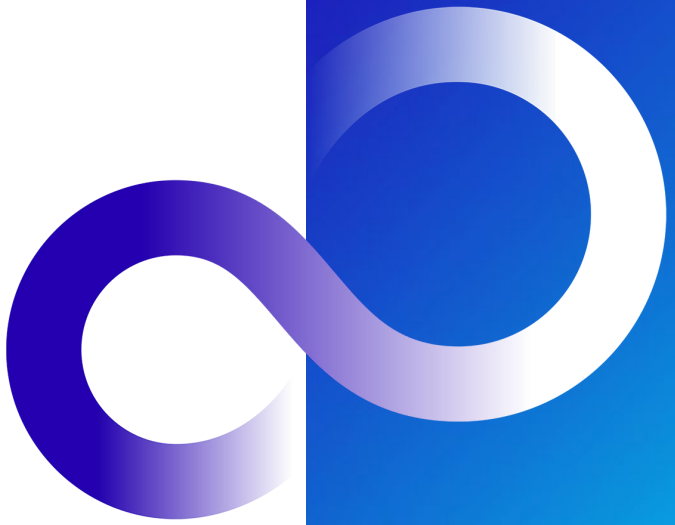
# Oracle Solarisにおける CPUリソースの制限方法

2019年9月

第1.1版

富士通株式会社

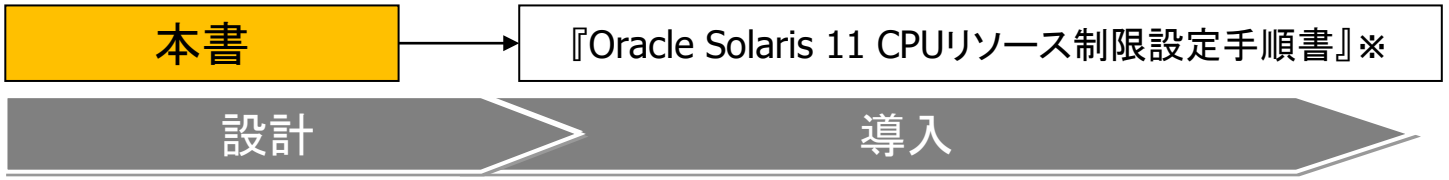


# はじめに

FUJITSU

- 本書の読み方
  - 本書の内容
    - Oracle Solaris環境で利用できるCPUリソースの制限方法を紹介しています。
    - アプリケーションのCPUライセンス形態（利用CPU単位での課金）の検討などにご利用ください。
    - 本書はOracle Solaris 11環境を前提に記載しています。
  - 留意事項
    - 本書では Oracle Solaris を Solaris と記載することがあります。
    - 本書では Oracle Solaris ゾーンを ゾーン、non-global zone と記載することがあります。
- ドキュメントの位置付け

本書の位置付けです。




※具体的な設定手順については、こちらを参照してください。

# 目次

- はじめに
- Oracle Solaris のCPUリソース制御の概要
- CPUリソース制限の設定と確認方法
- <参考>Oracle Solarisのプロジェクトについて

FUJITSU

# Oracle Solaris のCPUリソース制御の概要

- CPUリソース制御方法の分類
- OS環境のプロセス管理
- OS環境でのCPUリソース制限
- 各CPUリソース制御方式の特長

# CPUリソース制限方法の分類

FUJITSU

- SPARC Enterprise/Solaris で実現可能なCPUリソースの制限方法

| | ハードウェア<br>パーティション | Oracle VM Server for<br>SPARC（旧 LDoms） | Solaris OS |
|---|---|---|---|
| 最大分割可能数 | △<br>2～24分割(機種によって異なる) | ○<br>128分割*(機種によって異なる) | ◎<br>Solaris ゾーンにより<br>8,191分割(全機種) |
| CPUリソース<br>配分単位 | △<br>XSB（1CPU)単位 | ○<br>CPUスレッド単位 | 【リソースプール方式】:○ CPUスレッド単位<br>【CPUキャップ方式】:◎ %単位 |

本書での
記載範囲

Solaris OSでは、標準機能のリソースプールやCPUキャップ機能を利用してアプリケーションが利用するCPUリソースを制限することが可能です。

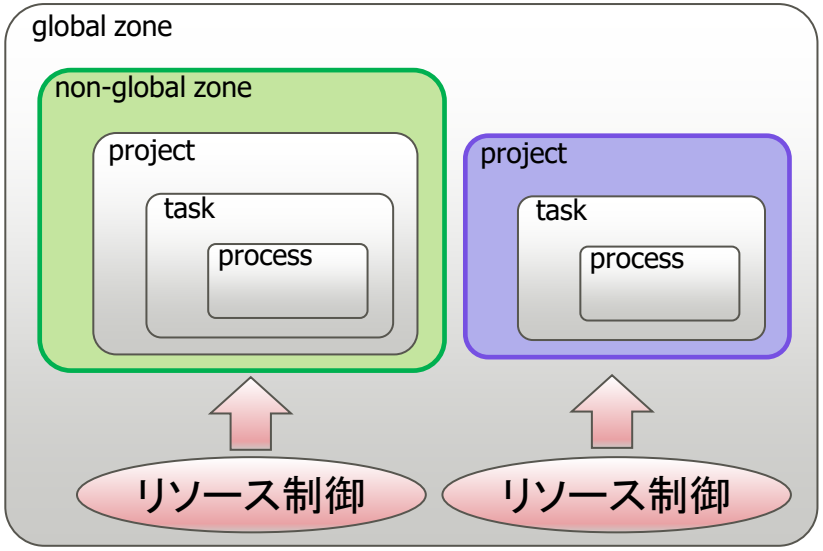
本書では、Solaris OS環境上で実現可能なCPUリソースの分割方法について記載していますが、ハードウェアパーティションやOracle VM Server for SPARCと組み合わせて実現することも可能です。

＊　1個の制御ドメイン＋127個のゲストドメインを構築可能

# OS環境のプロセス管理

- アプリケーション（プロセス）の動作環境

  Solaris環境で実行されるプロセスは、ゾーンまたはプロジェクトの単位で管理されています。




・ゾーンやプロジェクトは、Solaris上で実行される複数のプロセスを纏めて管理する単位です。これらの単位でプロセスの管理（リソース制限）が可能となります。

・ゾーンが構成されている環境ではゾーン単位、ゾーンが構成されていない環境ではプロジェクト単位が最上位の管理単位となります。

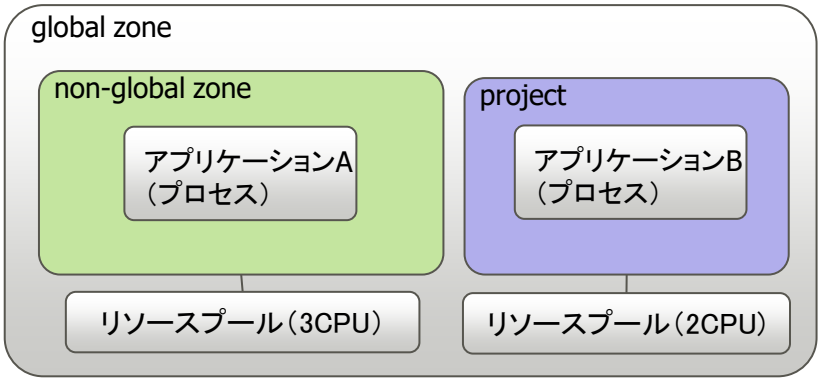
Solaris環境のCPUリソース制御は、最上位の単位であるゾーン（non-global zone）またはプロジェクト（project）に対して実行が可能です。
ゾーンやプロジェクトのCPUリソース制御方法として、リソースプール機能またはCPUキャップ機能を利用します。

# OS環境でのCPUリソース制限

FUJITSU

## リソースプール方式

- ✓ CPUリソースを纏めたリソースプールをゾーンまたはプロジェクトに結び付ける方式。

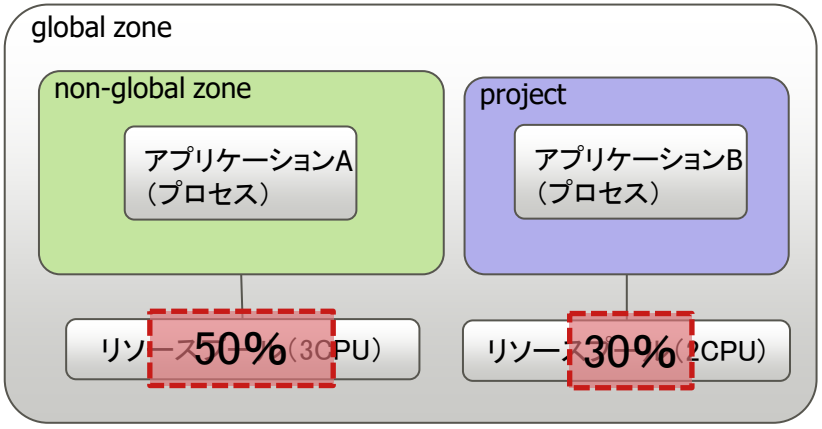



例）ゾーン上のアプリケーションAに3CPU、global zone上のアプリケーションBに2CPUのリソースを割り当て。

利用可能なCPUリソース
- アプリケーションA ： 3CPU
- アプリケーションB ： 2CPU

## CPUキャップ方式

- ✓ ゾーンまたはプロジェクトが利用可能なCPUの利用率を設定する方式。



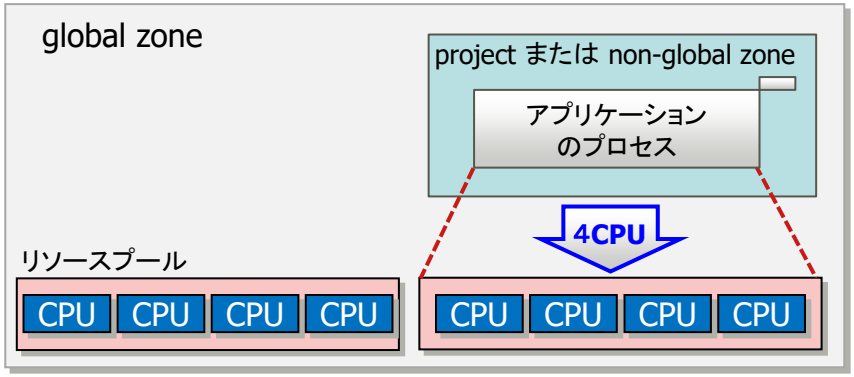

例）ゾーン上のアプリケーションAに3CPU、global zone上のアプリケーションBに2CPUのリソースを割り当て、それぞれに50%、30%のCPUキャップを設定。

利用可能なCPUリソース
- アプリケーションA ： 3CPU × 0.5 = 1.5CPU
- アプリケーションB ： 2CPU × 0.3 = 0.6CPU

# 各CPUリソース制限方式の特長

## リソースプール方式

- 事前にリソースプールを構成して、ゾーンorプロジェクトのパラメーターにリソースプールを指定する。

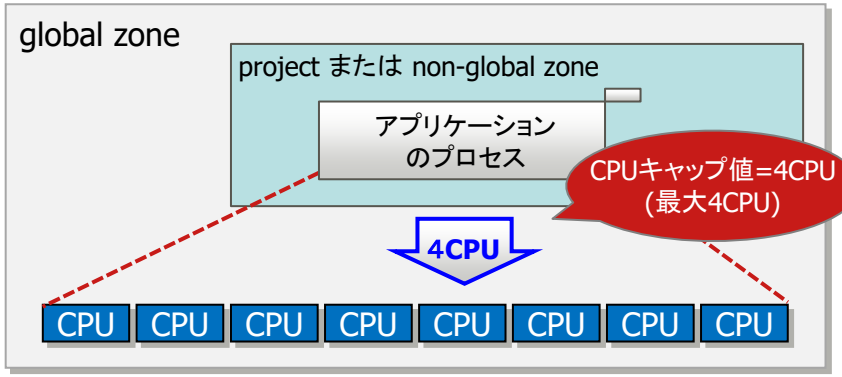



- ゾーンまたはプロジェクトから利用可能なCPUリソースはリソースプール内のCPUのみとなる。
- global zoneのCPUリソースとは別リソースとして確保されます。

| | 認識されるCPU数 | 使用可能なCPU数 |
|---|---|---|
| non-global zone（or project） | 4 | 4 |
| global zone | 8 | 4 |

## CPUキャップ方式

- ゾーンorプロジェクトのパラメーターにCPUキャップ値（CPU最大使用量）を指定する。




- ゾーンまたはプロジェクトから使用可能なCPUリソースはCPUキャップ値が上限となる。
- （リソースプールを指定しない場合）global zoneのCPUリソースを共有します。

| | 認識されるCPU数 | 使用可能なCPU数 |
|---|---|---|
| non-global zone（or project） | 8 | 4* |
| global zone | 8 | 8 |

＊物理的なCPU数ではなく、利用率としての4CPU分

・CPUリソースを占有したい場合 ⇒ リソースプール方式
・CPUリソースの上限を設定したい場合 ⇒ CPUキャップ方式

FUJITSU

# CPUリソース制限の設定と確認方法

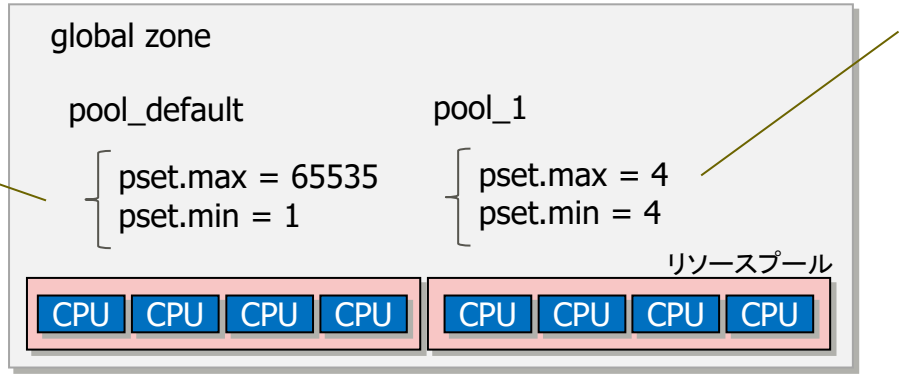
- リソースプールの設定
- リソースプールの確認
- CPUキャップの設定
- CPUキャップの確認

# リソースプールの設定

参照：手順書（P2～6）
「2-1.リソースプール方式」

- リソースプール方式
  - リソースプールの作成
    - global zone が認識しているCPUの内、アプリケーションに割り当てたいCPU数をリソースプールに割り当て
      - リソースプールに割り当てるCPU数は、プロセッサセットのパラメーターに設定
      - パラメーターに、最大CPU数（pset.max値）と最少CPU数（pset.min値）を定義
      - pset.max値と pset.min値を同じ値にすると、リソースプール構成時のCPU数が固定
      - SolarisにおいてCPUリソースはスレッド単位で認識

・リソースプール「pool_default」は、リソースプールに割り当てられていないCPUのうち、最少1CPU、最大65535CPU利用する設定




・リソースプール「pool_1」に4CPUを割り当て

※デフォルトで設定されているpool_defaultは、システムボリューム用のリソースプールであり、削除は不可

  - アプリケーションが動作する環境（global zone または non-global zone）に応じてリソースプールを指定
    - アプリケーションの動作する環境が
      - global zoneの場合 ⇨プロジェクトに設定（例.project.pool=pool_1）
      - non-global zoneの場合 ⇨non-global zoneに設定（例.set pool=pool_1）

機種によって1コアあたりのスレッド数が異なるため、CPU数の設定に注意してください。

# リソースプールの確認

FUJITSU

参照：手順書（P13～18）
「3-1.リソースプール方式」

## ■ リソースプールの確認コマンド

- リソースプールのCPU数の確認 [poolstat(1M)]

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

Point 各リソースプールに含まれるCPUの数（「size」の値）を確認

※上記は「pool_1,pool_default 共に4CPUのリソース」を持つことを表しています。

- プロジェクトが利用するリソースプールの確認 [id(1M)][poolbind(1M)]

```
$ id -p
uid=1000(user01) gid=1000(group01) projid=100(user.user01)
$ poolbind -q $$
24473 pool_1
```

※現在のプロジェクト名を確認
（user.user01プロジェクト）

- ゾーンが利用するリソースプールの確認 [zlogin(1)][poolbind(1M)]

```
# zlogin zone01
[Connected to zone 'zone01' pts/2]
Oracle Corporation      SunOS 5.11      11.0    November 2011
# poolbind -q $$
24542 pool_1
```

Point 実行中のプロセスが利用しているリソースプールを確認（「$$」指定でログインプロセス）
「$$」の代りに確認対象のプロジェクトが実行している任意のプロセスIDを指定することも可能。

※上記は「プロジェクト、ゾーンがpool_1を利用している」ことを表しています。

上記は、OS稼働状態での確認方法です。設定ファイルの確認は下記を参照ください。
「Oracle Solaris 11 CPUリソース制限設定手順書」

# CPUキャップの設定

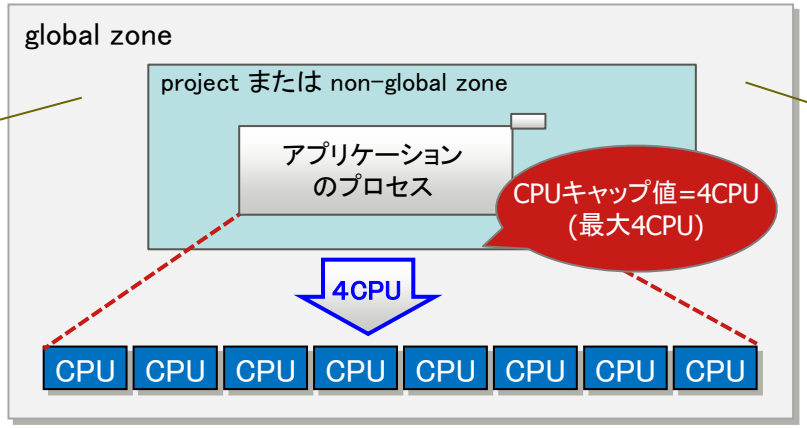
FUJITSU

参照：手順書（P7～12）
「2-2.CPUキャップ方式」

- CPUキャップ方式
  - cpu-capパラメーターの設定
    - アプリケーションが動作する環境に応じて、cpu-capパラメーターを設定する。
    - リソースプールと組み合わせて設定も可能。
    - アプリケーションの動作する環境が、
      - global zoneの場合⇨プロジェクトに設定（例. project.cpu-cap=(privileged,400,deny)）
      - non-global zoneの場合⇨non-global zoneに設定（例.set ncpus=4）

・プロジェクトの場合

```
# projmod -K 'project.cpu-cap=(privileged,400,deny)' user.root
```




・ゾーンの場合

```
# zonecfg -z zone01
zonecfg:zone01> add capped-cpu
zonecfg:zone01:capped-cpu> set ncpus=4
zonecfg:zone01:capped-cpu> end
zonecfg:zone01>exit
```

CPUライセンスがコア単位の場合は、機種によって1コアあたりのスレッド数が異なるため、設定値に注意してください。

# CPUキャップの確認

FUJITSU

参照：手順書（P19～21）
「3-2.CPUキャップ方式」

## CPUキャップの確認コマンド

- プロジェクトに設定されたCPUキャップ値の確認 [prctl(1)]

```
# prctl -n project.cpu-cap -i project user.user01
project: 100: user.user01
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
project.cpu-cap
        usage              0
        privileged       400      -   deny                                 -
        system          4.29G     inf   deny                                 -
```

Point privilegedのVALUE値（1CPU=100で表示）を確認

※上記は「user.user01プロジェクトのCPUリソース上限が4CPU」を表しています。

- ゾーンに設定されたCPUキャップ値の確認 [prctl(1)]

```
# prctl -n zone.cpu-cap -i zone zone01
zone: 3: zone01
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
zone.cpu-cap
        usage              0
        privileged       400      -   deny                                 -
        system          4.29G     inf   deny                                 -
```

Point privilegedのVALUE値（1CPU=100で表示）を確認

※上記は「zone01のCPUリソースの上限が4CPU」を表しています。

上記は、OS稼働状態での確認方法です。設定ファイルの確認は下記を参照ください。
「Oracle Solaris 11 CPUリソース制限設定手順書」

FUJITSU

# <参考>

- Oracle Solaris のプロジェクトについて

# Oracle Solaris のプロジェクトについて

FUJITSU

- プロジェクトを利用してゾーンと同様に、その環境内で実行されるプロセスに対して共通の資源パラメーターを設定することが可能となります。

・プロジェクトは、project データベース(/etc/project)によって構成されます。
・資源パラメーターは、project データベースエントリの最後のフィールドで設定します。
・アプリケーションに紐付ける場合は、実行プロセスが属するプロジェクトに設定する必要があります。

《設定例(/etc/projectファイル)》

```
user.guest:100::::project.cpu-cap=(privileged,400,deny)
   (A)                (B)          (C)  (D)(E)
```

(A):プロジェクト名
(B):資源制御名
(C):特権レベル(basic 、privileged 、system)
　basic(基本値)　　ー 呼び出し元プロセスの所有者が変更できる
　privileged(特権値)ー特権を持っている呼び出し元 (スーパーユーザー) だけが変更できる
　system(システム値)ーオペレーティングシステムによる処理が実行されている間は、固定される
(D):しきい値
(E):特定のしきい値に対応付けられたアクション(deny、signal=シグナル名)
　deny　ーしきい値を超える量の資源要求を拒否できる
　signal　ーしきい値に達した場合は、違反プロセスまたは監視プロセスにシグナルを送信できる

・特にマニュアル等に指示がない場合は、特権レベルに privileged、アクションに deny を設定してください。
・/etc/projectファイルの編集は直接ファイル編集しても有効になりますが、オペミスを防ぐためコマンド(project(1)、projadd(1M),projmod(1M))による編集を推奨します。

# 改版履歴

| 改版年月 | 版数 | 改版内容 |
|---|---|---|
| 2012.11 | 1.0 | 新規作成 |
| 2019.09 | 1.1 | 誤記修正 |

# 商標について

FUJITSU

## 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、SPARC64ロゴ、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。

FUJITSU