

**Oracle Solaris コンテナを使ってみよう**  
**（構築・運用手順書）**

**2011年10月**

**第2版**

**富士通株式会社**



## 本資料について

- Solaisコンテナに関する構築・運用手順を紹介しています。導入・構築ガイド「Solarisコンテナを使ってみよう」を参照しながら、本手順書を活用してください。  
参考URL <http://primeserver.fujitsu.com/sparcenterprise/technical/>
- 本資料では、Solaris OS 環境をOracle VM Server for SPARC上に構築した場合の手順を記載しております。そのため一部、Oracle VM Server for SPARCに依存した内容があります。例えば、ディスクやネットワークインターフェイスなどがOracle VM Server for SPARC固有の値となっており、物理サーバ環境とは異なりますのでご了承ください。

## 使用条件

- 著作権・商標権・その他の知的財産権について  
コンテンツ(文書・画像・音声等)は、著作権・商標権・その他の知的財産権で保護されています。本コンテンツは、個人的に使用する範囲でプリントアウトまたはダウンロードできます。ただし、これ以外の利用(ご自分のページへの再利用や他のサーバへのアップロード等)については、当社または権利者の許諾が必要となります。
- 保証の制限  
本コンテンツについて、当社は、その正確性、商品性、ご利用目的への適合性等に関して保証するものではなく、そのご利用により生じた損害について、当社は法律上のいかなる責任も負いかねます。本コンテンツは、予告なく変更・廃止されることがあります。

## 商標

- UNIXは、米国およびその他の国におけるオープン・グループの登録商標です。
- SPARC Enterprise、SPARC64およびすべてのSPARC商標は、米国SPARC International, Inc.のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- OracleとJavaは、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

## ■ Oracle Solaris コンテナを使ってみよう 項得・運用手順書 目次

## 1章 基礎編

## 1-1. Resource Pool の設定

- (1) デフォルトスケジューラ の設定
- (2) Resource Pool サービスの起動
- (3) Resource Pool の構成
- (4) Resource Pool 構成の反映
- 【参考】 一括定義方法
- 【参考】 スケジューラ の初期化
- 【参考】 その他の pooladm コマンド操作
- 【参考】 その他の poolcfg コマンド操作例

## 1-2. zone の作成

- (1) zone の状態確認方法
- (2) マウント先の確認
- (3) 新規 zone の構成
- (4) 構成済み zone のインストール
- (5) インストール済み zone の起動
- (6) 初回起動時の zone のセットアップ
- 【参考】 その他の zonecfg 設定例

## 1-3. Solaris コンテナの基本操作

- (1) 起動・停止・ログイン
- (2) zone におけるプロセスの見え方

## 2章 応用編

## 2-1. Solaris コンテナの検証環境利用

- (1) Solaris コンテナクローンの準備
- (2) Solaris コンテナクローン作成

## 2-2. Solaris コンテナの災対環境利用

- (1) Solaris コンテナの移動
- (2) 災対環境での Solaris コンテナ起動

## 2-3. Solaris コンテナ統合環境におけるリソース操作

- (1) リソース操作の事前準備と確認
- (2) シェア数の動的変更
- (3) cpu-cap の設定
- (4) CPU リソースの動的移動

## 2-4. Solaris コンテナのバッチ適用

- (1) サーバへのコンソール接続
- (2) バッチ適用前の準備
- (3) 並列バッチ適用の準備
- (4) コンテナ環境の並列バッチ適用
- (5) 適用したバッチの削除
- (6) バッチ適用前の準備 (従来の方法)
- (7) 従来の方法でのバッチ適用

## ■ 1章 基礎編

## 1-1. リソースプールの設定

## (1) デフォルトスケジューラ の設定

## 1) スケジューラの確認

```
global01# ps -efc | grep /sbin/init
```

```
global01# ps -efc | grep /sbin/init
root    1    0  TS  59 10:02:07 ?        0:00 /sbin/init
```

※ デフォルトで動作しているinitプロセスのスケジューラが、TS(タイムシェアリング)であることを確認します。

```
global01# dispadmin -d
```

```
global01# dispadmin -d
dispadmin: Default scheduling class is not set
```

※ OS起動時のスケジューラ設定の確認コマンド。未設定時は確認できません。

## 2) スケジューラの変更

```
global01# priocntl -s -c FSS -i pid 1
```

※ initプロセスのスケジューラをFSSに変更します。

```
global01# priocntl -s -c FSS -i class TS
```

※ TSクラスで動作しているプロセスのスケジューラをFSSに変更します。

```
global01# dispadmin -d "FSS"
```

※ OSのデフォルトのスケジューラをFSS(フェアシェアスケジューラ)に変更します。

## 3) スケジューラの確認

```
global01# ps -efc | grep /sbin/init
```

```
global01# ps -efc | grep /sbin/init
root    1    0  FSS 29 10:02:07 ?        0:00 /sbin/init
```

※ FSSに変更されたことを確認します。

```
global01# dispadmin -d
```

```
# dispadmin -d
FSS (Fair Share)
```

※ FSSに変更されたことを確認します。

## (2) リソースプール サービスの起動

## 1) リソースプールの状態確認

```
global01# pooladm
```

※ リソースプールの初期状態を確認します。

```
global01# pooladm
pooladm: プール状態ファイルをオープンできませんでした
:機能がアクティブではありません
```

## 2) リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスの状態を確認します。

```
global01# svcs -a | grep pools
disabled 10:02:14 svc:/system/pools:default
disabled 10:02:20 svc:/system/pools/dynamic:default
```

## 3) リソースプールの起動

```
global01# svcadm enable svc:/system/pools:default
```

※ リソースプールサービスを起動します。

## 4) リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスが起動していることを確認します。

```
global01# svcs -a | grep pools
disabled 10:02:20 svc:/system/pools/dynamic:default
online 10:07:13 svc:/system/pools:default
```

## 5) リソースプールの状態確認

```
global01# pooladm
```

```
global01# pooladm
--<省略>--
pool pool_default
  int pool.sys_id 0
  boolean pool.active true
  boolean pool.default true
  int pool.importance 1
  string pool.comment
  pset pset_default

pset pset_default
  int pset.sys_id -1
  boolean pset.default true
  uint pset.min 1
  uint pset.max 65536
  string pset.units population
  uint pset.load 3
```

pool\_defaultと  
pset\_defaultの  
み存在すること  
を確認します。

## (3) リソースプールの構成

## 1) リソースプール 構成の保存 (メモリ上で動作するリソースプールの構成をファイルに書き出します。)

```
global01# pooladm -s
```

※ 最後にファイル名を指定しない場合は、現在のリソースプールの状態を /etc/pooladm.conf に書き出します。  
また、既にファイルがある場合は上書きします。ファイル名を指定することで別ファイルに保存することもできます。

## 2) リソースプール 構成ファイル内容の確認

```
global01# poolcfg -c info
```

※ pool\_default と pset\_default のみの構成であることを確認します。

## 3) リソースプール 構成ファイルの編集 (/etc/pooladm.conf への書き込み)

## 1. デフォルトのプロセッサセットの最低CPU数を変更

```
global01# poolcfg -c 'modify pset pset_default (uint pset.min=8)'
```

## 2. 新規プロセッサセット (pset) の作成

```
global01# poolcfg -c 'create pset pset_1 (uint pset.min=8; uint pset.max=8)'
```

## 3. 新規プール (pool) の作成

```
global01# poolcfg -c 'create pool pool_1'
```

## 4. プロセッサセットとプールの関連付け

```
global01# poolcfg -c 'associate pool pool_1 (pset pset_1)'
```

## 5. プールのスケジューラ設定

```
global01# poolcfg -c 'modify pool pool_1 (string pool.scheduler="FSS")'
```

## (4) リソースプール 構成の反映

## 1. 構成ファイル内容の確認

```
global01# poolcfg -c info
```

※ 作成した pool\_1 と pset\_1 が追加されていることを確認します。

## 2. 構成の実行

```
global01# pooladm -c
```

※ メモリ上でリソースプール構成が実行されます。実行時にエラーが出る場合は、構成ファイルを見直してください。

## 3. 実行中のリソースプール構成の確認

```
global01# poolstat -r all
```

```
global01# poolstat -r all
id pool          type rid rset          min max size used load
1 pool_1        pset  1 pset_1          8  8  8  0.00 0.00
0 pool_default  pset -1 pset_default    8 66K 8  0.00 0.01
```

## 【参考】一括定義方法

リソースプール 構成ファイルの編集は一括定義も可能です。

## 1. 任意のファイルに定義情報を記述

```
global01# vi /tmp/poolcfg.txt
```

(記載例)

```
create system default
modify pset pset_default (uint pset.min=8)
create pset pset_1 (uint pset.min=8; uint pset.max=8)
create pool pool_1
associate pool pool_1(pset pset_1)
modify pool pool_1(string pool.scheduler="FSS")
```

## 2. ファイルの読み込み

```
global01# poolcfg -f /tmp/poolcfg.txt
```

## 【参考】スケジューラの初期化

## 1) デフォルトスケジューラを TS に設定する方法

```
global01# dispadmin -d TS
global01# init 6
```

## 2) スケジューラの設定ファイルを削除する方法

```
global01# rm /etc/dispadmin.conf
global01# init 6
```

## 【参考】その他の pooladm コマンド操作

注意事項: Solaris コンテナ稼働中には行わないで下さい。

## 1) 現在動作しているリソースプール 状態を破棄 (初期状態の設定に戻る)

```
global01# pooladm -x
```

## 2) 現在動作しているリソースプール を停止 (リソースプール サービスが停止される)

```
global01# svcadm disable svc:/system/pools:default
```

または、

```
global01# pooladm -d
```

※ Solaris 10 6/06 以前のバージョンの場合は、こちらのコマンドを使用する必要があります。

## 【参考】その他の poolcfg コマンド操作例

## 1) 設定の削除

## 1. 例: プール pool\_1 を削除

```
global01# poolcfg -c 'destroy pool pool_1'
```

## 2. 例: プロセッサセット pset\_1 を削除

```
global01# poolcfg -c 'destroy pset pset_1'
```

## 2) 設定した名前の変更

## 1. 例: プール pool\_1 を pool\_2 に変更

```
global01# poolcfg -c 'rename pool pool_1 to pool_2'
```

## 2. 例: プロセッサセット pset\_1 を pset\_2 に変更

```
global01# poolcfg -c 'rename pset pset_1 to pset_2'
```

## 3) 設定したプロパティの変更

## 1. 例: プール pool\_1 のコメントを変更

```
global01# poolcfg -c 'modify pool pool_1 (string pool.comment="TEST_POOL")'
```

## 2. 例: プロセッサセット pset\_1 の最大CPU数を 1 に変更

```
global01# poolcfg -c 'modify pset pset_1 (uint pset.max=1)'
```

## 4) 設定したプロパティの削除

## 1. 例: プール pool\_1 のスケジューラ設定を削除

```
global01# poolcfg -c 'modify pool pool_1 (~ string pool.scheduler)'
```

## 1-2. Zone の作成

## (1) zone の状態確認方法

```
global01# zoneadm list -vc
```

※ zone の状態一覧を表示します。(初期状態では、global zone のみが表示されます。)

```
global01# zoneadm list -vc
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared

## (2) マウントポイントの確認

```
global01# df -h
```

※ zone 用のディスクがマウント済みになっています。

```
global01# df -h
```

ファイルシステム	サイズ	使用済み	使用可能	容量	マウント先
/dev/dsk/c0d0s0	7.9G	4.9G	2.9G	64%	/
-<省略>-					
/dev/dsk/c0d1s0	7.8G	7.9M	7.7G	1%	/zone_tar_list
/dev/dsk/c0d3s0	5.9G	6.0M	5.8G	1%	
/export/zones_clone					
/dev/dsk/c0d2s0	5.9G	6.0M	5.8G	1%	/export/zones
/dev/dsk/c0d4s0	5.9G	4.0G	1.8G	69%	

## (3) 新規 zone の構成

## 1) zone 構成の開始

```
global01# zonecfg -z zone01
```

※ 実行後プロンプトが変わり、構成コマンドが実行可能となります。

## 2) zone の作成

```
zonecfg:zone01> create
```

※ 初期設定値を読み込みます。zone 作成時に必ず最初に行う必要があります。

継承ディレクトリを設定を行わない場合  
4GB以上の領域が必要です。  
4GB以上の領域があることを確認します。

## 3) zone 設定の確認

```
zonecfg:zone01> info
```

※ zone の設定状態を確認します。export コマンドでも確認できます。

## 4) zone バスの設定

```
zonecfg:zone01> set zonepath=/export/zones/zone01
```

※ zone 環境のルートディレクトリを、global zone 上の絶対パスで指定します。

ディスク構成を zone 専用にするためマウントポイント

## 5) zone の自動起動設定

```
zonecfg:zone01> set autoboot=false
```

※ true あるいは false を設定します。true を設定した場合は OS (global zone) 起動時に同時起動されます。

ここでは、OS 起動時に zone の自動起動  
をしない設定にします。

## 6) Resource Pool の関連付け設定

```
zonecfg:zone01> set pool=pool_1
```

※ zone で使用する Resource Pool を指定します。

## 7) スケジューラの設定

```
zonecfg:zone01> set scheduling-class="FSS"
```

※ zone で使用する Resource Pool のスケジューラを指定します。

## 8) インターフェースの共有設定

```
zonecfg:zone01> set ip-type=shared
```

## 9) ネットワークインタフェースの設定

```
zonecfg:zone01> add net
zonecfg:zone01:net> set address=XXX.XXX.XXX.XXX/XX
zonecfg:zone01:net> set physical=vnet0
zonecfg:zone01:net> end
zonecfg:zone01>
```

※ 他の zone と IP アドレスが競合しないように設定してください。  
競合した場合は、同時起動は出来ません。また、パッチ適用やパッケージインストールも失敗します。

## 10) 継承ディレクトリ設定の削除【注意】

```
zonecfg:zone01> remove inherit-pkg-dir dir=/lib
zonecfg:zone01> remove inherit-pkg-dir dir=/platform
zonecfg:zone01> remove inherit-pkg-dir dir=/sbin
zonecfg:zone01> remove inherit-pkg-dir dir=/usr
zonecfg:zone01>
```

※ zone インストール後は設定変更ができません。

non-global zone にミドルウェアなどをインストールする場合  
デフォルトのディレクトリの継承設定の削除を行う必要  
があります。



## 11) 共有ファイルシステムの設定

```
zonecfg:zone01> add fs
zonecfg:zone01:fs> set dir=/cdrom
zonecfg:zone01:fs> set special=/cdrom
zonecfg:zone01:fs> set type=lofs
zonecfg:zone01:fs> end
zonecfg:zone01>
```

share ディレクトリとなる「special=」パラメータは global zone に存在するディレクトリを指定します。  
mount ポイントとなる「dir=」パラメータは non-global zone にマウントさせるディレクトリを指定します。

※ global zone の cdrom を共有します。

## 12) リソースパラメータ(CPUシェア数)の設定

```
zonecfg:zone01> set cpu-shares=1
```

## 13) zone の設定確認

```
zonecfg:zone01> info
```

## 14) zone の設定の検証

```
zonecfg:zone01> verify
```

## 15) zone の設定の確定

```
zonecfg:zone01> commit
```

※ 設定内容を確定します。(省略可能)

## 参考) デバイスの設定

```
zonecfg:zone01> add device
zonecfg:zone01:fs> set match=/dev/dsk/cXdXsX
zonecfg:zone01:fs> end
zonecfg:zone01> add device
zonecfg:zone01:fs> set match=/dev/rdisk/cXdXsX
zonecfg:zone01:fs> end
zonecfg:zone01>
```

## 16) zone の設定の終了

```
zonecfg:zone01> exit
```

※ 設定内容を終了して、通常のプロンプトに戻ります。

## 便利な機能

zonecfg コマンドプロンプトでは、Tabキーを押すと、入力候補一覧が表示されます。  
また、入力候補がある場合は、途中まで入力したコマンドの補完もされます。

## (4) 構成済み zone のインストール

## 1) zone 設定のインストール前の確認

```
global01# zonecfg -z zone01 'info'
```

## 2) zone の状態を表示

```
global01# zoneadm list -vc
```

※ zone の状態一覧を表示し、構成した zone が表示されることを確認します。

```
global01# zoneadm list -vc
ID NAME STATUS PATH BRAND IP
0 global running / native shared
- zone01 configured /export/zones/zone01 native shared
```

## 3) zone のインストールの実行

```
global01# zoneadm -z zone01 install
```

※ zone のインストールを行います。数10分時間がかかります。

## 4) zone のインストール中の状態確認(他のコンソールから実行)

```
global01# zoneadm list -vc
```

※ インストール中の zone の状態(incomplete)が確認できます。

```
global01# zoneadm list -vc
ID NAME STATUS PATH BRAND IP
0 global running / native shared
- zone01 incomplete /export/zones/zone01 native shared
```

## 5) zone のインストール完了後の状態確認

```
global01# zoneadm list -vc
```

※ インストールが完了した zone の状態(installed)を確認します。

```
global01# zoneadm list -vc
ID NAME STATUS PATH BRAND IP
0 global running / native shared
- zone01 installed /export/zones/zone01 native shared
```

## (5) インストール済み zone の起動

## 1) インストールした zone の起動

```
global01# zoneadm -z zone01 boot
```

※ zone の起動方法になります。

zone の起動・停止操作については次の章で詳しく実施します。

## 2) zone の状態を表示

```
global01# zoneadm list -vc
```

※ 起動した zone の状態(running)を確認します。

```
global01# zoneadm list -vc
ID NAME      STATUS  PATH                                BRAND IP
0 global    running /                                     native shared
1 zone01    running /export/zones/zone01 native shared
```

## (6) 初回起動時の zone のセットアップ【必須】

## 1) zone のコンソールログイン

```
global01# zlogin -C zone01
```

※ zone をインストールしたら、zone の運用を始める前に、必ずコンソールログインして初期設定を行う必要があります。  
 ※ 初期設定を実施しなかった場合、必要なサービスが正常に起動しません。  
 ※ zone が起動している状態で実行してください。

## 2) zone の初回セットアップ

## 設定値 (例)

Language	1.Japanese
Local	0.Japanese EUC (ja)
端末タイプ	3)DEC VT100
Host Name	(任意に設定)
Kerberos Security	No
Name Service	None
NFSv4 Domain name	Use the NFSv4 domain derived by the system
Time Zone	Asia
Country or Region	Japan
Root Password	(任意に設定)

## 操作方法

Host Name以降の設定は、[矢印キー]でカーソル移動、[space]で選択し、[Esc] [2]で次の画面へ移動します。

## 3) コンソール接続を閉じる

```
zone01 console login:
zone01 console login: ~
[ゾーン 'zone1' コンソールへの接続が閉じられました]
global01#
```

「~」(チルダ)を入力し「.」(ドット)を入力します。

## &lt;注意&gt;

「~」を入力した直後は画面に表示されません。  
 「.」を入力した後に同時に「~」が表示されます。

**【参考】その他の zonecfg 設定例**

- 1) リソースプールの関連付け設定変更 (新しい設定が上書きされます。)

```
zonecfg:zone01> set pool=pool_default
```

- 2) IPアドレスの変更

```
zonecfg:zone01> select net address=XXX.XXX.XXX.XXX/XX
zonecfg:zone01:net> set address=XXX.XXX.XXX.XXX/XX
zonecfg:zone01:net> end
zonecfg:zone01>
```

- 3) 設定の削除(例:共有ファイルシステムの設定の削除)

```
zonecfg:zone01> remove fs dir=/mnt
```

- 4) 設定の中止(例:共有ファイルシステムの設定の中止)

```
zonecfg:zone01> add fs
zonecfg:zone01:fs> set dir=/mnt
zonecfg:zone01:fs> cancel
zonecfg:zone01>
```

## 1-3. Solaris コンテナの基本操作

## (1) 起動・停止・ログイン

## 1) zone の状態を表示

```
global01# zoneadm list -vc
```

## 2) zone の起動・再起動・停止操作 (緊急時)

## 1. zone の再起動 (緊急時再起動用)

```
global01# zoneadm -z zone01 reboot
```

## 2. zone の停止 (緊急時停止用)

```
global01# zoneadm -z zone01 halt
```

## 3. zone の待機

```
global01# zoneadm -z zone01 ready
```

## 4. zone の起動 (通常起動)

```
global01# zoneadm -z zone01 boot
```

通常運用時の停止や再起動と比較してみましょう。

※zone は起動する毎に zone IDが更新されます。

## 3) インタラクティブモード(対話型)でのログイン

```
global01# zlogin zone01
```

※ インタラクティブモードは「exit」もしくは「Ctrl-C」でコンテナからログアウトできます。

## 4) ノンインタラクティブモード(非対話型)での zone 操作

```
global01# zlogin zone01 'cat /etc/hosts'
```

※ zone で実行したコマンドの結果のみを表示します。(対話型コマンドは実行できません。)

※ オプション -l [ユーザ名] で実行ユーザを指定することが可能です。

【注意】 実行しないでください。

zone を incomplete 状態にするコマンド

```
global01# zoneadm -z zone01 mark incomplete
```

zone は使用できない状態になります。

## 5) zone の起動・再起動・停止操作 (通常運用時)

## 1. zone の再起動

```
global01# zlogin zone01 'shutdown -y -g0 -i6'
```

## 2. zone の停止

```
global01# zlogin zone01 'shutdown -y -g0 -i5'
```

## 3. zone のシングルユーザモード起動

```
global01# zoneadm -z zone01 boot -s
```

## 4. zone の再起動

```
global01# zlogin zone01
[ゾーン 'zone01' pts/2 に接続されました]
Last login: Wed Oct 31 15:16:39 on pts/2
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
#
# shutdown -y -g0 -i6
#
[ゾーン 'zone01' pts/2 への接続が閉じられました]
global01#
```

## 5. zone の停止

```
global01# zlogin zone01
[ゾーン 'zone01' pts/2 に接続されました]
Last login: Wed Oct 31 15:16:39 on pts/2
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
#
# shutdown -y -g0 -i5
#
[ゾーン 'zone01' pts/2 への接続が閉じられました]
global#
```

## 6. zone の起動

```
global01# zoneadm -z zone01 boot
```

緊急時の停止や再起動と比較してみましょう。

※zone は起動する毎に zone IDが更新されます。

## (2) zone におけるプロセスの見え方

## 1) global zone

## 1. 通常のプロセス確認

```
global01# ps -ef  
global01# ps -ef | wc -l
```

## 2. プロセスが動作している zone 名を表示して確認

```
global01# ps -efZ  
global01# ps -efZ | grep zone01
```

## 3. 動作しているプロセスのスケジューリングクラスを表示して確認

```
global01# ps -efc  
global01# ps -efc | grep /sbin/init
```

global zone と non-global zone の  
プロセスが表示されます。

## 2) non-global zone

## 1. インタラクティブモードでログイン

```
global01# zlogin zone01
```

## 2. 通常のプロセス確認

```
# ps -ef  
# ps -ef | wc -l
```

## 3. プロセスが動作している zone 名を表示して確認

```
# ps -efZ  
# ps -efZ | grep zone01
```

## 4. 動作しているプロセスのスケジューリングクラスを表示して確認

```
# ps -efc  
# ps -efc | grep /sbin/init
```

## 5. インタラクティブモードの接続を閉じる

```
# exit
```

```
[ゾーン 'zone01' pts/2 への接続が閉じられました]  
global01#
```

表示結果を比較して  
みましょう。

non-global zone のプロセスのみ  
が表示されます。

## ■ 2章 応用編

## 2-1. Solaris コンテナの検証環境利用

(1) Solaris コンテナクローンの準備

## 1) 構成ファイルの作成と編集

## 1. zone の停止

```
global01# zlogin zone01 'shutdown -y -g0 -i5'
```

## 2. zone の状態を表示

```
global01# zoneadm list -vc
```

※ zone の状態一覧を表示して状態を確認します。

クローン元の zone が完全に停止していることを確認してください。

## 3. zone の構成ファイルを複製

```
global01# zonecfg -z zone01 export -f /export/zones/zone01_master_file
```

複製した zone の構成ファイル名は任意のファイル名を指定することができます。

[参考] 編集前の /export/zones/zone01\_master\_file

```
global01# cat /export/zones/zone01_master_file
"/export/zones/zone01_master_file" 33 行、546 バイト
create -b
set zonepath=/export/zones/zone01
set autoboot=false
set pool=pool_1
set scheduling-class=FSS
set ip-type=shared
add fs
set dir=/cdrom
set special=/cdrom
set type=lofs
end
add net
set address=XXX.XXX.XXX.XXX/XX
set physical=vnet0
end
-<省略>-
```

## 4. 複製した構成ファイルの編集

```
global01# vi /export/zones/zone01_master_file
```

zone名や、IPアドレスなどは、設定一覧を参照し、読み替えて設定してください。

[参考] 編集後の /export/zones/zone01\_master\_file

```
global01# vi /export/zones/zone01_master_file
"/export/zones/zone01_master_file" 33 行、546 バイト
create -b
set zonepath=/export/zones_clone/zone02
set autoboot=false
set pool=pool_1
set scheduling-class=FSS
set ip-type=shared
add fs
set dir=/mnt
set special=/mnt
set type=lofs
end
add net
set address=XXX.XXX.XXX.XXX/XX
set physical=vnet0
end
-<省略>-
```

zonepath=/export/zones\_clone/zone02 へ変更

global zone との共有ファイルシステムを /mnt へ変更  
non-global zone のマウントポイントを /mnt へ変更  
※任意の値に設定してください

IP addressを変更

※ zonepath と IP アドレス、デバイス情報などは複製元の構成と重複するため変更する必要があります。

## 2) 編集した複製ファイルを使って新規 zone の構成

```
global01# zonecfg -z zone02 -f /export/zones/zone01_master_file
```

## 3) zone の状態確認

```
global01# zoneadm list -vc
```

```
global01# zoneadm list -vc
ID NAME      STATUS      PATH                BRAND IP
0 global     running    /                   native shared
- zone01     installed  /export/zones/zone01 native shared
- zone02     configured /export/zones_clone/zone02 native shared
```

## (2) Solaris コンテナクローン作成

## 1. 複製 zone02 の作成

```
global01# zoneadm -z zone02 clone zone01
/export/zones/zone1 をコピーしています...
grep: can't open /a/etc/dumpadm.conf
※ 約30分程かかります。
```

「grep: can't open /a/etc/dumpadm.conf」メッセージは問題はありませんので無視してください。

## 2. zone の状態確認

```
global01# zoneadm list -vc
```

```
global01# zoneadm list -vc
ID NAME      STATUS      PATH                BRAND IP
0 global     running    /                   native shared
- zone01     installed  /export/zones/zone01 native shared
- zone02     installed  /export/zones_clone/zone02 native shared
```

## 3. zone02 の起動

```
global01# zoneadm -z zone02 boot
```

## 4. zone のコンソールログイン(初期設定)

```
global01# zlogin -C zone02
```

- ※ 初期設定を実施しなかった場合、正常にコンテナが起動しないので必ず実行する必要があります。
- ※ zone が起動している状態で実行してください。

## 5. zone の初回セットアップ

## 設定値 (例)

Language	1.Japanese
Local	0.Japanese EUC (ja)
端末タイプ	3)DEC VT100
Host Name	(任意に設定)
Kerberos Security	No
Name Service	None
NFSv4 Domain name	Use the NFSv4 domain derived by the system
Time Zone	Asia
Country or Region	Japan
Root Password	(任意に設定)

## 操作方法

Host Name以降の設定は、[矢印キー]でカーソル移動、[space]で選択し、[Esc] [2]で次の画面へ移動します。

## 6. サービスの確認

```
# svcs -a
```

サービスの状態である「STATE」の部分に "legacy\_run"、"disabled"、"online"、"offline" 以外の表示がないことを確認します。

## 7. コンソールを閉じる

```
#~
[ゾーン 'zone1' コンソールへの接続が閉じられました]
global01#
```

「~」(チルダ)を入力し「.」(ドット)を入力します。

## &lt;注意&gt;

- 「~」を入力した直後は画面に表示されません。
- 「.」を入力した後に同時に「~」が表示されます。

## 2-2. Solaris コンテナの災対環境利用

## (1) Solaris コンテナの移動

## 1) zone の確認と切り離し

## 1. zone の状態確認

```
global01# zoneadm list -vc
```

※ 移動させるコンテナ(zone01)が完全に停止している(installed)ことを確認してください。

```
# zoneadm list -vc
ID NAME STATUS PATH BRAND IP
0 global running / native shared
- zone01 installed /export/zones/zone01 native shared
10 zone02 running /export/zones_clone/zone02 native
```

## 2. zone の切り離し

```
global01# zoneadm -z zone01 detach
```

※ zone を切り離して移動できる状態にします。

detach すると、zone の状態が installed から configured に変わります。

## 3. ディスク容量の確認

```
global01# df -h
```

※ zone のアーカイブの配置先と容量の確認をします。移動させる zone の容量に依存します。

```
global01# df -h
ファイルシステム サイズ 使用済み 使用可能 容量 マウント先
/dev/dsk/c0d0s0 7.9G 4.9G 2.9G 64% /
- <省略> -
/dev/dsk/c0d1s0 7.8G 7.9M 7.7G 1% /zone_tar_list
/dev/dsk/c0d4s0 5.9G 4.0G 1.8G 69% /export/zones_patch
/dev/dsk/c0d2s0 5.9G 4.0G 1.8G 69% /export/zones
/dev/dsk/c0d3s0 5.9G 4.0G 1.8G 69%
```

## 4. zone のアーカイブデータ作成

```
global01# cd /export/zones
```

```
global01# tar -cfe /zone_tar_list/zone01.tar ./zone01
```

## 5. zone の組み込み

```
global01# zoneadm -z zone01 attach
```

※ zone を組み込んで使用可能状態にします。

作成するアーカイブのサイズは4GBになります。  
4GB以上の領域があることを確認します。

## &lt;ご参考&gt;

zone の切り離し (detach) を実行すると、zonpath 上に zone の構成情報ファイル SUNWattached.xml が生成されます。zone の構成情報やパッケージ/パッチ等の情報が記載されており zone の組み込み (attach) 時に使用されます。

## 2) アーカイブ転送

```
global01# ftp XXX.XXX.XXX.XXX ←global02へftpの開始
Connected to XXX.XXX.XXX.XXX.
220 global02 FTP server ready.
Name (XXX.XXX.XXX.XXX:root): XXXXXX ←ログインするユーザ名
331 Password required for XXXXXX
Password: XXXXXX ←ログインユーザのパスワード
230 User root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bin ←バイナリ転送モードの指定
200 Type set to I.
ftp> cd /zone_tar_list ←リモートカレントディレクトリの移動
250 CWD command successful.
ftp> lcd /zone_tar_list ←ローカルのカレントディレクトリの移動
Local directory now /zone_tar_list
ftp> !pwd ←ローカルのカレントディレクトリの確認
/zone_tar_list
ftp> put zone01.tar ←ファイルのアップロード
200 PORT command successful.
150 Opening BINARY mode data connection for zone01.tar.
226 Transfer complete.
local: zone01.tar remote: zone01.tar
10517409280 bytes sent in 4.5e+02 seconds (22591.41 Kbytes/s)
ftp> bye ←ftpの終了
```



## (2) 災対環境での Solaris コンテナ起動

## 1) アーカイブの展開と組み込み

ここからは、もう1台のサーバでの作業となります。

## 1. アーカイブの展開

```
global02# cd /export/zones_disaster
global02# tar -xf /zone_tar_list/zone01.tar
```

アーカイブの展開には、時間がかかります。

※ zone のデータを移行先ホストで展開します。

## 2. 新規 zone の構成

```
global02# zonecfg -z zone01 create -a /export/zones_disaster/zone01
```

create 直後の zone の状態は configured になります。

## 3. zone の状態を確認

```
global02# zoneadm list -vc
```

## 4. 移行した zone の組み込み

```
global02# zoneadm -z zone01 attach
```

attach することにより、zone の状態が configured から installed に変わります。

## 5. zone の状態を確認

```
global02# zoneadm list -vc
```

## 6. zone01 の起動

```
global02# zoneadm -z zone01 boot
```

※ zone を起動し、動作確認をします。

## 2) zone01 を停止する

```
global02# zlogin zone01 'shutdown -y -g0 -i0'
```

## 2-3. Solaris コンテナ環境におけるリソース操作

## (1) リソース操作の事前準備と確認

## 1) zone の起動

```
global01# zoneadm -z zone01 boot
```

## 2) リソースの設定確認

## 1. zone01 のCPUシェア数の確認

```
global01# prctl -n zone.cpu-shares -i zone zone01
```

※ 実行結果より、privileged 権限の VALUE の値を確認します。

```
global01# prctl -n zone.cpu-shares -i zone zone01
zone: 23: zone01
NAME PRIVILEGE VALUE FLAG ACTION RECIPIENT
zone.cpu-shares
privileged 1 - none -
system 65.5K max none -
```

## 2. zone02 のCPUシェア数の確認

```
global01# prctl -n zone.cpu-shares -i zone zone02
```

※ 実行結果より、privileged 権限の VALUE の値を確認します。

```
global01# prctl -n zone.cpu-shares -i zone zone02
zone: 23: zone02
NAME PRIVILEGE VALUE FLAG ACTION RECIPIENT
zone.cpu-shares
privileged 1 - none -
system 65.5K max none -
```

## 3. サービスの確認

```
global01# svcs -a | grep pools
```

※ svc:/system/pools:default サービスが online になっていること。

```
global01# svcs -a | grep pools
disabled 13:03:25 svc:/system/pools/dynamic:default
online 13:49:17 svc:/system/pools:default
```

## 4. リソースプールの確認

```
global01# poolstat -r all
```

※ min,max,size 値を確認します。  
※ size値は現在のCPUの数であり、必ずmin値とmax値の間に設定されます。

```
global01# poolstat -r all
id pool type rid rset min max size used load
1 pool_1 pset 1 pset_1 8 8 8 0.00 1.01
0 pool_default pset -1 pset_default 8 66K 8 0.00 0.02
```

## (2) シェア数の動的変更

```
global01# prctl -n zone.cpu-shares -r -v 70 -i zone zone01
global01# prctl -n zone.cpu-shares -r -v 30 -i zone zone02
```

## 1. zone01 のCPUシェア数の確認

```
global01# prctl -n zone.cpu-shares -i zone zone01
```

※ zone01 が起動している必要があります。

```
global01# prctl -n zone.cpu-shares -i zone zone01
zone: 22: zone01
NAME PRIVILEGE VALUE FLAG ACTION RECIPIENT
zone.cpu-shares
privileged 70 - none -
system 65.5K max none -
```

## 2. zone02 のCPUシェア数の確認

```
global01# prctl -n zone.cpu-shares -i zone zone02
```

※ zone02 が起動している必要があります。

```
global01# prctl -n zone.cpu-shares -i zone zone02
zone: 23: zone02
NAME PRIVILEGE VALUE FLAG ACTION RECIPIENT
zone.cpu-shares
privileged 30 - none -
system 65.5K max none -
```

## (3) cpu-cap の設定

## 1) zone02 のリソース構成確認

```
global01# zonecfg -z zone02 info capped-cpu
```

※ 設定されていない場合、何も表示されません。

## 2) zone のリソース構成情報の変更

## 1. cpu-cap 数の新規設定

```
global01# zonecfg -z zone02
zonecfg:zone02> add capped-cpu
zonecfg:zone02:capped-cpu> set ncpu=4
zonecfg:zone02:capped-cpu> end
zonecfg:zone02> exit
```

※ 設定の変更は zone の次回起動時に反映されます。

zone02 のリソース設定に cpu-cap 設定を追加します。

パラメータ「ncpu=」の値には、その zone に占有させたいCPUの個数を小数第2位まで指定可能です。  
(=1CPUを1%単位で指定可能)

## 2. 設定の確認

```
global01# zonecfg -z zone02 info capped-cpu
capped-cpu:
  [ncpus: 4.00]
```

## 設定の確認(別パラメータ)

```
global01# zonecfg -z zone02 info rctl
rctl:
  name: zone.cpu-cap
  value: (priv=privileged, limit=400, action=deny)
```

capped-cpu パラメータを指定すると、内部で zone.cpu-cap パラメータが自動的に設定されます。

パラメータ「limit=」の値は100を1CPU分のリソースとして表示されます。

## 3. zone02 の再起動

```
global01# zlogin zone02 'shutdown -y -g0 -i6'
```

## 4. zone02 の cpu-cap 数の確認

```
global01# prctl -n zone.cpu-cap -i zone zone02
```

```
global01# prctl -n zone.cpu-cap -i zone zone02
zone: 24: zone02
NAME PRIVILEGE VALUE FLAG ACTION RECIPIENT
zone.cpu-cap
privileged 400 - deny -
system 4.29G inf deny -
```

## 3) zone02 の cpu-cap 数の動的変更

## 1. cpu-cap 数の変更

```
global01# prctl -n zone.cpu-cap -r -v 300 -i zone zone02
```

※ 再起動後は元の値に戻ります。

## 2. zone02 の cpu-cap 数の確認

```
global01# prctl -n zone.cpu-cap -i zoneid zone02
```

```
global01# prctl -n zone.cpu-cap -i zone zone02
zone: 24: zone02
NAME PRIVILEGE VALUE FLAG ACTION RECIPIENT
zone.cpu-cap
privileged 300 - deny -
system 4.29G inf deny -
```

## (4) CPUリソースの動的移動

## 1. Resource Pool の確認

```
global01# poolstat -r all
id pool type rid rset min max size used load
1 pool_1 pset 1 pset_1 8 8 8 0.00 0.00
0 pool_default pset -1 pset_default 8 66K 8 0.00 0.01
```

sizeの値(実際に割り当たっているCPU数)を確認して下さい。

## 2. pool\_default の最小CPU数を動的に変更(8→1へ変更)

```
global01# poolcfg -dc 'modify pset pset_default (uint pset.min=1)'
```

## 3. pool\_1 の最大CPU数を動的に変更(8→12へ変更)

```
global01# poolcfg -dc 'modify pset pset_1 (uint pset.max=12)'
```

## 4. CPUリソースの移動

```
global01# poolcfg -dc 'transfer 4 from pset pset_default to pset_1'
```

指定した数のCPUをプロセスセット間で移動します。

あるいは、  
global01# poolcfg -dc 'transfer to pset pset\_1 (cpu 5)'

CPU IDを指定して、特定のCPUをプロセスセット間で移動します。

## 5. リソースプールの状態を確認

```
global01# poolstat -r all
id pool          type rid rset          min max size used load
 1 pool_1        pset  1 pset_1          8  12  12 0.00 0.00
 0 pool_default  pset -1 pset_default    1 66K  4 0.00 0.01
```

sizeの値(実際に割り当たっているCPU数)を確認して下さい。

## 6. CPUリソースの移動

```
global01# poolcfg -dc 'transfer 4 from pset pset_1 to pset_default'
```

## 7. リソースプールの状態確認

```
global01# poolstat -r all
id pool          type rid rset          min max size used load
 1 pool_1        pset  1 pset_1          8  12  8 0.00 0.00
 0 pool_default  pset -1 pset_default    1 66K  8 0.00 0.01
```

sizeの値(実際に割り当たっているCPU数)を確認して下さい。

## 2-4. Solaris コンテナのバッチ適用

### (1) サーバへのコンソール接続

1. サーバのコンソールに接続します

### (2) バッチ適用前の準備

#### 1) zone の追加

##### 1. 新規 zone の構成

```
global01# zonecfg -z zone03 create -a /export/zones_patch/zone03
```

##### 2. zone の組込み

```
global01# zoneadm -z zone03 attach
```

##### 3. zone03 の起動

```
global01# zoneadm -z zone03 boot
```

※ 初期設定は必要ありません。

#### 2) zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が起動している (running) ことを確認します。

```
global01# zoneadm list -vc
ID NAME      STATUS  PATH                                BRAND IP
0  global     running /                                     native shared
13 zone01    running /export/zones/zone01              native shared
14 zone02    running /export/zones_clone/zone02        native shared
15 zone03    running /export/zones_patch/zone03       native shared
```

3) 適用されているパッチの確認

1. global zone のパッチを確認

```
global01# showrev -p | grep XXXXXX-XX
```

※ 適用するパッチが適用されていない(何も出力されない)ことを確認します。

2. non-global zone のパッチを確認

```
global01# zlogin zone01 'showrev -p | grep XXXXXX-XX'
global01# zlogin zone02 'showrev -p | grep XXXXXX-XX'
global01# zlogin zone03 'showrev -p | grep XXXXXX-XX'
```

※ 各 zone に XXXXXX-XX のパッチが適用されていない(何も出力されない)ことを確認します。

4) global zone をシングルユーザーモードで起動します。

1. 全ての zone を停止

```
global01# zlogin zone01 'shutdown -y -g0 -i5'
global01# zlogin zone02 'shutdown -y -g0 -i5'
global01# zlogin zone03 'shutdown -y -g0 -i5'
```

2. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
-	zone01	installed	/export/zones/zone01	native	shared
-	zone02	installed	/export/zones_clone/zone02	native	shared
-	zone03	installed	/export/zones_patch/zone03	native	shared

3. OBP への移行

```
global01# shutdown -y -g0 -i0
```

4. global zone をシングルユーザーモードで起動

```
{0} ok boot -s
```

5. ローカルファイルシステムのマウント

```
global01# mountall -l
```

6. マウントポイントの確認

```
global01# df -h
```

※ 全てのファイルシステムがマウントされていることを確認します。

```
global01# df -h
```

ファイルシステム	サイズ	使用済み	使用可能	容量	マウント先
/dev/dsk/c0d0s0	5.9G	4.6G	1.2G	80%	/
-<省略>-					
/dev/dsk/c0d1s0	7.9G	4.1G	3.7G	53%	/zone_tar_list
/dev/dsk/c0d3s0	5.9G	3.9G	1.9G	67%	/export/zones_clone
/dev/dsk/c0d2s0	5.9G	3.9G	1.9G	67%	/export/zones
/dev/dsk/c0d4s0	5.9G	3.9G	1.9G	67%	/export/zones_patch

7. リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスが停止状態 (disabled) であることを確認します。

```
global01# svcs -a | grep pools
```

disabled	16:02:31	svc:/system/pools:default
disabled	16:02:37	svc:/system/pools/dynamic:default

8. リソースプールサービスの起動

```
global01# svcadm enable svc:/system/pools:default
```

※ シングルユーザーモードでは、リソースプールのサービスは自動で起動されないため手動で起動する必要があります。

9. リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスが起動状態 (online) であることを確認します。

```
global01# svcs -a | grep pools
```

disabled	16:02:37	svc:/system/pools/dynamic:default
online	16:03:38	svc:/system/pools:default

10. zone の状態を確認

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
-	zone01	installed	/export/zones/zone01	native	shared
-	zone02	installed	/export/zones_clone/zone02	native	shared
-	zone03	installed	/export/zones_patch/zone03	native	shared

## (3) 並列パッチ適用の準備

## 1) 並列実行数の設定

## 1. プールの確認

```
global01# poolstat -r all
```

※ pool\_default のCPU数(表示結果の"size"の値)を確認します。

```
global01# poolstat -r all
id pool          type rid rset          min max  size used load
 1 pool_1        pset  1 pset_1           8  8   8  0.00 0.00
 0 pool_default  pset -1 pset_default    8 66K 8  0.00 0.02
```

## 2. 並列実行数の設定

```
global01# vi /etc/patch/pdo.conf
# ident "@(#)pdo.conf 1.1 09/05/20 SM1"
#
# Copyright 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#                                     -<省略>-
# num_proc=<num_of_processes>
#
# where:
#
# num_proc      is the key for this record. It specifies the number of process
#               to fork that will be used by the patching utilities
#
# =            the standard delimiter used for parsing the record
#
# <num_of_processes> the number of processes to fork. The actual number of
#                   processes to fork is determined by the three factors mentioned
#                   in 'Factors determining processes to be forked' in the man page
#                   of pdo.conf (4)
#
# num_proc=3 # default entry
```

ここでは、non-global zone の環境数を指定します。

※ 最大値は、global zone が利用可能なオンラインCPU(スレッド)数の1.5倍です(リソースプールが構成されている環境では、pool\_default のCPU数になります)。例えば、pool\_default が8CPU構成の場合、num\_proc の最大値は12になります。

## (4) コンテナ環境の並列パッチ適用

## 1) パッチの適用

```
global01# timex patchadd XXXXXX-XX
```

※ timex コマンドを使用してパッチの適用時間を計測します。

```
global01# timex patchadd XXXXXX-XX
--<省略>-
Patching zone zone01
Checking installed patches...
Executing prepatch script...
Installing patch packages...

Patch XXXXXX-XX has been successfully installed.
See /var/sadm/patch/XXXXXX-XX/log for details
Executing postpatch script...

Patch packages installed:
SUNWxxx

Done!
--<省略>-
Patching zone zone03
Checking installed patches...
Executing prepatch script...
Installing patch packages...

Patch XXXXXX-XX has been successfully installed.
See /var/sadm/patch/XXXXXX-XX/log for details
Executing postpatch script...

real    2:22.74
user    1:07.75
sys     37.90

global01#
```

zone へのパッチ適用のログが一度に出力されます。全ての zone に同時にパッチを適用していることが分かります。

コマンド実行の経過時間  
この場合、約2分22秒です。

## 2) 適用済みパッチの確認

## 1. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
-	zone01	installed	/export/zones/zone01	native	shared
-	zone02	installed	/export/zones_clone/zone02	native	shared

## 2. global zone の再起動

```
global01# shutdown -y -g0 -i6
```

## 3. zone の起動

```
global01# zoneadm -z zone01 boot
global01# zoneadm -z zone02 boot
global01# zoneadm -z zone03 boot
```

## 4. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が起動している (running) ことを確認します。

```
global01# zoneadm list -vc
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
1	zone01	running	/export/zones/zone01	native	shared
2	zone02	running	/export/zones_clone/zone02	native	shared

## 5. global zone のパッチを確認

```
global01# showrev -p | grep XXXXXX-XX
```

※ XXXXXX-XXのパッチが適用されていることを確認します。

```
global01# showrev -p | grep XXXXXX-XX
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV
Requires: Incompatibles: Packages: SUNWxxx
```

## 6. non-global zone のパッチを確認

```
global01# zlogin zone01 'showrev -p | grep XXXXXX-XX'
global01# zlogin zone02 'showrev -p | grep XXXXXX-XX'
global01# zlogin zone03 'showrev -p | grep XXXXXX-XX'
```

※ XXXXXX-XXのパッチが適用されていることを確認します。

```
global01# zlogin zone01 'showrev -p | grep XXXXXX-XX'
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV
Requires: Incompatibles: Packages: SUNWxxx
```

```
global01# zlogin zone02 'showrev -p | grep XXXXXX-XX'
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV
Requires: Incompatibles: Packages: SUNWxxx
```

```
global01# zlogin zone03 'showrev -p | grep XXXXXX-XX'
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV
Requires: Incompatibles: Packages: SUNWxxx
```



## (5) 適用したパッチの削除

## 1) global zone をシングルユーザモードに移行

## 1. zone の停止

```
global01# zlogin zone01 'shutdown -y -g0 -i5'
global01# zlogin zone02 'shutdown -y -g0 -i5'
global01# zlogin zone03 'shutdown -y -g0 -i5'
```

## 2. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID NAME	STATUS	PATH	BRAND	IP
0 global	running	/	native	shared
- zone01	installed	/export/zones/zone01	native	shared
- zone02	installed	/export/zones_clone/zone02	native	shared

## 3. OBP への移行

```
global01# shutdown -y -g0 -i0
```

## 4. シングルユーザモードでの起動

```
{0} ok boot -s
```

## 5. ローカルファイルシステムのマウント

```
global01# mountall -l
```

## 6. マウントポイントの確認

```
global01# df -h
```

※ 全てのファイルシステムがマウントされていることを確認します。

```
global01# df -h
```

ファイルシステム	サイズ	使用済み	使用可能	容量	マウント先
/dev/dsk/c0d0s0	5.9G	4.6G	1.2G	80%	/
-<省略>-					
/dev/dsk/c0d1s0	7.9G	4.1G	3.7G	53%	/zone_tar_list
/dev/dsk/c0d3s0	5.9G	3.9G	1.9G	67%	/export/zones_clone
/dev/dsk/c0d2s0	5.9G	3.9G	1.9G	67%	/export/zones
/dev/dsk/c0d4s0	5.9G	3.9G	1.9G	67%	/export/zones_patch

## 7. リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスが停止状態 (disabled) であることを確認します。

```
global01# svcs -a | grep pools
```

```
disabled 16:18:07 svc:/system/pools:default
disabled 16:18:13 svc:/system/pools/dynamic:default
```

## 8. リソースプールサービスの起動

```
global01# svcadm enable svc:/system/pools:default
```

※ シングルユーザモードでは、リソースプールのサービスは自動で起動されないため手動で起動する必要があります。

## 9. リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスが起動状態 (online) であることを確認します。

```
global01# svcs -a | grep pools
```

```
disabled 16:18:13 svc:/system/pools/dynamic:default
online 16:18:58 svc:/system/pools:default
```

## 10. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID NAME	STATUS	PATH	BRAND	IP
0 global	running	/	native	shared
- zone01	installed	/export/zones/zone01	native	shared
- zone02	installed	/export/zones_clone/zone02	native	shared

## 2) パッチの削除

```
global01# patchrm XXXXXX-XX
```

※ pdo.conf が定義されていると、並列にパッチの削除がおこなわれます。処理時間は、10分程で終了します。

## 3) 適用されているパッチの確認

## 1. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
-	zone01	installed	/export/zones/zone01	native	shared
-	zone02	installed	/export/zones_clone/zone02	native	shared

## 2. global zone の再起動

```
global01# shutdown -y -g0 -i6
```

## 3. zone の起動

```
global01# zoneadm -z zone01 boot
global01# zoneadm -z zone02 boot
global01# zoneadm -z zone03 boot
```

## 4. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が起動している (running) ことを確認します。

```
global01# zoneadm list -vc
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
1	zone01	running	/export/zones/zone01	native	shared
2	zone02	running	/export/zones_clone/zone02	native	shared

## 5. 削除した global zone のパッチを確認

```
global01# showrev -p | grep XXXXXX-XX
```

※ XXXXXX-XX のパッチが削除されている (何も出力されない) ことを確認します。

## 6. 削除した non-global zone のパッチを確認

```
global01# zlogin zone01 'showrev -p | grep XXXXXX-XX'
global01# zlogin zone02 'showrev -p | grep XXXXXX-XX'
global01# zlogin zone03 'showrev -p | grep XXXXXX-XX'
```

※ XXXXXX-XX のパッチが削除されている (何も出力されない) ことを確認します。

## 7. 並列実行数の設定

```
global01# vi /etc/patch/pdo.conf
# ident "@(#)pdo.conf 1.1 09/05/20 SM1"
#
# Copyright 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#                               <省略>
# num_proc=<num_of_processes>
#
# where:
#
# num_proc      is the key for this record. It specifies the number of process
#               to fork that will be used by the patching utilities
#
# =            the standard delimiter used for parsing the record
#
# <num_of_processes> the number of processes to fork. The actual number of
#                   processes to fork is determined by the three factors mentioned
#                   in 'Factors determining processes to be forked' in the man page
#                   of pdo.conf (4)
#
# num_proc=1 # default entry
```

デフォルト値である「1」に修正します。

## (6) パッチ適用前の準備 (従来の方法)

## 1) global zone をシングルユーザモードへ移行

## 1. zone の停止

```
global01# zlogin zone01 'shutdown -y -g0 -i5'
global01# zlogin zone02 'shutdown -y -g0 -i5'
global01# zlogin zone03 'shutdown -y -g0 -i5'
```

## 2. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID NAME	STATUS	PATH	BRAND	IP
0 global	running	/	native	shared
- zone01	installed	/export/zones/zone01	native	shared
- zone02	installed	/export/zones_clone/zone02	native	shared

## 3. OBP への移行

```
global01# shutdown -y -g0 -i0
```

## 4. シングルユーザモードでの起動

```
{0} ok boot -s
```

## 5. ローカルファイルシステムのマウント

```
global01# mountall -l
```

## 6. マウントポイントの確認

```
global01# df -h
```

※ 全てのファイルシステムがマウントされていることを確認します。

```
global01# df -h
```

ファイルシステム	サイズ	使用済み	使用可能	容量	マウント先
/dev/dsk/c0d0s0	5.9G	4.6G	1.2G	80%	/
-<省略>-					
/dev/dsk/c0d1s0	7.9G	4.1G	3.7G	53%	/zone_tar_list
/dev/dsk/c0d3s0	5.9G	3.9G	1.9G	67%	/export/zones_clone
/dev/dsk/c0d2s0	5.9G	3.9G	1.9G	67%	/export/zones
/dev/dsk/c0d4s0	5.9G	3.9G	1.9G	67%	/export/zones_patch

## 7. リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスが停止状態 (disable) であることを確認します。

```
global01# svcs -a | grep pools
```

```
disabled 16:33:36 svc:/system/pools:default
disabled 16:33:43 svc:/system/pools/dynamic:default
```

## 8. リソースプールサービスの起動

```
global01# svcadm enable svc:/system/pools:default
```

※ シングルユーザモードでは、リソースプールのサービスは起動されないため手動で起動する必要があります。

## 9. リソースプールサービスの状態確認

```
global01# svcs -a | grep pools
```

※ リソースプールサービスが起動状態 (online) であることを確認します。

```
global01# svcs -a | grep pools
```

```
disabled 16:33:43 svc:/system/pools/dynamic:default
online 16:34:21 svc:/system/pools:default
```

## 10. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
```

ID NAME	STATUS	PATH	BRAND	IP
0 global	running	/	native	shared
- zone01	installed	/export/zones/zone01	native	shared
- zone02	installed	/export/zones_clone/zone02	native	shared

## (7) 従来の方法でのパッチ適用

## 1) パッチの適用

```
global01# timex patchadd XXXXXX-XX
```

※ timex コマンドを使用してパッチの適用時間を計測します。

```
global01# timex patchadd XXXXXX-XX
                                -<省略>-
Patching global zone
Adding patches...
Checking installed patches...
Executing prepatch script...
Installing patch packages...

Patch XXXXXX-XX has been successfully installed.
See /var/sadm/patch/XXXXXX-XX/log for details
Executing postpatch script...

Patch packages installed:
SUNWxxx

Done!
Patching non-global zones...

Patching zone zone01
Adding patches...

Checking installed patches...
Executing prepatch script...
Installing patch packages...
                                -<省略>-

real    4:03.33
user    1:03.83
sys     30.36

global01#
```

並列パッチの適用時と比べて  
パッチの適用が1環境毎に  
行われていることが  
確認できます。

コマンド実行の経過時間  
約4分3秒です。

## 2) 適用されているパッチの確認

## 1. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が停止している (installed) ことを確認します。

```
global01# zoneadm list -vc
ID NAME      STATUS  PATH                                BRAND IP
0 global    running /                                     native shared
- zone01    installed /export/zones/zone01             native shared
- zone02    installed /export/zones_clone/zone02        native shared
```

## 2. global zone の再起動

```
global01# shutdown -y -g0 -i6
```

## 3. zone の起動

```
global01# zoneadm -z zone01 boot
global01# zoneadm -z zone02 boot
global01# zoneadm -z zone03 boot
```

## 4. zone の状態を表示

```
global01# zoneadm list -vc
```

※ 全ての zone が起動している (running) ことを確認します。

```
global01# zoneadm list -vc
ID NAME      STATUS  PATH                                BRAND IP
0 global    running /                                     native shared
1 zone01    running /export/zones/zone01             native shared
2 zone02    running /export/zones_clone/zone02        native shared
3 zone03    running /export/zones_patch/zone03        native shared
```

## 5. global zone のパッチを確認

```
global01# showrev -p | grep XXXXXX-XX
```

※ XXXXXX-XXのパッチが適用されていることを確認します。

```
global01# showrev -p | grep XXXXXX-XX
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV  
Requires: Incompatibles: Packages: SUNWxxx
```

## 6. non-global zone のパッチを確認

```
global01# zlogin zone01 'showrev -p | grep XXXXXX-XX'
```

```
global01# zlogin zone02 'showrev -p | grep XXXXXX-XX'
```

```
global01# zlogin zone03 'showrev -p | grep XXXXXX-XX'
```

※ XXXXXX-XXのパッチが適用されていることを確認します。

```
global01# zlogin zone01 'showrev -p | grep XXXXXX-XX'
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV  
Requires: Incompatibles: Packages: SUNWxxx
```

```
global01# zlogin zone02 'showrev -p | grep XXXXXX-XX'
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV  
Requires: Incompatibles: Packages: SUNWxxx
```

```
global01# zlogin zone03 'showrev -p | grep XXXXXX-XX'
```

```
Patch: XXXXXX-XX Obsoletes: YYYYYY-YY, ZZZZZZ-ZZ, VVVVVV-VV  
Requires: Incompatibles: Packages: SUNWxxx
```

## 【参考】. コンテナ環境の削除

## (1) zone のアンインストール

## 1) zone の停止

```
global02# zoneadm -z zone01 halt
```

※ アンインストールする zone が起動している場合は停止します。

## 2) zone の状態確認

```
global02# zoneadm list -vc
```

※ zone 停止していることを確認します。

```
global01# zoneadm list -vc
```

ID NAME	STATUS	PATH	BRAND	IP
0 global	running	/	native	shared
- zone01	installed	/export/zones/zone01	native	shared

## 3) zone のアンインストール

```
global02# zoneadm -z zone01 uninstall
```

※ -F オプションを付けると強制的に処理を実行します。

## 4) zone の状態確認

```
global02# zoneadm list -vc
```

※ zone の状態が configured であることを確認します。

```
global01# zoneadm list -vc
```

ID NAME	STATUS	PATH	BRAND	IP
0 global	running	/	native	shared
- zone01	configured	/export/zones/zone01	native	shared

## (2) zone の構成情報の削除

## 1) zone の構成情報の削除

```
global02# zonecfg -z zone01 delete
```

※ 削除の確認が表示されるので、y を押して削除します。

## 2) zone の状況を確認

```
global02# zoneadm list -vc
```

※ 削除されたことを確認します。

```
global01# zoneadm list -vc
```

ID NAME	STATUS	PATH	BRAND	IP
0 global	running	/	native	shared

参考) zone を incomplete 状態にするコマンド

```
global02# zoneadm -z zone01 mark incomplete
```

※ zone 停止状態で実行してください。

## (3) リソースプール構成の削除

## 1) リソースプール構成の初期化

```
global02# pooladm -x
```

※ リソースプールは、pool\_default のみの状態になります。

## 2) リソースプールサービスの停止

```
global02# svcadm disable svc:/system/pools:default
```

```
global02# svcadm disable svc:/system/pools/dynamic:default
```

## 3) リソースプールの状態確認

```
global02# svcs -a | grep pools
```

※ リソースプールサービスが停止していることを確認します。