

ORACLE®

VM

Oracle VM Server for x86での ハード・パーティショニング

Oracle ホワイト・ペーパー | 2016 年 7 月



はじめに

このドキュメントでは、Oracle VM Server for x86 によるハード・パーティショニング、および、[パーティション環境のオラクル・ライセンス・ポリシー](#)に準拠したハード・パーティショニングの使用方法について説明します。

CPUコアとCPUスレッド

x86 ベース・システムの場合、CPU コア（ハイパースレッディング無効）またはコア内の CPU スレッド（ハイパースレッディング有効）が、ハイパーバイザーまたはベア・メタル・オペレーティング・システムにおいて物理 CPU として表示されます。vCPU（仮想 CPU）は、ゲスト仮想マシンでは CPU として表示されます。ゲストはこれらの vCPU 上でアプリケーションをスケジューリングし、ハイパーバイザーは物理 CPU コアまたはスレッドを介してこれらの vCPU をスケジューリングします。ゲストの vCPU はすべて対称型です。CPU pinning の使用などのスケジューリング・パラメータが変更されない限り、Oracle VM Server はこれらを同等に処理します。

Oracle VM は、ハード・パーティショニングのための高度な機能を提供します。これは CPU pinning とも呼ばれます。ハード・パーティショニングとは、vCPU を物理 CPU スレッドまたはコアにバインドし、これらの vCPU が指定以外の物理 CPU（スレッドまたはコア）でスケジューリングされないようにすることです。

オラクルのハード・パーティショニング・ライセンス

オラクルのハード・パーティショニング・ライセンス要件に準拠するには、このホワイト・ペーパーで説明する指示に従って、vCPU を物理 CPU スレッドまたはコアにバインドする必要があります。

CPU pinning 仮想マシンを他の Oracle VM Server にライブ移行することは、ハード・パーティショニング・ライセンスの条件下では許可されていません。したがって、Oracle VM Release 3において、CPU pinning ゲストを実行しているサーバーは、DRS（Distributed Resource Scheduler）および DPM（Distributed Power Management）ポリシーに含めることはできません。

Oracle VM サーバー・プールでライブ移行を使用する場合、ハード・パーティショニング・ライセンスは適用されません。Oracle Software を実行する仮想マシンの数を決め、同数の物理サーバーにライセンス付与します。ライセンス付与は、CPU コア数が最も多いサーバーから始め、最大でプール内の総物理サーバー数までです。たとえば、サーバー・プール内に 32 台のサーバーがあり、Oracle Software を実行する仮想マシンが 20 ある場合、プール内の最も大きな物理サーバーから 20 台に対してライセンスを付与する必要があります。32 台の物理サーバーからなるプール内で、Oracle Software を実行する仮想マシンが 50 ある場合は、プール内の 32 台の物理サーバーにしかライセンスを付与する必要はありません。

Oracle Software を実行しない仮想マシンをサーバー・プール内でライブ移行する場合は、Oracle Software のハード・パーティショニングは関係しません。また Oracle Software ライセンスの算出方法に影響しません。

“Trusted Partition”により、ライブ移行の制約なしにライセンスの部分的な付与が可能になりますが、これが利用できるのは、[パーティション環境のオラクル・ライセンス・ポリシー](#)に掲載されている認定済みの Oracle Engineered Systems のみです。

Oracle VMにおけるCPUトポロジについて

サーバー・ハードウェアのサマリーの取得

Oracle VM Server 上で xm info コマンドを実行すると、サーバー・ハードウェアの基本的な CPU 構成を出力できます。次の出力行で、システムの CPU の詳細を確認してください。

```
# xm info
...
nr_cpus :8
nr_nodes :1
cores_per_socket :4
threads_per_core :2
cpu_mhz :3200
...
```

このサーバーは、コアあたり 2 スレッド、4 コアの单一ソケットを持ちます。合計 8"CPU"です。したがって、CPU 0~7 は実際にはスレッド 0~7 になります。

```
# xm info
...
nr_cpus :12
nr_nodes :1
cores_per_socket :6
threads_per_core :2
...
```

このサーバーは、コアあたり 2 スレッド、6 コアの单一ソケットを持つため、合計 12"CPU"です。

CPU トポロジの取得

xenpm コマンドにより、特定のサーバーのスレッド、コア、ソケットのトポロジを出力できます。

```
# xenpm get-cpu-topology
CPU      core      socket      node
CPU0      0          0          0
CPU1      0          0          0
CPU2      1          0          0
CPU3      1          0          0
```

上の例は、コアあたり 2 スレッド、2 コアの单一ソケット・マシンを示しています。CPU0 は、コア 0 のスレッド 0、CPU1 はコア 0 のスレッド 1、CPU2 はコア 1 のスレッド 0、CPU3 はコア 1 のスレッド 1 です。仮想マシン構成ファイル (vm.cfg) で cpus="0,1"とした場合は、コア 0 で VM が実行されます。vm.cfg で cpus="0-3"とした場合は、コア 0 とコア 1 の両方のコアで仮想マシンが実行されます。

```
# xenpm get-cpu-topology
CPU      core      socket      node
CPU0      0          0          0
CPU1      0          0          0
CPU2      1          0          0
CPU3      1          0          0
CPU4      2          0          0
CPU5      2          0          0
CPU6      3          0          0
CPU7      3          0          0
```

上の例は、コアあたり 2 スレッド、4 コアの單一ソケット・サーバーの場合です。CPU0 はコア 0 のスレッド 0 にマッピングし、CPU1 はコア 0 のスレッド 1 にマッピングするという具合になります。vm.cfg ファイルで cpus="4-7"とした場合は、コア 2 と 3 で仮想マシンが実行されます。

```
# xenpm get-cpu-topology
CPU      core      socket      node
CPU0      0          0          0
CPU1      0          0          0
CPU2      1          0          0
CPU3      1          0          0
CPU4      2          0          0
CPU5      2          0          0
CPU6      8          0          0
CPU7      8          0          0
CPU8      9          0          0
CPU9      9          0          0
CPU10    10         0          0
CPU11    10         0          0
```

上の例は、ハイパースレッディングが有効化した、6 コアの單一ソケット・サーバーを示しています。

vCPUが物理CPUにバインドされたCPUトポロジの取得

xm vcpu-list コマンドにより、どの仮想 CPU がどの物理 CPU で実行しているのかを示すサマリーが表示されます。

```
# xm vcpu-list 1
Name                   ID  VCPU  CPU  State  Time(s)  CPU  Affinity
0004fb00000600007c351fa24276c63f  1      0      5  -b-  4673.6  5-6
0004fb00000600007c351fa24276c63f  1      1      5  -b-  4534.0  5-6
```

仮想マシンまたはドメイン ID をコマンド `xm vcpu-list 1` のように引数に与えると、そのゲストの情報を取得できます。上の例では、ゲストは 2 個の仮想 CPU を持ち、この時点で両方とも物理 CPU5 (この場合はスレッド) で実行されています。CPU Affinity 列には 5-6 と表示されており、このことは、両方の仮想 CPU がスレッド 5 または 6 で実行されていることを意味します。これは、ゲストがこれら 2 つのスレッドに固定されていることを示します。`xenpm get-cpu-topology` の情報を組み合わせると、この場合、CPU5 がコア 2 のスレッド 1、CPU6 がコア 8 のスレッド 0 であることがわかります。したがって、この 2vCPU のゲストは、2 つの別々の物理コアに固定されています。

```
# xm vcpu-list
Name           ID  VCPU  CPU  State  Time(s)  CPU  Affinity
0004fb00000600007c351fa24276c63f  1    0    5  -b-   4676.8  5-6
0004fb00000600007c351fa24276c63f  1    1    5  -b-   4537.0  5-6
Domain-0       0    0    0  -b-   932.1   any  cpu
Domain-0       0    1    6  -b-   1168.0  any  cpu
Domain-0       0    2    7  -b-   1010.8 any  cpu
Domain-0       0    3    11  -b-   903.0   any  cpu
Domain-0       0    4    8  -b-   494.2   any  cpu
Domain-0       0    5    9  r--   773.8   any  cpu
Domain-0       0    6    1  -b-   522.7   any  cpu
Domain-0       0    7    2  -b-   785.1   any  cpu
Domain-0       0    8    4  -b-   473.8   any  cpu
Domain-0       0    9    3  -b-   728.1   any  cpu
Domain-0       0   10    10  -b-   490.8   any  cpu
Domain-0       0   11    0   r--  1219.6  any  cpu
```

これは同じシステムですが、`xm vcpu-list` に引数が指定されていません。また、`dom0` ゲストが表示されています。この例でわかるように、`dom0` はどの物理スレッド上でも実行でき、CPU Affinity は `any cpu` になっています。つまり、仮想 CPU はどの物理スレッド上でもスケジューリングできるので、固定化またはパーティショニングは行われていません。

Oracle VM 3：ハード・パーティショニングの構成

Oracle VM 3 を使用しながら、Oracle VM ユーティリティ (`ovm_vmcontrol`) を使用して、ハード・パーティショニングを設定できます。

Oracle VM 3 ユーティリティを使用したハード・パーティショニングの設定

Oracle VM 3 ユーティリティ (`ovm_vmcontrol`) を使用して、Oracle VM Manager 3 を介して仮想マシンの CPU/vCPU バインディングの設定および取得をできます。

Oracle VM 3 ユーティリティは、一連の基本的な管理タスクを実行できるコマンドライン・スクリプトの集合です。Oracle VM ユーティリティは、[My Oracle Support](#) からパッチ ID [13602094](#) で検索し、zip ファイルとしてダウンロードできます。Oracle VM ユーティリティのバージョンは、Oracle VM のバージョンに関連しているので、パッチの `readme` ファイルを確認してください。また [Oracle VM ドキュメント](#) の管理者ガイドも参照してください。

```
# ./ovm_vmcontrol -u admin -p Manager1 -h oracle_vm_manager_hostname -v apitest -c
  getvcpu
Oracle VM VM Control utility 2.1.
Connecting to OVM Manager using Web Service.
Connected.
OVM Manager version:3.4.1.1369
Command : getvcpu
Getting pinned CPU list...
Current pinned CPU:5,6
```

上の例では、apitest という仮想マシンに対して、このゲストの仮想 CPU がスレッド 5 と 6 にバインドされていることを表示するアクション getvcpu を実行しています。

次のコマンドを実行して、vCPU をコア 0 にバインドしてみましょう。

```
# ./ovm_vmcontrol -u admin -p Manager1 -h oracle_vm_manager_hostname -v apitest -c
  setvcpu -s 0
Oracle VM VM Control utility 2.1.
Connecting to OVM Manager using Web Service.
Connected.
OVM Manager version:3.4.1.1369
Command : setvcpu
Pinning vCPU '0' to VM 'apitest'
Pinning vCPU succeed.
```

場合によっては、仮想マシンの CPU pinning の構成後、CPU pinning を有効にするために、仮想マシンを停止して再起動する必要があります。次に、xm vcpu-list コマンドを実行して、CPU pinning のステータスを見てみましょう。

```
# xm vcpu-list 1
Name                   ID  VCPU    CPU  State  Time(s)  CPU      Affinity
0004fb00000600007c351fa24276c63f    1      0      0  -b-  4687.6  0
0004fb00000600007c351fa24276c63f    1      1      0  -b-  4547.2  0
```

実際、この VM の仮想 CPU は、両方とも **CPU Affinity** になっています。

上の例では、Oracle VM Manager とは別の Oracle Linux ホストから 2.1ovm-utilities を実行しました。これが可能なのは、Oracle VM 3.4 が完全に Web サービス・ベースだからです。

次の項では、Oracle VM 2 の仮想マシン構成ファイル (vm.cfg) を変更して、ハード・パーティショニングを設定する方法を示します。

Oracle VM 2：ハード・パーティショニングの構成

ストレージ・リポジトリ内で仮想マシンを特定します。たとえば、仮想マシンは、/OVS/running_pool/directory-to-virtual-machine に格納されています。次に、該当するゲストまたは仮想マシンの vm.cfg ファイルを変更します。

```
cpus = '0-3'
```

または

```
cpus = '0,1'
```

最初の例では、CPU 0、1、2、3 のみをそのゲストで使用できます。2 番目の例では、CPU 0 と 1 が使用されます。

4 vCPU のゲストに対して、vm.cfg ファイルで cpus = '0' と指定した場合、4 vCPU のすべてが同じ物理 CPU 上でスケジューリングされます。ゲストに 4 vCPU を割り当て、さらに 2 つの CPU を使用したい場合は、8-CPU システムでこの構成に cpus = '0,1' と追加します。

# xm vcpu-list guest1							
Name	ID	VCPU	CPU	State	Time(s)	CPU	Affinity
guest1	4	0	4	-b-	8645.7	any	cpu
guest1	4	1	4	-b-	9843.6	any	cpu

仮想マシン guest1 は 2 vCPU を持つ、これらは 8 つの CPU のどの CPU 上でも実行できます。

# xm vcpu-list guest1							
Name	ID	VCPU	CPU	State	Time(s)	CPU	Affinity
guest1	26	0	0	-b-	8646.6	0	cpu
guest1	26	1	0	-b-	9844.3	0	cpu

仮想マシン guest1 は 2 vCPU を持つ、これらは物理 CPU 0 のみで実行できます。

# xm vcpu-list guest1							
Name	ID	VCPU	CPU	State	Time(s)	CPU	Affinity
guest1	26	0	0	-b-	8647.8	0	cpu
guest1	26	1	1	-b-	9845.0	1	cpu

仮想マシン guest1 は 2 vCPU を持つ、これらは物理 CPU 0、1 でのみ実行できます。

結論

Oracle VM Server for x86 でオラクルのハード・パーティショニング・ライセンス要件を満たすには、仮想マシンを物理 CPU またはコアにバインドする必要があります。そうすることで、指定のコア以外の物理コアでソフトウェアが実行されるのを防ぐことができます。このような場合、仮想マシンはデフォルトのリソース・スケジューリング（サーバーの利用可能なすべての CPU を使用）ではなく、専用の CPU リソースで構成されます。ハード・パーティショニングを使用して、オラクル製品のソフトウェア・ライセンスを制限すると、ライブ移行、DRS、DPM など、その他の制限も加わります。

オラクルの仮想化ソリューションの詳細については、oracle.com/virtualization にアクセスしてください。



CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口
電話 : +1.650.506.7000
ファクシミリ : +1.650.506.7200

Integrated Cloud Applications & Platform Services

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による默示的保証を含め、商品性ないし特定目的適合性に関する默示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle および Java は Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0116

Oracle VM Server for x86 でのハード・パーティショニング
2016 年 7 月

 | Oracle is committed to developing practices and products that help protect the environment