracle VM Server for SPARC。

有关物理分区动态重新配置所要求的 Oracle Solaris 的版本和条件的详情，请参见表 1-3 和 表 1-4。

有关安装的详情，请参见显示在 Oracle 公司网站 (<http://docs.oracle.com/>) 上的以下文档。

- Oracle Solaris 11

《Installing Oracle Solaris 11.2 Systems》

- Oracle VM Server for SPARC

《Oracle VM Server for SPARC Administration Guide》中的 "Installing and Enabling Software"

11. 创建控制域的冗余物理 I/O 配置。

- a. 创建控制域系统卷的冗余配置。

创建控制域系统卷的冗余配置。此项目介绍使用 ZFS 创建镜像配置的命令示例。有关详情，请参见显示在 Oracle 公司网站 (http://docs.oracle.com) 上的以下文档。

《Oracle Solaris 11.2 Administration: ZFS File Systems》中的 "How to Configure a Mirrored Root Pool (SPARC or x86/VTOC)"

如需使用其他冗余配置软件，请参见该软件的手册。

执行 **zpool status** 命令检查根池的状态。

下面的示例表明已经将 c2t50000393E802CCE2d0s0 分配给默认根池 (rpool)。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: none requested
config:
    NAME                                STATE      READ WRITE CKSUM
```



```

rpool                                ONLINE          0      0      0
c2t50000393E802CCE2d0s0 ONLINE          0      0      0
errors: No known data errors

```

执行 **format** 命令检查可添加哪个磁盘。

下面的示例表明 **c3t50000393A803B13Ed0** 作为其他磁盘而存在。

```

# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2t50000393E802CCE2d0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
    /pci@8000/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393e802cce2,0
    /dev/chassis/FUJITSU-BBEXP.500000e0e06d027f/0123_HDD00/disk
  1. c3t50000393A803B13Ed0 <TOSHIBA-MBF2300RC-3706 cyl 46873 alt 2 hd 20
sec 625>
    /pci@8800/pci@4/pci@0/pci@0/scsi@0/iport@f/disk@w50000393a803b13e,0
    /dev/chassis/FUJITSU-BBEXP.500000e0e06d243f/022U_HDD01/disk
Specify disk (enter its number): ^C

```

执行 **zpool attach** 命令将第二个磁盘添加到 **rpool**，以便提供镜像配置。

下面的示例使用 **zpool attach** 命令添加 **c3t50000393A803B13Ed0s0**，然后使用 **zpool status** 命令检查同步处理 (resilver) 的状态。通过参照状态和动作，您可以确定正在进行同步处理。定期执行 **zpool status** 命令来检查同步处理的状态，直到处理结束。

```

# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered. The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Wed Jan 29 21:35:39 2014
  3.93G scanned out of 70.6G at 71.9M/s, 0h15m to go 3.90G resilvered, 5.56%
done
config:
    NAME                                STATE      READ  WRITE CKSUM
    rpool                                DEGRADED   0     0     0
      mirror-0                           DEGRADED   0     0     0
        c2t50000393E802CCE2d0s0          ONLINE    0     0     0
        c3t50000393A803B13Ed0s0          DEGRADED   0     0     0
(resilvering)

```

当完成同步处理时，[state] 将被设置为 "ONLINE"，如下所示。

```

# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h10m with 0 errors on Wed Jan 29 21:45:42 2014

```

config:					
NAME	STATE	READ	WRITE	CKSUM	
rpool	ONLINE	0	0	0	
mirror-0	ONLINE	0	0	0	
c2t50000393E802CCE2d0s0	ONLINE	0	0	0	
c3t50000393A803B13Ed0s0	ONLINE	0	0	0	
errors: No known data errors					

b. 创建控制域网络的冗余配置。

执行 `dladm` 命令确认网络设备可见。下面的示例表明可以把板载网络接口称为 `net0` 至 `net7`。

# <code>dladm show-phys -L</code>		
LINK	DEVICE	LOC
net0	igb0	/BB0/CMUL
net1	igb1	/BB0/CMUL
net2	igb6	/BB0/CMUU
net3	igb7	/BB0/CMUU
net4	igb14	/BB1/CMUL
net5	igb15	/BB1/CMUL
net6	igb12	/BB1/CMUU
net7	igb13	/BB1/CMUU

下面介绍使用IPMP创建 BB0 的网络接口 `net0` 和 BB1 的网络接口 `net4` 的冗余配置的示例。
首先，执行 `ipadm show-if` 命令确认 `net0`（位于 BB0 上）和 `net4`（位于 BB1 上）未显示。

# <code>ipadm show-if</code>				
IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--

执行 `ipadm create-ip` 命令创建 IP 接口 `net0` 和 `net4`，然后执行 `ipadm show-if` 命令确认已经将其正常创建。

# <code>ipadm create-ip net0</code>				
# <code>ipadm create-ip net4</code>				
# <code>ipadm show-if</code>				
IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--
net0	ip	down	no	--
net4	ip	down	no	--

执行 `ipadm create-ipmp` 命令创建 IPMP 接口 `ipmp0`，然后执行 `ipadm add-ipmp` 命令将 IP 接口 `net0` 和 `net4` 添加到 IPMP 组。

# <code>ipadm create-ipmp ipmp0</code>	
# <code>ipadm add-ipmp -i net0 -i net4 ipmp0</code>	

执行 `ipadm create-addr` 命令将 IP 地址分配给 IPMP 接口 `ipmp0`，然后执行 `ipadm`

show-addr 命令检查此设置。在下面的示例中，分配了固定 IP 地址。

```
# ipadm create-addr -T static -a local=xx.xx.xx.xx/24 ipmp0/v4
# ipadm show-addr
ADDROBJ          TYPE      STATE      ADDR
lo0/v4            static    ok         127.0.0.1/8
ipmp0/v4          static    ok         xx.xx.xx.xx/24
lo0/v6            static    ok         ::1/128
```

执行 ipadm set-ifprop 命令来设置备用接口，并执行 ipmpstat -i 命令检查 IPMP 配置。

```
# ipadm set-ifprop -p standby=on -m ip net4
# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net4       no      ipmp0      is-----  up        disabled   ok
net0       yes     ipmp0      --mbM--    up        disabled   ok
```

A.6.3 活动状态更换的步骤示例

本节介绍使用 "图 A-8 分配所有资源的 2BB 配置的配置示例" 所述的 2BB 配置系统的 PPAR DR 活动状态更换 BB#01 的步骤示例。此示例适用于可使用动态 PCIe 总线分配的环境（Oracle VM Server for SPARC 3.2 或更高版本）。

本介绍也适用于 SPARC M12。

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。
在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

- 1. 登录主 XSCF。
执行 showbbstatus 命令来检查您已经登录的 XSCF 是否是主 XSCF。
如果您已经登录到备用 XSCF，注销然后再次登录到主 XSCF。

```
XSCF> showbbstatus
BB#00 (Master)
```

- 2. 执行 showhardconf 命令检查需要更换的 SPARC M10-4S 中的 XSCF 的 [Status] 是否是 "Normal"。

```
XSCF> showhardconf
SPARC M10-4S;
+ Serial: 2081230011; Operator_Panel_Switch:Locked;
+ System_Power:On; System_Phase:Cabinet Power On;
Partition#0 PPAR_Status:Running;
BB#00 Status:Normal; Role:Master; Ver:2003h; Serial:2081231002;
+ FRU-Part-Number: CA07361-D202 A1;
```

```

+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP1236052K ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
CPU#0 Status:Normal; Ver:4142h; Serial:00322658;
+ Freq:3.000 GHz; Type:0x10;
+ Core:16; Strand:2;
:
BB#01 Status:Normal; Role:Standby; Ver:0101h;Serial:7867000297;
+ FRU-Part-Number: CA20393-B50X A2 ;
+ Power_Supply_System: ;
+ Memory_Size:256 GB;
CMUL Status:Normal; Ver:0101h; Serial:PP123406CB ;
+ FRU-Part-Number:CA07361-D941 C4 /7060911 ;
+ Memory_Size:128 GB; Type: A ;
:

```

注 - 如果需要活动状态更换的 SPARC M10-4S 中的 XSCF 存在缺陷，您将无法使用 PPAR DR 执行活动状态更换。

在进行维护作业时，必须停止需要活动状态更换的 SPARC M10-4S 所属的物理分区、切断需要更换的 SPARC M10-4S 的输入电源。

3. 执行 **showbbstatus** 命令来确认需要更换的 **SPARC M10-4S** 中的 **XSCF** 不是主 **XSCF**。

```

XSCF> showbbstatus
BB#00 (Master)

```

如果需要更换的 SPARC M10-4S 是主 XSCF，则执行 **switchscf** 命令切换 XSCF。

```

XSCF> switchscf -t Standby
The XSCF unit switch between the Active and Standby states.
Continue? [y|n] :y

```

注 - 释放 SPARC M10-4S 前，请确认 XSCF 已经切换并重新启动。

4. 执行 **console** 命令连接到控制域的控制台，然后登录。

```

XSCF> console -p 0

```

5. 释放控制域中系统卷和物理 I/O 设备的冗余配置。

释放需要更换的 SPARC M10-4S（位于 BB#01 上）的物理 I/O 设备（在控制域中使用的）。有关取消冗余配置的步骤的详情，请参见该冗余配置软件的文档。

a. 取消控制域中系统卷的冗余配置。

下面的示例介绍如何取消控制域中系统卷的 ZFS 镜像功能。

在控制域中执行 `zpool status` 命令检查镜像配置状态。

```
# zpool status rpool
pool: rpool
state: ONLINE
  scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
  NAME                                STATE      READ  WRITE CKSUM
  rpool                               ONLINE    0     0     0
    mirror-0                          ONLINE    0     0     0
      c2t50000393E802CCE2d0s0        ONLINE    0     0     0
      c3t50000393A803B13Ed0s0        ONLINE    0     0     0
errors: No known data errors
```

执行 `zpool detach` 命令从镜像配置中释放磁盘。

```
# zpool detach rpool c3t50000393A803B13Ed0
```

执行 `zpool status` 命令确认镜像配置已经取消。

```
# zpool status rpool
pool: rpool
state: ONLINE
  scan: resilvered 28.7M in 0h0m with 0 errors on Tue Jan 21 10:10:01 2014
config:
  NAME                                STATE      READ  WRITE CKSUM
  rpool                               ONLINE    0     0     0
    mirror-0                          ONLINE    0     0     0
      c2t50000393E802CCE2d0s0        ONLINE    0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则移除冗余配置或停止使用那些设备。有关如何取消冗余配置或停止使用设备的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

b. 取消控制域网络的冗余配置。

执行 `ipmpstat -i` 命令检查配置 IPMP 的网络接口的配置信息。

```
# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
net0       yes     ipmp0      -smbM--    up        disabled   ok
net4       no      ipmp0      is-----  up        disabled   ok
```

执行 `if_mpadm -d net4` 命令从 IPMP 组释放 `net4`，然后执行 `ipmpstat -i` 命令确认已经将其释放。下面的示例确认 STATE 是 `offline`。

```
# if_mpadm -d net4
# ipmpstat -i
INTERFACE  ACTIVE  GROUP      FLAGS      LINK      PROBE      STATE
```

net0	yes	ipmp0	-smbM--	up	disabled	ok
net4	no	ipmp0	-s---d-	up	disabled	offline

6. 从物理分区释放 **SPARC M10-4S** 的系统板 (**PSB<BB>**)。
 - a. 执行 **deleteboard -c disconnect** 命令将 **PSB** 从物理分区释放。
 如果您尚未提前手动释放任何 CPU 内核、内存和 PCIe 根合成体资源，确保指定 **"-m unbind=resource"** 选项。当指定了此选项时，资源被自动删除，然后 **PSB** 被释放。

```
XSCF> deleteboard -c disconnect -m unbind=resource 01-0
PSB#01-0 will be unconfigured from PPAR immediately.
Continue?[y|n] :y
Start unconfigure preparation of PSB. [1200sec]
0end
Unconfigure preparation of PSB has completed.
Start unconfiguring PSB from PPAR. [7200sec]
0..... 30..... 60....end
Unconfigured PSB from PPAR.
PSB power off sequence started. [1200sec]
0..... 30..... 60..... 90.....120.....150.....end
Operation has completed.
```

- b. 执行 **showresult** 命令检查刚刚执行的 **deleteboard** 命令的结束状态。
 结束值 0 表明 **deleteboard** 命令正常终止。
 如果结束值非 0，或如果在执行 **deleteboard** 命令时显示错误信息，则表明 **deleteboard** 命令异常终止。根据错误信息，通过参考 **"C.1.2 deleteboard"** 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

- c. 执行 **showboards** 命令检查 **PSB** 状态。
 确认需要更换的 **SPARC M10-4S** 内的 **PSB** 是否处于 **"Assigned"** 状态，以及 **[Pwr]**、**[Conn]** 和 **[Conf]** 栏是否全显示 **"n"**。

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB)  Assignment  Pwr  Conn  Conf  Test  Fault
-----
00-0  00(00)         Assigned    y    y     y     Passed Normal
01-0  00(01)         Assigned    n    n     n     Passed Normal
```

7. 执行 **replacefru** 命令来更换 **SPARC M10-4S**。

```
XSCF> replacefru
```

注 - 关于使用 **replacefru** 命令更换 **SPARC M10-4S** 的详情，请参见《Fujitsu M10-4/Fujitsu

8. 将 **PSB** 整合到物理分区。

a. 执行 **showboards** 命令检查 **PSB** 状态。

确认已更换的 SPARC M10-4S 内的 PSB 是否处于 Assigned 状态，以及 [Pwr]、[Conn] 和 [Conf] 栏是否全显示 "n"。

XSCF> showboards -p 0							
PSB	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
-----	-----	-----	-----	-----	-----	-----	-----
00-0	00 (00)	Assigned	y	y	y	Passed	Normal
01-0	00 (01)	Assigned	n	n	n	Passed	Normal

b. 执行 **addboard -c configure** 命令将 **PSB** 整合到物理分区。

若需恢复初始逻辑域配置，请执行指定 -m bind=resource 选项的 addboard -c configure 命令。

```
XSCF> addboard -c configure -m bind=resource -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
0..... 30..... 60..... 90.....120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
810.....840.....870.....900.....930.....960.....end
Connected PSB to PPAR.
Start configuring PSB to Logical Domains (LDoms) Manager.
[1800sec] 0.....end
Configured PSB to Logical Domains (LDoms) Manager.
Operation has completed.
```

注 - 如果执行 addboard 命令过程中出现错误信息，请参见 "C.1.1 addboard"，然后识别错误并采取应对措施。

c. 执行 **showresult** 命令检查刚刚执行的 **addboard** 命令的结束状态。

结束值 0 表明 addboard 命令正常终止。

如果结束值非 0，或如果在执行 addboard 命令时显示错误信息，则表明 addboard 命令异常终止。根据错误信息，通过参考 "C.1.1 addboard" 识别错误，然后采取应对措施。

```
XSCF> showresult
0
```

d. 执行 **showboards** 命令检查 **PSB** 状态。

在已更换的 SPARC M10-4S 中的 PSB 已经被成功整合后，确认 [Conn] 和

[Conf] 栏是否都显示 "y"。

```
XSCF> showboards -p 0
```

PSB	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
00-0	00(00)	Assigned	y	y	y	Passed	Normal
01-0	00(01)	Assigned	y	y	y	Passed	Normal

9. 将控制域上的系统卷和物理 I/O 设备恢复到冗余配置。

a. 将控制域中的系统卷设置为冗余配置。

在控制域中执行 `zpool status` 命令检查镜像配置状态。

下面的示例介绍如何配置控制域中系统卷的 ZFS 镜像功能。

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 29.1M in 0h0m with 0 errors on Thu Jan 23 17:27:59 2014
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                               ONLINE    0     0     0
    c2t50000393E802CCE2d0s0          ONLINE    0     0     0
errors: No known data errors
```

执行 `zpool attach` 命令，以便将磁盘整合到镜像配置中。

```
# zpool attach rpool c2t50000393E802CCE2d0s0 c3t50000393A803B13Ed0s0
Make sure to wait until resilver is done before rebooting.
#
```

执行 `zpool status` 命令，然后确认镜像配置是否已经创建。

使用 `zpool status` 命令确认同步处理 (resilver) 是否完成。

下面显示同步处理过程中显示的示例。

```
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
       continue to function in a degraded state.
action: Wait for the resilver to complete.
       Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Mon Jan 27 15:55:47 2014
      21.1G scanned out of 70.6G at 120M/s, 0h7m to go
      21.0G resilvered, 29.84% done
config:
  NAME                                STATE      READ WRITE CKSUM
  rpool                               DEGRADED    0     0     0
    mirror-0                          DEGRADED    0     0     0
      c2t50000393E802CCE2d0s0          ONLINE    0     0     0
      c3t50000393A803B13Ed0s0          DEGRADED    0     0     0 (resilvering)
```



```
errors: No known data errors
```

同步处理完成后，将显示如下画面：

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: resilvered 70.6G in 0h9m with 0 errors on Mon Jan 27 16:05:34 2014
config:
  NAME                                STATE      READ  WRITE CKSUM
  rpool                               ONLINE    0     0     0
    mirror-0
      c2t50000393E802CCE2d0s0        ONLINE    0     0     0
      c3t50000393A803B13Ed0s0        ONLINE    0     0     0
errors: No known data errors
```

如果您正在使用 BB#01 中的其他设备，则创建冗余配置或恢复使用那些设备。有关如何创建冗余配置或恢复设备使用的详情，请参见该冗余配置软件的文档和 Oracle Solaris。

使用动态重新配置补充资料

本附录对使用物理分区动态重新配置功能，以及如何处理问题提供了补充资料。

- XSCF 重新启动或故障转移时间的注意事项
- CPU 运行模式补充资料
- 其他注意事项

B.1 XSCF 重新启动或故障转移时间的注意事项

如果当您通过添加或删除系统板 (PSB<BB>) 执行物理分区动态重新配置时发生 XSCF 重新启动或故障转移，相关的物理分区动态重新配置操作可能无法完成。在这种情况下，请执行以下操作，检查操作是否已经完成。如果操作尚未完成，请再次使用动态重新配置添加或移除 PSB (BB)。

检查添加操作的完成状况

1. 在 XSCF 上执行 **showboards** 命令检查系统板 (PSB<BB>) 状态。
下面的示例检查已安装的 PSB (BB) 是否处于 Assigned 状态，并且检查 [Pwr]、[Conn] 和 [Conf] 栏是否都显示 "y"。

XSCF> showboards -p PPAR_ID							
PSB	PPAR-ID (LSB)	Assignment	Pwr	Conn	Conf	Test	Fault
----	-----	-----	----	----	----	-----	-----
00-0	00 (00)	Assigned	y	y	y	Passed	Normal
01-0	00 (01)	Assigned	y	y	y	Passed	Normal

2. 执行 **console** 命令连接到控制域的控制台，然后登录。

```
XSCF> console -p PPAR_ID
```

3. 执行 **ldm list-devices** 命令检查已添加的硬件资源的状态。
执行 **ldm list-devices** 命令后，将列出未分配的硬件资源。
检查是否已添加硬件资源。

```
# ldm list-devices
CORE
  ID      %FREE    CPUSSET
  128      100      (256, 257)
  132      100      (264, 265)
  136      100      (272, 273)
  140      100      (280, 281)
  ....
MEMORY
  PA      SIZE      BOUND
  0x7000000000000000 32G
  0x7200000000000000 32G
  0x7400000000000000 32G
  0x7600500000000000 31488M
  ....
```

检查移除操作的完成状况

执行 **showboards** 命令检查系统板 (PSB<BB>) 状态。
检查已移除的 PSB (BB) 是否处于 **Assigned** 状态，并且检查 [Pwr]、[Conn] 和 [Conf] 栏是否都显示 "n"。或者检查已移除的系统板是否处于 **"SP"**（系统板池）状态和 **"Available"** 状态，并且检查 [Pwr]、[Conn] 和 [Conf] 栏是否都显示 "n"。

下面的示例显示各个检查。

```
XSCF> showboards -p PPAR_ID
PSB  PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test  Fault
-----
00-0 00(00)      Assigned    y    y    y    Passed Normal
01-0 00(01)      Assigned    n    n    n    Passed Normal
```



```
XSCF> showboards -p PPAR_ID
PSB  PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test  Fault
-----
00-0 00(00)      Assigned    y    y    y    Passed Normal
01-0 SP          Available    n    n    n    Passed Normal
```

B.2 CPU 运行模式补充资料

如果由 `setpparmode` 命令设置的 CPU 运行模式与用于启动物理分区的逻辑域配置信息不匹配，则 XSCF 将逻辑域配置信息切换到出厂默认设置。
下表显示了指定的物理分区配置、CPU 运行模式设置和可将信息切换到出厂默认设置的逻辑域配置信息的组合。

注 - SPARC M12 系统不支持 CPU 运行模式。

表 B-1 CPU 配置、CPU 运行模式和将信息切换到出厂默认设置的逻辑域配置信息之间的关系

物理分区 CPU 配置	CPU 运行模式设置	逻辑域配置信息
所有 SPARC64 X+ 处理器	compatible	指定在运行期间使用 <code>setdomainconfig</code> 或 <code>ldm set-spconfig</code> 命令的 SPARC64 X+ 功能所创建的逻辑域配置信息
所有 SPARC64 X 处理器	auto 和 compatible	
混合使用 SPARC64 X+ 处理器和 SPARC64 X 处理器	auto 和 compatible	

下文显示了基于上表可将逻辑域配置信息切换到出厂默认设置的操作。

- 当仅配置 SPARC64 X+ 处理器的物理分区的 CPU 运行模式从 "auto" 变更为 "compatible"，然后启动物理分区时
- 当使用 `addboard -c assign` 命令将配置了 SPARC64 X 处理器的系统板添加到仅配置 SPARC64 X+ 处理器的物理分区，然后启动物理分区时

B.3 其他注意事项

- 当构成物理分区 (PPAR) 的所有系统板上安装 CPU 内存单元 (上部) (CMUU) 时，如果使用动态重新配置功能 (PPAR DR) 从 PPAR 配置中临时移除 PSB 以移除 CMUU，则当 PSB 返回时，CMUL 的一些 I/O 会不可见。如有必要，请重新设置逻辑域的配置信息。
- 如果执行 `deleteboard` 命令时发生异常结束，请勿执行 `addboard` 命令。否则，`addboard` 命令将检测到称为 "internal error" 的错误，造成异常结束。此命令可能也影响以后的操作。
- 如果在动态重新配置功能正在使用时以及使用 `deleteboard` 命令将 PSB (BB) 移除时发生异常结束，请执行以下步骤。
 - 1.根据错误信息确定错误原因。
 - 2.移除错误原因。
 - 3.使用动态重新配置功能并再次执行 `deleteboard` 命令从 PPAR 配置中删除 PSB。

如果删除 PSB 再次失败，请停止物理分区 (PPAR)，然后执行 ["6.3.1 删除系统板分配的操作示例"](#) 或 ["6.3.3 预约系统板取消分配的操作示例"](#) 中阐述的操作。

- 如果在使用动态重新配置功能，并执行 `addboard` 命令来添加 PSB (BB) 时发生异常结束，请执行以下步骤或操作：
 - 1.根据错误信息确定错误原因。
 - 2.移除错误原因。
 - 3.使用动态重新配置功能并再次执行 `addboard` 命令将 PSB 添加到 PPAR 配置。
 或者，使用动态重新配置功能，并执行 `deleteboard` 命令从 PPAR 配置中删除 PSB，然后再次执行 `addboard` 命令将 PSB 添加到 PPAR 配置。
 如果采用上述两种操作添加 PSB 再次失败，请停止物理分区 (PPAR)，然后执行 "6.2.1 系统板分配示例" 或 "6.2.3 预约系统板整合的操作示例" 中阐述的操作。
- 当您使用动态重新配置功能，以便使用 `addboard` 命令添加 PSB (BB) 或以便使用 `deleteboard` 命令删除 PSB 时，请勿停止或重新启动 Oracle VM Server for SPARC。否则，`addboard` 命令或 `deleteboard` 命令将异常结束。
 如果重新启动 Oracle VM Server for SPARC，然后重新运行 `addboard` 命令或 `deleteboard` 命令，异常结束可能再次发生。
 在这种情况下，如果添加系统板，请停止物理分区 (PPAR)，然后执行 "6.2.1 系统板分配示例" 或 "6.2.3 预约系统板整合的操作示例" 中阐述的操作。
 当移除系统板时，请停止物理分区，然后执行 "6.3.1 删除系统板分配的操作示例" 或 "6.3.3 预约系统板取消分配的操作示例" 中阐述的操作。
- 当您使用动态重新配置功能，以便使用 `addboard` 命令添加 PSB (BB) 或以便使用 `deleteboard` 命令删除 PSB 时，请勿使用 `rebootxscf -a` 命令或 XSCF Web 重新启动所有的 XSCF。由于物理分区正在运行时 Oracle Solaris 挂起、硬件故障，此操作不但可能导致 DR 处理的异常终止，而且可能导致物理分区的异常停止，因此可能需要更换部件。
- 如果使用动态重配置功能删除给内部硬盘的分配，可能会出现以下消息，并且删除处理可能会失败。

```
# ldm remove-io /BB0/CMUL/SASHBA primary
The removal of /BB0/CMUL/SASHBA from the domain primary failed.
Error message from svc:/ldoms/agents in domain primary:
ERROR: devices or resources are busy.
```

当fmd服务使用将被删除的设备可能会出现上述错误。在分配目标设备的域上向fmd 服务发出 `stop` 命令，检查服务是否已停止，然后再次执行删除处理。此外，在目标设备的分配被删除之后重新启动fmd服务。

使用下面的命令以停止/重新启动fmd服务。

- 停止fmd服务

```
# svcadm disable svc:/system/fmd:default
```

- 检查 fmd 服务是否已停止

fmd 服务可能需要几分钟才能停止。

发出 `stop` 命令后，检查 STATE 是否为 disabled，服务是否已停止。

```
# svcadm fmd
STATE STIME FMRI
disabled 00:00:00 svc:/system/fmd:default
```

- 重新启动fmd服务

```
# svcadm enable svc:/system/fmd:default
```

- 如果 Oracle Solaris 在分配了 SPARC M12-2S 板载 LAN 或双 10G 以太网卡 (SP1X7HF1F、SE1X7HE3G) 的逻辑域中暂停或恢复动态重新配置功能，则可能检测到 "NIC-8000-1C" 错误。忽略此错误，因为它不会影响该操作。
- 当使用 `deleteboard` 删除 PSB 时，或者从冗余配置中释放要删除的 PSB 的 LAN 或 I/O 设备，或者提前更改为未使用状态。
- 使用 `addboard` 命令添加 PSB 时，或者整合要添加的 PSB 的 LAN 或 I/O 设备，或者变更设置以便可重新使用。

信息含义及应对措施

本附录介绍域配置操作期间显示的信息的含义及应对措施。

- 命令信息

C.1 命令信息

本节介绍物理分区配置操作期间显示的各个信息的含义及应对措施。

C.1.1 addboard

A hardware error occurred.Please check the error log for details.

[含义] 由于硬件错误，命令终止。

[措施] 检查可能的故障位置的错误日志，并更换故障组件。

An internal error has occurred.Please contact your system administrator.

[含义] 发生内部错误。联系系统管理员。

[措施] XSCF 中正在使用的库发生错误或 XSCF 内部资源不足。稍等片刻，然后再次执行此操作。如果再次出现错误，请联系富士通现场工程师进行错误调查。

Another DR operation is in progress.Try again later.

[含义] 由于其他会话正在进行动态重新配置，无法访问指定的物理分区。

[措施] 其他会话正在进行动态重新配置。检查物理分区的状态，稍等片刻，然后再次执行此操作。

Cannot communicate with Hypervisor or Logical Domains Manager.Please check the domain's state.

[含义] 与虚拟机监控程序或 Logical Domains Manager（逻辑域管理器）的通讯发生错误。

[措施] 从输出的监控信息和控制台信息，确定逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。然后，消除原因，并再次执行此操作。

Cannot communicate with XXX.Please check XXX's state.

[含义] 备用或从 XSCF 发生错误。

[措施] 解决备用或从 XSCF 的错误。在单机配置的 SPARC M12/M10 系统中不会发生此错误。

Configured PSB to Logical Domains (LDoms) Manager.

[含义] 已经完成整合到 Logical Domains Manager（逻辑域管理器）。

Connected PSB to PPAR.

[含义] 已经完成整合到物理分区配置。

Control domain is stopped for CoD resource violation.

[含义] 由于 CPU 激活冲突导致控制域停止，指定的系统板无法整合到物理分区。

[措施] 解决 CPU 激活冲突，然后再次执行此操作。

DR operation canceled by operator.

[含义] 由于用户发出指令将其停止，动态重新配置被取消。

Initial diagnosis has completed.

[含义] 硬件诊断已经完成。

Initial diagnosis started.

[含义] 硬件诊断已经开始。

Invalid parameter.

[含义] 参数错误

[措施] 检查参数和操作数。

Operation has completed.

[含义] 整合已经完成。

Operation is continued as "no".

[含义] 由于响应 [y|n] 提示时，未执行正确的操作，执行了选择 [n] 时应用的操作。

[措施] 检查系统板 (PSB) 信息。如果存在需要整合的 PSB，再次将其整合。

Operation not supported on this system.

[含义] 系统不支持所输入的命令。

[措施] 检查型号和 XCP 版本，查明是否可以执行此命令。

Permission denied.

[含义] 您没有访问权限。或者，您试图在备用 XSCF 上执行不可执行的命令，导致执行权限错误。

[措施] 比较您的用户权限和此命令所要求的执行权限。

PPAR DR function setting is disabled.

[含义] 由于 PPAR DR 功能被禁用，无法执行动态重新配置。

[措施] 检查 PPAR DR 功能设置，并再次执行此操作。

PPAR is currently unavailable for DR, because XBU status has failed.

[含义] 无法执行动态重新配置，由于在指定的物理分区或系统板 (PSB) 的交叉开关单元 (XBU) 中，或所配置的交叉开关盒的交叉开关单元中存在错误。

[措施] 检查监控信息和错误日志，以便确认相关交叉开关单元是否存在错误。如果存在错误，确定故障部件并进行更换。

PPAR (PPAR-ID XX) is currently unavailable for DR, because another operation is in progress.Try again later.

[含义] 由于其他操作正在访问指定的物理分区，无法执行动态重新配置。

[措施] 可以关闭其他会话的电源，或变更所连接 PCI 扩展单元配置信息的操作可以在其他会话中进行。检查物理分区的状态需要等待一段时间，然后再次执行此操作。

PPAR (PPAR-ID XX) is not currently running.

[含义] 由于目标物理分区未运行，虽然您指定了 -c configure 选项，但整合失败。

[措施] 检查物理分区的运行状态，或指定 -c assign 选项。

PSB#XX-X could not be configured into PPAR-ID XX due to operating system or Logical Domains Manager error.

[含义] 由于逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）返回的动态重新配置错误，整合失败。

[措施] 从输出的监控信息和控制台信息，确定 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。消除原因，并再次执行此操作。

PSB#XX-X encountered a hardware error.See error log for details.

[含义] 动态重新配置的硬件诊断检测到错误。硬件可能存在错误。

[措施] 检查监控信息和错误日志，以便确认硬件是否存在错误。如果存在错误，确定故障部件并进行更换。

PSB#XX-X has not been registered in PCL.

[含义] 由于指定的系统板在物理分区 (PPAR) 配置信息中未登录，无法执行此操作。

[措施] 登录物理分区 (PPAR) 配置信息，然后执行此操作。

PSB#XX-X is already assigned to another PPAR.

[含义] 由于指定的 PSB 被分配给其他物理分区，无法将其整合到指定的物理分区。

[措施] 将 PSB 分配给其他物理分区。检查 PSB 的状态。

PSB#XX-X is currently unavailable for DR.Try again later.

[含义] 由于其他操作正在访问指定的 PSB，无法执行动态重新配置。

[措施] 在其他会话可能正在执行动态重新配置、启动/停止物理分区，或维护操作。检查 PSB 的状态，稍等片刻，然后再次执行此操作。

PSB#XX-X is not installed.

[含义] 由于指定的 PSB 未安装，无法执行此操作。

[措施] 您可能已经误指定了 PSB。检查系统板的状态。

PSB#XX-X status is FAILED.

[含义] 指定的 PSB 处于故障状态。

[措施] 检查监控信息和错误日志，以便确认硬件是否存在错误。如果存在错误，确定故障部件并进行更换。

PSB#XX-X will be assigned to PPAR-ID XX.Continue?[y|n]:

[含义] PSB 编号 XX-X 将要被分配给物理分区 YY。您需要继续此操作吗？输入 "y" 继续，输入 "n" 取消。

PSB#XX-X will be configured into PPAR-ID XX.Continue?[y|n]:

[含义] PSB 编号 XX-X 将要被整合到物理分区 YY。您需要继续此操作吗？输入"y"继续，输入"n"取消。

Start configuring PSB to Logical Domains (LDoms) Manager.

[含义] 已经开始整合到 Logical Domains Manager（逻辑域管理器）。

Start connecting PSB to PPAR.

[含义] 已经开始整合到物理分区。

The current configuration does not support this operation.

[含义] 由于配置错误，命令终止。

[措施] 检查硬件安装状态和命令执行所需的功能设置。

The other XSCF is currently unavailable for DR.Try again later.

[含义] 由于备用或从 XSCF 处于关闭状态，无法执行动态重新配置。

[措施] 备用或从 XSCF 可能被关闭。再次执行动态重新配置，检查备用或从 XSCF 的状态。

The system is currently unavailable for DR, because another operation is in progress.Try again later.

[含义] 由于正在执行整个 SPARC M12/M10 系统的操作，无法执行动态重新配置。

[措施] 固件可能在其他会话正在更新。固件完全更新后，再次执行动态重新配置。

Timeout detected during communicate with Logical Domains Manager.

[含义] 与 Logical Domains Manager（逻辑域管理器）通讯时发生超时。

[措施] 从输出的监控信息和控制台信息，确定逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。然后，消除原因，并再次执行此操作。

Timeout detected during configuration into PPAR-ID XX.

[含义] 整合时间超时。

[措施] 从输出的监控信息和控制台信息，确定 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。消除原因，并再次执行此操作。

Timeout detected during self-test of PSB#XX-X.

[含义] 动态重新配置的硬件诊断时间超时。硬件可能存在错误。

[措施] 检查监控信息和错误日志，以便确认硬件是否存在错误。如果存在错误，确定故障部件并进行更换。

C.1.2 deleteboard

A hardware error occurred.Please check the error log for details.

[含义] 由于硬件错误，命令终止。

[措施] 检查可能的故障位置的错误日志，并更换故障组件。

An internal error has occurred.Please contact your system administrator.

[含义] 发生内部错误。联系系统管理员。

[措施] XSCF 中正在使用的库发生错误或 XSCF 内部资源不足。稍等片刻，然后再次执行此操作。如果再次出现错误，请联系富士通现场工程师进行错误调查。

Cannot communicate with Hypervisor or Logical Domains Manager.Please check the domain's state.

[含义] 与虚拟机监控程序或 Logical Domains Manager（逻辑域管理器）的通讯发生错误。

[措施] 从输出的监控信息和控制台信息，确定逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。然后，消除原因，并再次执行此操作。

Cannot communicate with XXX.Please check XXX's state.

[含义] 备用或从 XSCF 发生错误。

[措施] 解决备用或从 XSCF 的错误。在单机配置的 SPARC M12/M10 系统中不会发生此错误。

Control domain is stopped for CoD resource violation.

[含义] 由于 CPU 激活冲突导致控制域停止，指定的 PSB (BB) 无法释放。

[措施] 解决 CPU 激活冲突，然后再次执行此操作。

DR operation canceled by operator.

[含义] 由于用户发出指令将其停止，动态重新配置被取消。

Invalid parameter.

[含义] 参数错误

[措施] 检查参数和操作数。

Operation has completed.

[含义] 释放已经完成。

Operation is continued as "no".

[含义] 由于响应 [y|n] 提示时，未执行正确的操作，执行了选择 [n] 时应用的操作。

[措施] 检查系统板 (PSB) 信息。如果存在需要释放的 PSB，再次将其释放。

Operation not supported on this system.

[含义] 系统不支持所输入的命令。

[措施] 检查型号和 XCP 版本，查明是否可以执行此命令。

Permission denied.

[含义] 您没有访问权限。或者，您试图在备用 XSCF 上执行不可执行的命令，导致执行权限错误。

[措施] 比较您的用户权限和此命令所要求的执行权限。

PPAR DR function setting is disabled.

[含义] 由于 PPAR DR 功能被禁用，无法执行动态重新配置。

[措施] 检查 PPAR DR 功能设置，并再次执行此操作。

PPAR is currently unavailable for DR, because XBU status has failed.

[含义] 无法执行动态重新配置，由于在指定的物理分区或系统板 (PSB) 的交叉开关单元 (XBU) 中，或所配置的交叉开关盒的交叉开关单元中存在错误。

[措施] 检查监控信息和错误日志，以便确认相关交叉开关单元是否存在错误。如果存在错误，确定故障部件并进行更换。

PPAR (PPAR-ID XX) is currently unavailable for DR, because another operation is in progress.Try again later.

[含义] 由于其他操作正在访问指定的物理分区，无法执行动态重新配置。

[措施] 可以关闭其他会话的电源，或变更所连接 PCI 扩展单元配置信息的操作可以在其他会话中进行。检查物理分区的状态需要等待一段时间，然后再次执行此操作。

PPAR (PPAR-ID XX) is not currently running.

[含义] 由于目标物理分区未运行，虽然您指定了 -c unassign 或 -c disconnect 选项，但释放失败。

[措施] 检查物理分区的运行状态。

PSB power off sequence started.

[含义] PSB (BB) 的电源关闭已经启动。

PSB#XX-X can not be unconfigured because there is Hypervisor global memory.

[含义] 由于虚拟机监控程序的内存需要迁移，指定的 PSB 无法释放。

PSB#XX-X could not be unconfigured from PPAR-ID XX due to Hypervisor error.

[含义] 由于虚拟机监控程序返回动态重新配置错误，释放失败。

[措施] 从输出的监控信息和控制台信息，确定逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。然后，消除原因，并再次执行此操作。

PSB#XX-X could not be unconfigured from PPAR-ID XX due to operating system or Logical Domains Manager error.

[含义] 由于逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）返回的动态重新配置错误，释放失败。

[措施] 从输出的监控信息和控制台信息，确定逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。然后，消除原因，并再次执行此操作。

PSB#XX-X has not been registered in PCL.

[含义] 由于指定的 PSB 在物理分区 (PPAR) 配置信息中未登录，无法执行此操作。

[措施] 登录物理分区 (PPAR) 配置信息，然后执行此操作。

PSB#XX-X is currently unavailable for DR.Try again later.

[含义] 由于其他操作正在访问指定的 PSB，无法执行动态重新配置。

[措施] 在其他会话可能正在执行动态重新配置、启动/停止物理分区，或维护操作。检查 PSB 的状态，稍等片刻，然后再次执行此操作。

PSB#XX-X is not installed.

[含义] 由于指定的 PSB 未安装，无法执行此操作。

[措施] 您可能已经误指定了 PSB。检查 PSB 的状态。

**PSB#XX-X is the last LSB for PPAR-ID XX, and this PPAR is still running.
Operation failed.**

[含义] 由于指定的 PSB (BB) 是构成物理分区最后的 PSB (BB)，将无法从配置中移除。

[措施] 停止物理分区后，执行此操作，或通过指定 -c reserve 选项停止物理分区。

PSB#XX-X status is FAILED.

[含义] 由于指定的 PSB (BB) 的状态出现故障，将无法释放该系统板。

[措施] 检查监控信息和错误日志，以便确认硬件是否存在错误。如果存在错误，确定故障部件并进行更换。

PSB#XX-X will be unassigned from PPAR after the PPAR restarts.Continue?[y|n]:

[含义] 物理分区重新启动后，将预约 PSB 编号 XX-X 从物理分区的释放。您需要继续此操作吗？

PSB#XX-X will be unassigned from PPAR immediately.Continue?[y|n]:

[含义] PSB 编号 XX-X 将从物理分区中释放。您需要继续此操作吗？输入"y" 继续，输入"n" 取消。

PSB#XX-X will be unconfigured from PPAR immediately.Continue?[y|n]:

[含义] PSB 编号 XX-X 将从物理分区中释放，并将处于已分配的状态。您需要继续此操作吗？输入"y" 继续，输入"n" 取消。

Start unconfigure preparation of PSB.

[含义] PSB (BB) 释放的准备已经启动。

Start unconfiguring PSB from PPAR.

[含义] PSB (BB) 的释放处理已经开始。

System may be temporarily suspended, proceed?[y|n]:

[含义] 系统将被暂时挂起。您需要继续此操作吗？输入"y" 继续，输入"n" 取消。

The current configuration does not support this operation.

[含义] 由于配置错误，命令终止。

[措施] 检查硬件安装状态和命令执行所需的功能设置。

The other XSCF is currently unavailable for DR.Try again later.

[含义] 由于备用或从 XSCF 处于关闭状态，无法执行动态重新配置。

[措施] 备用或从 XSCF 可能被关闭。再次执行动态重新配置，检查备用或从 XSCF 的状态。

The system is currently unavailable for DR, because another operation is in progress.Try again later.

[含义] 由于正在执行整个 SPARC M12/M10 系统的操作，无法执行动态重新配置。

[措施] 固件可能在其他会话正在更新。固件完全更新后，再次执行动态重新配置。

There is no available action for the following resources.

或者，

DR operation would require shutting down domains, use force to proceed

[含义] 需要删除的系统板 (PSB) 的物理 I/O 设备可能被分配给逻辑域。

[措施] 从控制域执行 `ldm list-io` 命令来检查相关物理 I/O 设备的分配状态。如果将相关物理 I/O 设备分配给逻辑域，请使用 `ldm remove-io` 命令将其删除。

Timeout detected during communicate with Logical Domains Manager.

[含义] 与 Logical Domains Manager（逻辑域管理器）通讯时发生超时。

[措施] 从输出的监控信息和控制台信息，确定逻辑域的 Oracle Solaris 或 Logical Domains Manager（逻辑域管理器）的动态重新配置的故障原因。然后，消除原因，并再次执行此操作。

Timeout detected during poweroff.

[含义] 动态重新配置的系统板 (PSB) 关闭电源处理时间超时。

[措施] 确认硬件是否出现故障。如果存在错误，确定故障部件并进行更换。

Timeout detected during unconfiguration of PSB#XX-X.

[含义] 动态重新配置的释放超时。

[措施] 检查监控信息和错误日志，以便确认硬件是否存在错误。如果存在错误，确定故障部件并进行更换。

Unconfigure preparation of PSB has completed.

[含义] PSB (BB) 释放的准备已经完成。

Unconfigured PSB from PPAR.

[含义] PSB (BB) 释放已经完成。

索引

A

addboard, 417

C

CPU, 19

CPU 插槽, 各资源使用状况检查, 99

CPU 插槽限制条件, 设置, 112

CPU 激活密钥, 添加, 83

CPU 激活密钥信息, 检查, 76

CPU 激活注册, 78

CPU 内核资源的使用状况, 检查, 76

CPU 运行模式, 23, 65

D

deleteboard, 420

动态重新配置, 411

动态重新配置的设置, 48

动态重新配置的条件, 48

F

服务, 检查, 95

G

关闭组, 指定, 124

H

恢复模式, 139

恢复模式, 启用, 124

恢复模式, 配置, 139

I

I/O 设备, 配置, 113

I/O 设备的使用状况, 检查, 101

J

检查 CPU 激活设置信息, 78

交换空间, 34

K

可分配的虚拟 CPU, 检查数量, 96

控制域控制台, 93

控制域, 初始设置, 137

控制域, 登录, 135

L

来宾域概述, 175

来宾域, 启动, 123

来宾域, 迁移, 175, 178

来宾域迁移示例, 178

来宾域, 配置, 138

逻辑域, 15

逻辑域重新配置, 46

逻辑域, 创建, 113

逻辑域的设置, 35

逻辑域的条件, 35

逻辑域管理器正在运行, 确认, 95

逻辑域配置, 36, 93, 201, 269

逻辑域配置, 操作, 16

逻辑域配置操作示例, 134

- 逻辑域配置流程, 133
- 逻辑域配置示例, 133
- 逻辑域配置信息, 92
- 逻辑域配置信息, 设置, 112
- 逻辑域配置信息, 显示, 101
- 逻辑域配置信息, 保存, 140
- 逻辑域状态, 检查, 74, 100
- 逻辑域组件, 15

M

- 命令信息, 417
- 默认服务, 设置, 104, 136

N

- 内存镜像模式, 33
- 内存镜像, 设置, 81

O

- Oracle Solaris 的设置, 34
- Oracle Solaris 的条件, 34

P

- PPAR DR 模式, 58

Q

- 启动设备, 配置, 123

R

- 软件条件, 20

S

- 设备, 重新配置, 124
- 设备设置信息, 检查, 73
- 实时处理, 35
- SPARC64 X 处理器, 22, 64
- SPARC64 X+ 处理器, 22, 60, 64
- SR-IOV 虚拟功能, 创建或销毁, 116

W

- 物理分区, 11
- 物理分区编号, 变更注意事项, 173
- 物理分区重新配置, 141
- 物理分区重新配置, 流程, 141

- 物理分区动态重新配置, 18, 65, 182
- 物理分区动态重新配置, 环境配置示例和步骤使用, 181
- 物理分区配置, 67
- 物理分区配置, 操作, 12
- 物理分区配置操作示例, 126
- 物理分区配置流程, 125
- 物理分区配置信息, 检查, 68
- 物理分区配置信息, 设置, 82
- 物理分区, 启动, 92
- 物理分区, 停止, 92
- 物理分区运行模式, 检查, 79
- 物理分区运行模式, 设置, 90
- 物理分区状态, 检查, 70
- 物理分区组件, 11
- 物理分区, 动态重新配置, 概述, 18
- 物理分区, 动态重新配置, 使用, 19
- 物理分区状态, 25
- 物理 I/O 设备, 18

X

- 系统运行, 设置, 25
- 系统运行, 条件, 25
- XSCF 的设置, 31
- XSCF 的条件, 31
- 虚拟磁盘服务器, 配置, 120
- 虚拟磁盘, 配置, 121
- 虚拟 CPU, 配置, 105
- 虚拟控制台, 配置, 122
- 虚拟内存, 配置, 109
- 虚拟网络设备, 配置, 119

Y

- 延迟重新配置模式, 启动, 103
- 硬件资源, 8
- 域配置, 1, 67
- 域配置, 概括, 1

Z

- 状态管理, XSCF, 25
- 资源, 绑定, 123
- 资源的分配状况, 检查, 96
- 资源的使用状况, 检查, 96

资源组，各使用状况检查， 98

