

vSphere リソース管理

Update 1
ESXi 5.0
vCenter Server 5.0

このドキュメントは新しいエディションに置き換わるまで、ここで書いてある各製品と後続のすべてのバージョンをサポートします。このドキュメントの最新版をチェックするには、<http://www.vmware.com/jp/support/pubs> を参照してください。

JA-000790-00

vmware[®]

最新の技術ドキュメントは VMware の Web サイト (<http://www.vmware.com/jp/support/pubs/>) にあります
VMware の Web サイトでは最新の製品アップデートも提供されています。

このドキュメントに関するご意見およびご感想がある場合は、docfeedback@vmware.com までお送りください。

Copyright © 2006–2012 VMware, Inc. 無断転載を禁ず。本製品は、米国著作権法および米国知的財産法ならびに国際著作権法および国際知的財産法により保護されています。VMware 製品には、<http://www.vmware.com/go/patents-jp> に列記されている 1 つ以上の特許が適用されます。

VMware は、米国およびその他の地域における VMware, Inc. の登録商標または商標です。他のすべての名称ならびに製品についての商標は、それぞれの所有者の商標または登録商標です。

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

ヴァイエルムウェア株式会社
105-0013 東京都港区浜松町 1-30-5
浜松町スクエア 13F
www.vmware.com/jp

目次

vSphere リソース管理について	7
1 リソース管理について	9
リソースのタイプ	9
リソースの提供者	9
リソースの消費者	10
リソース管理の目的	10
2 リソース割り当て設定の構成	11
リソース割り当てのシェア	11
リソース割り当ての予約	12
リソース割り当て制限	12
リソース割り当て設定の推奨事項	13
リソース割り当て設定の変更：例	13
アドミッション コントロール	14
3 CPU 仮想化の基礎	15
ソフトウェア ベースの CPU 仮想化	15
ハードウェア アシストによる CPU 仮想化	16
仮想化とプロセッサ固有の動作	16
CPU 仮想化によるパフォーマンスの影響	16
4 CPU リソースの管理	17
プロセッサ情報の表示	17
CPU 構成の指定	17
マルチコア プロセッサ	18
ハイパースレッド	18
CPU アフィニティの使用	21
ホスト電力管理ポリシー	22
5 メモリ仮想化の基礎	25
仮想マシンのメモリ	25
メモリのオーバーコミット	26
メモリの共有	26
ソフトウェア ベースのメモリ仮想化	26
ハードウェア アシストによるメモリ仮想化	27
6 メモリ リソースの管理	29
メモリ オーバーヘッドについて	29
ESXi ホストによるメモリ割り当ての方法	30
アイドル仮想マシンのメモリ税	31

- メモリの解放 31
 - スワップ ファイルの使用 32
 - ホスト キャッシュへのスワップ 34
 - 仮想マシン間でのメモリの共有 35
 - メモリ圧縮 36
 - メモリ使用量の測定とタイプの識別 37
 - メモリの信頼性 38
- 7 ストレージ I/O リソースの管理 39**
- ストレージ I/O コントロールの要件 39
 - ストレージ I/O コントロールのリソースのシェアおよび制限 40
 - ストレージ I/O コントロールのリソースのシェアおよび制限の設定 41
 - ストレージ I/O コントロールの有効化 41
 - ストレージ I/O コントロールのしきい値の設定 42
- 8 リソース プールの管理 43**
- リソース プールを使用する理由 44
 - リソース プールの作成 45
 - リソース プールの編集 46
 - リソース プールへの仮想マシンの追加 46
 - リソース プールからの仮想マシンの削除 47
 - リソース プールの削除 47
 - リソース プールのアドミッション コントロール 47
- 9 DRS クラスタの作成 51**
- アドミッション コントロールと初期配置 52
 - 仮想マシンの移行 53
 - DRS クラスタの要件 55
 - DRS クラスタの作成 56
 - 仮想マシンのカスタム自動化レベルの設定 57
 - DRS の無効化 58
- 10 DRS クラスタを使用したリソース管理 59**
- クラスタへのホストの追加 59
 - クラスタへの仮想マシンの追加 60
 - クラスタからの仮想マシンの削除 61
 - クラスタからのホストの削除 61
 - DRS クラスタの妥当性 62
 - 電力リソースの管理 67
 - DRS アフィニティ ルールの使用 71
- 11 データストア クラスタの作成 77**
- 初期配置と実行中のバランシング 78
 - ストレージの移行の推奨 78
 - データストア クラスタの作成 78
 - ストレージ DRS の有効化と無効化 79
 - データストア クラスタの自動化レベルの設定 79
 - ストレージ DRS の積極性レベルの設定 80

データストア クラスタの要件	81
データストアのデータストア クラスタへの追加と削除	82
12 データストア クラスタを使用したストレージ リソースの管理	83
ストレージ DRS メンテナンス モードの使用	83
ストレージ DRS 推奨の適用	84
仮想マシンのストレージ DRS 自動レベルの変更	85
ストレージ DRS の勤務時間外スケジュールの設定	86
ストレージ DRS 非アフィニティ ルール	87
ストレージ DRS 統計の削除	89
ストレージ vMotion のデータストア クラスタとの互換性	90
13 NUMA システムと ESXi の併用	91
NUMA とは	91
ESXi の NUMA スケジュール設定の機能	92
VMware NUMA 最適化のアルゴリズムと設定	93
NUMA アーキテクチャでのリソース管理	94
仮想 NUMA の使用	94
NUMA 制御の指定	96
14 詳細属性	99
ホストの詳細属性の設定	99
仮想マシンの詳細属性の設定	102
インデックス	103

vSphere リソース管理について

『vSphere リソース管理』では、VMware® ESXi および vCenter® サーバ環境のためのリソース管理について説明します。本書では次の内容に重点を置いて説明しています。

- リソース割り当ておよびリソース管理の概念
- 仮想マシンの属性とアドミッション コントロール
- リソース プールとその管理方法
- クラスタ、vSphere® DRS (Distributed Resource Scheduler)、vSphere DPM (Distributed Power Management)、およびそれらの操作方法
- データストア クラスタ、ストレージ DRS、ストレージ I/O コントロール、およびそれらの操作方法
- リソース管理の詳細オプション
- パフォーマンスについての考慮事項

対象読者

本書は、システムによるリソースの管理方法およびデフォルト動作のカスタマイズ方法を理解する必要があるシステム管理者を対象としています。また、リソース プール、クラスタ、DRS、データストア クラスタ、ストレージ DRS、ストレージ I/O コントロール、または vSphere DPM を理解および使用するユーザーにも、本書が必要になります。

本書は、ユーザーが VMware ESXi および vCenter Server についての実用的な知識があることを前提としています。

リソース管理について

リソース管理について理解するには、その構成要素、目的、およびクラスタ設定における最良の実装方法について認識しておく必要があります。

仮想マシンのリソース割り当て設定（シェア、予約、制限）について、その設定方法と表示方法も含めて説明します。また、アドミッションコントロールのほか、既存リソースに対するリソース割り当て設定の検証プロセスについても説明します。

リソース管理とは、リソースの提供者からリソースの消費者へリソースを割り当てることです。

リソース管理の必要性は、リソースのオーバーコミット、つまり容量を上回る需要から生じ、また、需要と容量が時間とともに変化するという事実からも発生します。リソース管理を使用すると、使用可能な容量をより効率的に使用できるよう、リソースを動的に割り当て直すことができます。

この章では次のトピックについて説明します。

- [リソースのタイプ \(P. 9\)](#)
- [リソースの提供者 \(P. 9\)](#)
- [リソースの消費者 \(P. 10\)](#)
- [リソース管理の目的 \(P. 10\)](#)

リソースのタイプ

リソースには、CPU、メモリ、電力、ストレージ、およびネットワーク リソースなどがあります。

注意 ESXi はホストごとにネットワーク バンド幅とディスク リソースを管理し、その際、ネットワーク トラフィックシェーピングと比例的なシェア メカニズムが、それぞれ使用されます。

リソースの提供者

ホストとクラスタは、データストア クラスタも含め、物理リソースの提供者です。

ホストの場合、使用可能なリソースとは、仮想化ソフトウェアによって使用されるリソースをホストのハードウェア仕様から差し引いた分です。

クラスタは、ホストのグループです。vSphere Client を使用してクラスタを作成し、そのクラスタに複数のホストを追加できます。vCenter Server は、それらのホストのリソースをまとめて管理します。クラスタは、すべてのホストの CPU とメモリをすべて所有します。共通のロード バランシングとフェイルオーバーについて、クラスタを有効にすることができます。詳細については、[第 9 章「DRS クラスタの作成 \(P. 51\)」](#) を参照してください。

データストア クラスタは、データストアのグループです。DRS クラスタ同様、vSphere Client を使用してデータストア クラスタを作成し、そのクラスタに複数のデータストアを追加できます。vCenter Server は、データストア リソースをまとめて管理します。ストレージ DRS を有効にして、I/O 負荷と領域の使用率をバランスできます。[第 11 章「データストア クラスタの作成 \(P. 77\)」](#) を参照してください。

リソースの消費者

仮想マシンは、リソースの消費者です。

作成時に割り当てるデフォルトのリソース設定で、ほとんどのマシンは十分に機能します。仮想マシンの設定をあとで編集して、リソースの提供者の CPU、メモリ、およびストレージ I/O の合計、または保証された CPU とメモリの予約に対する、シェアベースの割合を割り当てることができます。仮想マシンをパワーオンする場合、サーバは、未予約の十分なリソースが使用可能かどうかを確認して、十分なリソースがある場合にのみパワーオンすることを許可します。このプロセスをアドミッション コントロールと呼びます。

リソース プールは、リソースを柔軟に管理するための論理上の抽象概念です。リソース プールは階層にグループ化することができます、使用可能な CPU リソースとメモリ リソースを階層状にパーティショニングするために使用できます。したがって、リソース プールはリソースの提供者と消費者の両方であるとみなすことができます。リソース プールは子リソース プールと仮想マシンにリソースを提供しますが、親のリソースを消費するので、リソースの消費者でもあります。[第 8 章「リソース プールの管理 \(P. 43\)」](#) を参照してください。

ESXi ホストは、次のようないくつかの要素に基づいて、基盤となるハードウェア リソースの一部を各仮想マシンに割り当てます。

- ESXi ホスト（またはクラスタ）用に使用可能なリソースの合計。
- パワーオンする仮想マシンの数、およびそれらの仮想マシンによるリソースの使用率。
- 仮想化の管理に必要なオーバーヘッド。
- ユーザーが定義するリソースの制限。

リソース管理の目的

リソースを管理する場合、その目的が何であるかを認識しておく必要があります。

リソース管理は、リソースのオーバーコミットを解決できるだけでなく、次のことの実現にも役立ちます。

- パフォーマンスの分離：仮想マシンがリソースを独占するのを防ぎ、予測可能なサービス率を保証する。
- 効率的な利用：アンダーコミットされているリソースを活用し、性能低下を抑えたオーバーコミットを行う。
- 容易な管理：仮想マシンの相対的な重要度を制御し、柔軟な動的パーティショニングを提供し、絶対的なサービスレベル アグリーメントを達成する。

リソース割り当て設定の構成

使用可能なリソース容量がリソースの消費者（および仮想化オーバーヘッド）の需要を満たさない場合、管理者は、仮想マシンまたは仮想マシンが配置されているリソース プールに割り当てるリソースの量をカスタマイズする必要がある場合があります。

リソース割り当て設定（シェア、予約、制限）は、仮想マシンに提供する CPU、メモリ、およびストレージリソースの量を決定するために使用します。特に、システム管理者にはリソース割り当てのオプションがいくつかあります。

- ホストまたはクラスタの物理リソースを予約する。
- 仮想マシン用の一定のメモリ量が、ESXi マシンの物理メモリから提供されるようにする。
- 特定の仮想マシンに、常にほかの仮想マシンより高い割合の物理リソースが割り当てられることを保証する。
- 仮想マシンに割り当てることができるリソースの上限を設定する。

この章では次のトピックについて説明します。

- [リソース割り当てのシェア \(P. 11\)](#)
- [リソース割り当ての予約 \(P. 12\)](#)
- [リソース割り当て制限 \(P. 12\)](#)
- [リソース割り当て設定の推奨事項 \(P. 13\)](#)
- [リソース割り当て設定の変更：例 \(P. 13\)](#)
- [アドミッションコントロール \(P. 14\)](#)

リソース割り当てのシェア

シェアは、仮想マシン（またはリソース プール）の相対的な重要度を指定します。ある仮想マシンのリソースのシェアが別の仮想マシンの 2 倍である場合、その仮想マシンは、別の仮想マシンの 2 倍のリソースを消費できます（2 台の仮想マシンがリソースを獲得するために競合する場合）。

シェアは、通常、[高]、[標準]、または [低] で指定され、これらの値によって、それぞれのシェア値が 4:2:1 の割合で指定されます。特定のシェア数（比重を表す）を各仮想マシンに割り当てる場合は、[カスタム] を選択することもできます。

シェアの指定が意味を持つのは、同レベルの仮想マシンまたはリソース プール（つまり、リソース プール階層内で親が同じである仮想マシンまたはリソース プール）に関してだけです。同レベルの場合、予約および制限によって制限される、相対的なシェア値に応じてリソースを共有します。仮想マシンにシェアを割り当てる場合は、その仮想マシンの、パワーオンされたそれ以外の仮想マシンに対する相対的な優先順位を常に指定します。

次の表は、仮想マシンについて、CPU とメモリのデフォルトのシェア値を示しています。リソース プールの場合、CPU とメモリのデフォルトのシェア値は同じですが、そのリソース プールを、4 つの仮想 CPU と 16 GB のメモリを備えた 1 台の仮想マシンと見なして乗算する必要があります。

表 2-1. シェア値

設定	CPU シェア値	メモリ シェア値
高	仮想 CPU あたり 2000 シェア	構成された仮想マシン メモリ 1MB あたり 20 シェア
標準	仮想 CPU あたり 1000 シェア	構成された仮想マシン メモリ 1MB あたり 10 シェア
低	仮想 CPU あたり 500 シェア	構成された仮想マシン メモリ 1MB あたり 5 シェア

たとえば、2 つの仮想 CPU と 1GB の RAM を持ち、CPU とメモリのシェアが [標準] に設定されている SMP 仮想マシンは、 $2 \times 1,000 = 2,000$ シェアの CPU と、 $10 \times 1,024 = 10,240$ シェアのメモリを持ちます。

注意 複数の仮想 CPU を持つ仮想マシンを、SMP (対称型マルチプロセッシング) 仮想マシンと呼びます。ESXi は、仮想マシンあたり最大 32 個の仮想 CPU をサポートできます。

各シェアによって表される相対的な優先順位は、新しい仮想マシンがパワーオンされると変化します。これは、同じリソースプールのすべての仮想マシンに影響を与えます。すべての仮想マシンは、同じ数の仮想 CPU を持ちます。次の例について考えます。

- 2 台の CPU バインドの仮想マシンが、全体の CPU 容量が 8GHz であるホスト上で稼働しています。CPU シェアは [標準] に設定されていて、それぞれ 4GHz を獲得します。
- 3 台目の CPU バインドの仮想マシンがパワーオンされます。その仮想マシンの CPU シェア値は [高] に設定されています。これは、その仮想マシンが、[標準] に設定されているマシンの 2 倍のシェアを持つことを意味します。この新しい仮想マシンは 4GHz を獲得し、ほかの 2 台のマシンはそれぞれ 2GHz だけを獲得します。ユーザーが 3 台目の仮想マシンで、カスタム シェア値を 2,000 に指定した場合も、同じ結果になります。

リソース割り当ての予約

予約は、仮想マシンに保証される最小割り当てを指定します。

vCenter Server または ESXi では、仮想マシンの予約を達成するのに十分な未予約リソースが存在する場合にのみ、仮想マシンをパワーオンできます。サーバは、物理サーバの負荷が大きくても、その量を保証します。予約は、具体的な単位 (メガヘルツまたはメガバイト) で表されます。

たとえば、2GHz が使用可能であり、VM1 に 1GHz の予約を指定し、VM2 に 1GHz の予約を指定するとします。これで、必要な場合に各仮想マシンが 1GHz を獲得できることが保証されます。ただし、VM1 が 500MHz しか使用していない場合、VM2 は 1.5GHz を使用できます。

予約は、デフォルトでは 0 です。仮想マシンが最低限必要な CPU またはメモリを常に使用できることを保証する必要がある場合に予約を指定してください。

リソース割り当て制限

制限は、仮想マシンに割り当てることができる CPU、メモリ、またはストレージ I/O リソースの上限を指定します。

サーバは、予約より多くの量を仮想マシンに割り当てることができますが、システムに未使用のリソースがあっても、制限を超える割り当ては行いません。制限は、具体的な単位 (メガヘルツ、メガバイト、または 1 秒あたりの I/O 処理数) で表されます。

CPU、メモリ、およびストレージ I/O リソースの制限は、デフォルトでは制限なしです。メモリの制限が制限なしのときは、仮想マシンを作成するときに構成されたメモリの量が事実上の制限になります。

ほとんどの場合、制限を指定する必要はありません。制限には、メリットとデメリットがあります。

- **メリット**：少数の仮想マシンから使用をはじめて、ユーザーの要望に対応する場合に、制限を割り当てると役立ちます。仮想マシンを追加するごとに、パフォーマンスは低下します。制限を指定することによって、使用可能なリソースが少なくなることをシミュレートできます。
- **デメリット**：制限を指定すると、アイドルリソースが無駄になることがあります。システムが活用されていない状態で、アイドルリソースが使用可能な場合でも、システムは、制限以上のリソースを仮想マシンが使用することを許可しません。適切な理由がある場合にのみ制限を指定します。

リソース割り当て設定の推奨事項

使用する ESXi 環境に適切なリソース割り当て設定（シェア、予約、制限）を選択します。

次のガイドラインは、仮想マシンのパフォーマンスを向上させるのに役立ちます。

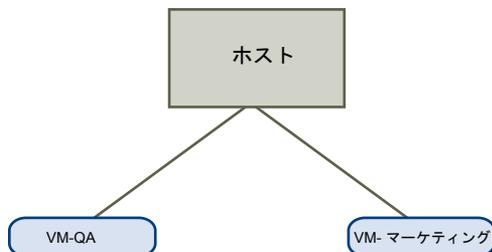
- 合計使用可能リソースが頻繁に変化することが予想される場合は、[シェア] を使用して、仮想マシン間で適正にリソースを割り当てます。[シェア] を使用していて、ホストをアップグレードする場合、各シェアがより多くのメモリ、CPU、またはストレージ I/O リソースを表していても、各仮想マシンの優先順位は変わりません（同じシェア数のままです）。
- [予約] では、ユーザーが使用可能にしたい量ではなく、条件に合った最小の CPU またはメモリの量を指定します。ホストは、仮想マシンのシェア数、需要予測、および制限に基づいて、使用可能な追加のリソースを割り当てます。予約によって表される具体的なリソースの量は、仮想マシンの追加や削除など、環境を変更しても変化しません。
- 仮想マシンの予約を指定する場合、すべてのリソースをコミットしないでください（10% 以上を未予約にしてください）。システム内のすべての容量が完全に予約された状態に近づくほど、アドミッション コントロールに違反せずに予約とリソース プール階層に変更を加えることが困難になっていきます。DRS の有効なクラスタでは、クラスタの容量またはクラスタ内の個々のホストの容量を完全にコミットする予約によって、DRS が仮想マシンをホスト間で移行できなくなることがあります。

リソース割り当て設定の変更：例

次の例は、リソース割り当て設定を変更して、仮想マシンのパフォーマンスを向上させる方法を示しています。

ESXi ホストで、2 台の新しい仮想マシンを作成したとします。この 2 台はそれぞれ、QA（VM-QA）部門とマーケティング（VM-マーケティング）部門用です。

図 2-1. 2 台の仮想マシンを持つ単一ホスト



次の例では、VM-QA が多くのメモリを必要とし、そのために 2 台の仮想マシンのリソース割り当て設定を変更して、次のことを行うものとします。

- システム メモリがオーバーコミットされる場合に、VM-QA が VM-マーケティングの 2 倍のメモリと CPU を使用できるように指定する。VM-QA のメモリシェアと CPU シェアを [高] に設定し、VM-マーケティングでは [標準] に設定する。
- VM-マーケティングが、一定量の保証された CPU リソースを所有するようにする。これは、予約設定を使用して実行できます。

手順

- 1 vSphere Client を起動して vCenter Server システムに接続します。
- 2 [VM-QA]（シェアを変更する仮想マシン）を右クリックして、[設定の編集] を選択します。
- 3 [リソース] タブを選択し、CPU パネルで、[シェア] ドロップダウン メニューから [高] を選択します。
- 4 メモリ パネルで、[シェア] ドロップダウン メニューから [高] を選択します。
- 5 [OK] をクリックします。
- 6 マーケティング仮想マシン（[VM-マーケティング]）を右クリックし、[設定の編集] を選択します。

- 7 CPU パネルで、[予約] の値を目的の数値に変更します。
- 8 [OK] をクリックします。

クラスタの [リソース割り当て] タブを選択して [CPU] をクリックすると、[VM-QA] のシェアがほかの仮想マシンのシェアの 2 倍になっていることがわかります。また、仮想マシンがパワーオンされていないので、[使用済みの予約] フィールドは変化していません。

アドミッション コントロール

仮想マシンをパワーオンすると、まだ予約されていない CPU リソースとメモリ リソースの量をシステムが確認します。システムは、未予約の使用可能なリソースに基づいて、仮想マシンが構成されている予約を保証できるかどうかを判定します（予約がある場合）。このプロセスをアドミッション コントロールと呼びます。

未予約の CPU とメモリが十分に使用可能な場合、または予約がない場合、仮想マシンはパワーオンされます。それ以外の場合は、**リソース不足です** という警告が表示されます。

注意 ユーザー指定のメモリ予約のほかに、仮想マシンごとにオーバーヘッドメモリがあります。この余分なメモリコミットは、アドミッション コントロールの計算に含まれます。

vSphere DPM 機能が有効になっている場合、電力の消費を減らすために、ホストがスタンバイモード（パワーオフ状態）になることがあります。これらのホストが提供する未予約のリソースは、アドミッション コントロールに使用できると見なされます。これらのリソースがないと仮想マシンをパワーオンできない場合、十分なスタンバイホストをパワーオンする推奨が作成されます。

CPU 仮想化の基礎

CPU 仮想化はパフォーマンスを重視し、可能な限りプロセッサ上で直接実行されます。基盤となる物理リソースが可能な限り使用され、必要な場合にだけ仮想化レイヤーが命令を実行して、あたかも物理マシン上で直接実行されているかのように仮想マシンを動作させます。

CPU 仮想化は、エミュレーションと同じものではありません。ESXi は、仮想 CPU の実行にエミュレーションを使用しません。エミュレーションでは、すべての処理がソフトウェア内でエミュレータによって実行されます。ソフトウェアエミュレータにより、プログラムが元々記述されたコンピュータ システム以外のシステム上でそのプログラムを実行できるようになります。エミュレータは、これを行うために、同じデータまたは入力を受け入れて、同じ結果を達成することで、元のコンピュータの動作をエミュレート（再生）します。エミュレーションは移植性を提供し、1 つのプラットフォーム用に設計されたソフトウェアをいくつかのプラットフォームをまたいで実行します。

CPU リソースがオーバーコミットされている場合、ESXi ホストは、すべての仮想マシンにわたって物理プロセッサを時分割するため、指定された数の仮想プロセッサが備わっているかのように各仮想マシンが実行されます。ESXi ホストが複数の仮想マシンを実行する場合、ESXi ホストは物理リソースのシェアを各仮想マシンに割り当てます。デフォルトのリソース割り当て設定では、同じホストに関連付けられているすべての仮想マシンは、1 つの仮想 CPU につき同等の CPU シェアを取得します。つまり、単一プロセッサ仮想マシンには、デュアル プロセッサ仮想マシンのリソースの半分だけが割り当てられます。

この章では次のトピックについて説明します。

- [ソフトウェア ベースの CPU 仮想化 \(P. 15\)](#)
- [ハードウェア アシストによる CPU 仮想化 \(P. 16\)](#)
- [仮想化とプロセッサ固有の動作 \(P. 16\)](#)
- [CPU 仮想化によるパフォーマンスの影響 \(P. 16\)](#)

ソフトウェア ベースの CPU 仮想化

ソフトウェア ベースの CPU 仮想化では、ゲスト アプリケーション コードはプロセッサ上で直接実行されます。このとき、ゲスト特権コードは変換され、変換されたコードがプロセッサ上で実行されます。

変換されたコードはわずかに大きくなり、通常はネイティブバージョンより実行速度が遅くなります。結果として、小さな特権コード コンポーネントを持つゲスト プログラムは、ネイティブに非常に近い速度で実行されます。大きな特権コード コンポーネント（システム呼び出し、トラップ、ページテーブル アップデートなど）を持つプログラムは、仮想化環境では実行速度が遅くなる場合があります。

ハードウェア アシストによる CPU 仮想化

一部のプロセッサでは、CPU を仮想化するハードウェア アシストが提供されます。

このアシストを使用すると、ゲストではゲストモードと呼ばれる個別の実行モードを使用できます。ゲストコードは、アプリケーションコードが特権コードかにかかわらず、ゲストモードで実行されます。場合によっては、プロセッサがゲストモードを終了し、ルートモードを開始します。この場合、ハイパーバイザーがルートモードで実行され、ゲストモードを終了した理由を判断し、必要なアクションを実行し、ゲストモードでゲストを再起動します。

仮想化にハードウェアアシストを使用すると、コードを変換する必要がなくなります。その結果、システム呼び出しやトラップを多用するワークロードが、ネイティブに近い速度で実行されます。一部のワークロード（ページテーブルのアップデートを行うものなど）では、ゲストモードからルートモードへの移行が頻繁に発生します。この移行の回数および移行で費やされる合計時間によっては、ハードウェアアシストによるCPU仮想化の実行速度は大幅に速くなることがあります。

仮想化とプロセッサ固有の動作

VMware ソフトウェアは CPU を仮想化しますが、仮想マシンは実行しているプロセッサの特定のモデルを認識します。

各プロセッサモデルは、提供する CPU 機能が異なる場合があります。仮想マシン内で実行されるアプリケーションは、それらの機能を使用できます。したがって、異なる機能セットを備えたプロセッサ上で実行されているシステム間で、vMotion を使用して仮想マシンを移行させることはできません。この機能をサポートするプロセッサで Enhanced vMotion Compatibility (EVC) を使用すると、この制限を回避できることもあります。詳細は、『vCenter Server and Host Management』ドキュメントを参照してください。

CPU 仮想化によるパフォーマンスの影響

CPU 仮想化によって追加されるオーバーヘッドの量は、ワークロードと使用する仮想化のタイプによって異なります。

アプリケーションの時間のほとんどが、外部イベント（ユーザーの対話、デバイス入力、データ検索など）の待機のためではなく命令の実行に費やされる場合、そのアプリケーションは CPU バインドです。そのようなアプリケーションの場合、CPU 仮想化オーバーヘッドには、実行する必要がある追加命令が含まれます。このオーバーヘッドは、アプリケーション自体が使用できる CPU 処理時間を奪います。通常、CPU 仮想化オーバーヘッドによって、全体的にパフォーマンスが低下します。

CPU バインドでないアプリケーションの場合、CPU 仮想化は CPU 使用率の増加を意味することが多くなります。オーバーヘッドを吸収するのに予備の CPU 容量を使用できる場合、CPU は、全体的なスループットに関して同等のパフォーマンスを提供できます。

ESXi は 1 台の仮想マシンにつき最大 32 の仮想プロセッサ（CPU）をサポートします。

注意 最高のパフォーマンスとリソース使用率を得るためには、単スレッドアプリケーションを（SMP 仮想マシンではなく）ユニプロセッサ仮想マシンに導入してください。

単スレッドアプリケーションは、単一 CPU のみを活用できます。そのようなアプリケーションをデュアルプロセッサ仮想マシンに導入しても、アプリケーションの速度は向上しません。それどころか、ほかの仮想マシンが使用可能な物理リソースを 2 番目の仮想 CPU が使用してしまいます。

CPU リソースの管理

1 つ以上の仮想プロセッサを使用して、それぞれ独自のレジスタ セットおよび制御構造を備える仮想マシンを構成できます。

仮想マシンをスケジューリング設定すると、その仮想プロセッサが物理プロセッサ上で実行するようにスケジューリング設定されます。VMkernel リソース マネージャは、物理 CPU 上に仮想 CPU をスケジューリング設定することによって、仮想マシンの物理 CPU リソースへのアクセスを管理します。ESXi は、最大 32 までの仮想 CPU を持つ仮想マシンをサポートします。

この章では次のトピックについて説明します。

- [プロセッサ情報の表示 \(P. 17\)](#)
- [CPU 構成の指定 \(P. 17\)](#)
- [マルチコア プロセッサ \(P. 18\)](#)
- [ハイパースレッド \(P. 18\)](#)
- [CPU アフィニティの使用 \(P. 21\)](#)
- [ホスト電力管理ポリシー \(P. 22\)](#)

プロセッサ情報の表示

vSphere Client または vSphere SDK を使用して、現在の CPU 構成に関する情報にアクセスできます。

手順

- 1 vSphere Client でホストを選択し、[構成] タブをクリックします。
- 2 [プロセッサ] を選択します。

物理プロセッサの数とタイプ、および論理プロセッサの数についての情報を参照できます。

注意 ハイパースレッドシステムでは、各ハードウェアスレッドが論理プロセッサになります。たとえば、ハイパースレッドが有効なデュアル コア プロセッサでは、コアが 2 つで論理プロセッサが 4 つになります。

- 3 (オプション) [プロパティ] をクリックして、ハイパースレッドを無効または有効にすることもできます。

CPU 構成の指定

CPU 構成を指定して、リソース管理を向上できます。ただし、CPU 構成をカスタマイズしない場合、ESXi ホストは、ほとんどの状況で良好に機能するデフォルトを使用します。

次の方法で CPU 構成を指定できます。

- vSphere Client 通じて使用可能な属性と特殊な機能を使用します。vSphere Client のグラフィカル ユーザー インターフェイス (GUI) を使用して ESXi ホストまたは vCenter Server システムに接続できます。

- 特定の状況下で高度な設定を使用します。
- スクリプト化した CPU 割り当てに vSphere SDK を使用します。
- ハイパースレッドを使用します。

マルチコア プロセッサ

ホストが仮想マシンのマルチタスクを実行する場合、マルチコア プロセッサにはさまざまな利点があります。

Intel および AMD は、複数のプロセッサ コアを 1 つの集積回路（通常はパッケージまたはソケットと呼ばれます）に組み合わせたプロセッサをそれぞれ開発しました。当社では、1 つ以上のプロセッサ コアを装備できる 1 つのまとまりを表すのにソケットという用語を使用します。

各プロセッサ コア内には、1 つ以上の論理プロセッサが含まれます。たとえば、デュアルコア プロセッサは、同時に 2 つの仮想 CPU を実行できるため、シングルコア プロセッサのほぼ 2 倍のパフォーマンスを発揮します。同じプロセッサ内のコアは、通常、すべてのコアが使用する最終レベルの共有キャッシュを装備して構成されており、より低速のメインメモリにアクセスする必要性は軽減されます。実行中の仮想マシンが、同じメモリ バス リソースで競合するようなメモリを多用するワークロードを実行している場合、物理プロセッサをメインメモリに接続する共有メモリバスによって論理プロセッサのパフォーマンスが制限されることがあります。

仮想マシンを実行する ESXi CPU スケジューラによって各プロセッサ コアの各論理プロセッサを独立して使用できるため、SMP システムと同様のパフォーマンスが得られます。たとえば、2 方向の仮想マシンは、仮想プロセッサを同じコアに属する論理プロセッサ上で実行することも、異なる物理コア上の論理プロセッサで実行することもできます。

ESXi CPU スケジューラは、プロセッサトポロジと、そのプロセッサ コアと論理プロセッサとの関係を検出します。この情報を使用して、仮想マシンをスケジュール設定し、パフォーマンスを最適化します。

ESXi CPU スケジューラは、ソケット、コア、および論理プロセッサ間の関係も含め、プロセッサトポロジを認識できます。スケジューラはトポロジ情報を使用して、さまざまなソケットへの仮想 CPU の配置を最適化します。これにより、キャッシュ全体の使用率が最大化され、仮想 CPU の移行が最小化されるために、キャッシュのアフィニティが向上します。

アンダーコミットされているシステムでは、ESXi CPU スケジューラはデフォルトですべてのソケットに負荷を分散します。これにより、稼働中の仮想 CPU が使用できるキャッシュの合計量が最大になり、パフォーマンスが向上します。その結果、単一の SMP 仮想マシンの複数の仮想 CPU が複数のソケットに分散されます（ただし、各ソケットが NUMA ノードでもある場合は除きます。その場合、NUMA スケジューラは仮想マシンのすべての仮想 CPU が同じソケット上に存在するよう制限します）。

場合によっては、たとえば、SMP 仮想マシンが仮想 CPU 間での顕著なデータ共有を示している場合は、このデフォルトの動作は次善の策になる可能性があります。そのようなワークロードでは、ESXi ホストがアンダーコミットされていても、同じソケット上にあるすべての仮想 CPU を最終レベルの共有キャッシュ付きでスケジュール設定した方がメリットがある場合があります。そのようなシナリオでは、仮想 CPU を複数のパッケージにわたって分散するデフォルト動作は、仮想マシンの `.vmx` 構成ファイルに次の構成オプションを組み込むことによってオーバーライドできます。

```
sched.cpu.vmpsConsolidate="TRUE".
```

ハイパースレッド

ハイパースレッドテクノロジーを使用すると、単一の物理プロセッサ コアを 2 つの論理プロセッサのように機能させることができます。プロセッサは、同時に 2 つの異なるアプリケーションを実行できます。論理プロセッサと物理プロセッサの混同を避けるために、Intel は物理プロセッサをソケットと呼んでおり、このセクションの説明でもこの用語を使用します。

Intel Corporation は、Pentium IV および Xeon のプロセッサ ラインのパフォーマンスを向上させるハイパースレッドテクノロジーを開発しました。ハイパースレッドテクノロジーにより、単一のプロセッサ コアが 2 つの独立したスレッドを同時に実行できます。

ハイパースレッドはシステムのパフォーマンスを2倍にするわけではありませんが、アイドルリソースを有効に活用することによってパフォーマンスを向上でき、特定の重要なワークロードタイプについてスループットを向上させます。ビジーなコアの1つの論理プロセッサ上で実行中のアプリケーションは、非ハイパースレッドプロセッサ上で単独で実行される場合に取得するスループットの半分よりわずかに多いスループットを期待できます。ハイパースレッドのパフォーマンス向上はアプリケーションに強く依存し、アプリケーションによっては、ハイパースレッドを使用するとパフォーマンスが低下する場合があります。これは、両方の論理プロセッサ間で多くのプロセッサリソース（キャッシュなど）が共有されるためです。

注意 Intel ハイパースレッドテクノロジーを使用するプロセッサでは、メモリキャッシュや機能単位など、コアのリソースの大部分を共有する2つの論理プロセッサを、1つのコアに担当させることができます。このような論理プロセッサを、通常、スレッドと呼びます。

多くのプロセッサはハイパースレッドをサポートしておらず、その結果、1つのコアにつき1つのスレッドだけを備えています。それらのプロセッサの場合、コアの数も論理プロセッサの数に一致します。次の各プロセッサはハイパースレッドをサポートしており、1つのコアにつき2つのスレッドを備えています。

- Intel Xeon 5500 プロセッサ マイクロアーキテクチャに基づいたプロセッサ
- Intel Pentium 4 (HT 有効)
- Intel Pentium EE 840 (HT 有効)

ハイパースレッドと ESXi ホスト

ハイパースレッドが有効になっているホストは、ハイパースレッドがないホストと同様な動作をする必要があります。しかし、ハイパースレッドを有効にする場合は、特定の要素について検討する必要があることがあります。

ESXi ホストは、プロセッサ時間をインテリジェントに管理して、システム内のプロセッサ コア間で負荷が円滑に分散されるように保証します。CPU 0 と 1 はともに第 1 のコア上にあり、CPU 2 と 3 は第 2 のコア上にあるといったように、同じコア上の論理プロセッサは、連続した CPU 番号を持ちます。仮想マシンは、同じコアの2つの論理プロセッサ上よりも、2つの異なるコア上に優先的にスケジューリング設定されます。

論理プロセッサの作業がない場合、論理プロセッサは停止状態になり、論理プロセッサの実行リソースが解放されて、同じコア上のほかの論理プロセッサ上で実行中の仮想マシンが、コアの実行リソース全体を使用できます。VMware スケジューラは、この停止時間を適切に捕捉し、コアのリソース全体を使用して実行中の仮想マシンを、半分のコア上で実行されている仮想マシンよりも多くチャージします。このプロセッサ管理方法により、標準の ESXi リソース割り当てルールにサーバが違反しないようにします。

ハイパースレッドを使用するホストでは、CPU アフィニティを有効にする前にリソース管理の必要性を検討してください。たとえば、優先順位の高い仮想マシンを CPU 0 にバインドしていて、別の優先順位の高い仮想マシンを CPU 1 にバインドしている場合、この2台の仮想マシンは同じ物理コアを共有する必要があります。この場合、これらの仮想マシンのリソース要求を満たせないことがあります。カスタム アフィニティ設定が、ハイパースレッドシステムにとって意味があることを確認してください。

ハイパースレッドの有効化

ハイパースレッドを有効にするには、最初に使用するシステムの BIOS 設定でハイパースレッドを有効にし、vSphere Client でオンにする必要があります。ハイパースレッドは、デフォルトで有効になっています。

使用している CPU がハイパースレッドをサポートしているかどうかを確認するには、システムのドキュメントを参照してください。

手順

- 1 システムがハイパースレッドテクノロジーをサポートしていることを確認します。
- 2 システムの BIOS でハイパースレッドを有効にします。

メーカーによって、このオプションは [論理プロセッサ] と呼ばれることもあれば、[ハイパースレディングの有効化] と呼ばれることもあります。

- 3 必ず、ESXi ホスト用にハイパースレッドをオンにしてください。
 - a vSphere Client でホストを選択し、[構成] タブをクリックします。
 - b [プロセッサ] を選択し、[プロパティ] をクリックします。
 - c ダイアログ ボックスでハイパースレッドの状態を確認して、ハイパースレッドをオフまたはオン（デフォルト）にすることができます。

ハイパースレッドが有効になりました。

仮想マシンのハイパースレッド共有オプションの設定

ハイパースレッド システム上の物理コアを仮想マシンの仮想 CPU がどのように共有できるかを指定できます。

2 つの仮想 CPU がコアの論理 CPU 上で同時に実行されている場合、それらの仮想 CPU はコアを共有します。これは、個々の仮想マシンについて設定できます。

手順

- 1 vSphere Client のインベントリ パネルで仮想マシンを右クリックし、[設定の編集] を選択します。
- 2 [リソース] タブをクリックして、[CPU の詳細] をクリックします。
- 3 [モード] ドロップダウン メニューから、この仮想マシンのハイパースレッド モードを選択します。

ハイパースレッド コア共有オプション

vSphere Client を使用して、仮想マシンにハイパースレッド コア共有モードを設定できます。

表 4-1. ハイパースレッド コア共有モード

オプション	説明
任意	ハイパースレッド システムのすべての仮想マシンに対するデフォルトです。この設定の仮想マシンの仮想 CPU は、この仮想マシンまたはほかの仮想マシンの仮想 CPU と、いつでも自由にコアを共有できます。
なし	仮想マシンの仮想 CPU は、互いに、またはほかの仮想マシンの仮想 CPU とコアを共有できません。つまり、そのコア上のほかの論理 CPU を停止状態にして、この仮想マシンからの各仮想 CPU が、常にコア全体を取得する必要があります。
内部	このオプションは、[なし] に似ています。この仮想マシンからの仮想 CPU は、ほかの仮想マシンからの仮想 CPU とコアを共有することはできません。同じ仮想マシンの別の仮想 CPU とは、コアを共有できます。 このオプションは、SMP 仮想マシンについてのみ選択できます。ユニプロセッサ仮想マシンに適用された場合、システムのオプションを [なし] に変更します。

これらのオプションは、公平性または CPU 時間の割り当てには影響しません。仮想マシンのハイパースレッド設定にかかわらず、CPU シェアに比例する CPU 時間を受け取り、CPU 予約と CPU 制限の値によって制約されます。

通常のワークロードの場合、カスタム ハイパースレッド設定は不要のはずです。これらのオプションは、ハイパースレッドと不適切に対話する異常なワークロードの場合に役に立ちます。たとえば、キャッシュ スラッシングの問題があるアプリケーションは、物理コアを共有するアプリケーションの速度を低下させる場合があります。アプリケーションを実行中の仮想マシンのハイパースレッド状態を [なし] または [内部] にして、ほかの仮想マシンから隔離できます。

仮想 CPU に、別の仮想 CPU とのコアの共有を許可しないハイパースレッドの制約がある場合、システムは、ほかの仮想 CPU がプロセッサ時間を消費する資格を与えられた場合にそのスケジュール設定を解除します。ハイパースレッドの制約がない場合は、同じコア上の両方の仮想 CPU をスケジュール設定できます。

(仮想マシンあたりの) コア数が制限されているシステムでは、問題がさらに悪化します。そのような場合、スケジュール設定が解除された仮想マシンを移行できるコアがない場合があります。結果的に、ハイパースレッドが [なし] または [内部] に設定された仮想マシンのパフォーマンスが、特にコア数の制限されたシステム上で、低下する可能性があります。

隔離

まれにですが、特定の環境では、ESXi ホストがアプリケーションと Pentium IV ハイパースレッド テクノロジーとの対話の不良を検出する場合があります（これは、Intel Xeon 5500 プロセッサ マイクロアーキテクチャに基づいたシステムには当てはまりません）。このような場合、ユーザーに対して透過的な隔離が必要になる場合があります。

たとえば、特定のタイプの自己修正コードが Pentium IV トレース キャッシュの正常な動作を妨害し、問題のあるコードとコアを共有しているアプリケーションの速度を大きく（最大 90 パーセント）低下させることがあります。このような場合、ESXi ホストは、このコードを実行している仮想 CPU を隔離して、仮想マシンを適宜、[なし] モードまたは [内部] モードにします。

CPU アフィニティの使用

各仮想マシンに CPU アフィニティ設定を指定することにより、仮想マシンの割り当てをマルチプロセッサ システム内の使用可能なプロセッサのサブセットに制限できます。この機能を使用することで、各仮想マシンを指定したアフィニティセットのプロセッサに割り当てることができます。

CPU アフィニティは、仮想マシンのプロセッサへの配置の制約を指定します。これは、仮想マシンの仮想マシン ホストへの配置の制約を指定する、仮想マシン間または仮想マシンとホスト間のアフィニティ ルールによって作成された関係とは異なります。

ここでは、CPU という用語は、ハイパースレッド システム上の論理プロセッサを指し、また、非ハイパースレッド システム上のコアのことを指します。

仮想マシンの CPU アフィニティ設定は、その仮想マシンに関連付けられているすべての仮想 CPU と、その仮想マシンに関連付けられているその他のすべてのスレッド（ワールドとも呼ばれる）に適用されます。これらの仮想マシン スレッドは、マウス、キーボード、画面、CD-ROM、およびその他のレガシー デバイスのエミュレーションに必要な処理を実行します。

表示を多用するワークロードなど、一部のケースでは、仮想 CPU とその他の仮想マシン スレッドの間で大量の通信が発生することがあります。仮想マシンのアフィニティ設定によって、これらのその他のスレッドを仮想マシンの仮想 CPU と同時にスケジューリングできない場合、パフォーマンスが低下する可能性があります。たとえば、ユニプロセッサ仮想マシンに 1 つの CPU がアフィニティ設定されている場合や、2 方向の SMP 仮想マシンに 2 つの CPU がアフィニティ設定されている場合などです。

最大のパフォーマンスを得るために、手動でアフィニティを設定する場合は、少なくとも 1 つの仮想マシン スレッドをその仮想 CPU と同時にスケジューリングできるように、アフィニティ設定に少なくとも 1 つの物理 CPU を追加して含めておくことをお勧めします。たとえば、ユニプロセッサ仮想マシンに少なくとも 2 つの CPU をアフィニティ設定したり、2 方向の SMP 仮想マシンに少なくとも 3 つの CPU をアフィニティ設定します。

特定のプロセッサへの仮想マシンの割り当て

CPU アフィニティを使用すると、特定のプロセッサに仮想マシンを割り当てることができます。これにより、仮想マシンの割り当てをマルチプロセッサ システム内で使用可能な特定のプロセッサだけに制限できます。

手順

- 1 vSphere Client のインベントリ パネルで仮想マシンを選択し、[設定の編集] を選択します。
- 2 [リソース] タブを選択して、[CPU の詳細] を選択します。
- 3 [次のプロセッサで実行します] ボタンをクリックします。
- 4 仮想マシンを実行するプロセッサを選択し、[OK] をクリックします。

CPU アフィニティでの潜在的な問題

CPU アフィニティを使用する前に、いくつかの問題について検討する必要がある場合があります。

CPU アフィニティでの潜在的な問題には、次のようなものがあります。

- マルチプロセッサ システムの場合、ESXi システムは自動ロード バランシングを実行します。仮想マシン アフィニティを手動で指定しないようにすると、プロセッサ間の負荷を均衡させるためのスケジューラの機能が向上します。
- アフィニティは、仮想マシンに指定された予約とシェアを満たす ESXi ホストの機能を妨害する場合があります。
- CPU アドミッション コントロールはアフィニティを考慮しないので、手動のアフィニティ設定を持つ仮想マシンは、予約を完全に受け取るとはかぎりません。

手動のアフィニティ設定がない仮想マシンは、手動のアフィニティ設定がある仮想マシンから不都合な影響を受けません。
- あるホストから別のホストへ仮想マシンを移動する場合、新しいホストの持つプロセッサの数が異なる場合がありますので、アフィニティが適用しなくなることがあります。
- NUMA スケジューラは、アフィニティを使用するまでに特定のプロセッサに割り当て済みの仮想マシンを管理できない場合があります。
- アフィニティは、マルチコア プロセッサまたはハイパースレッド プロセッサ上の仮想マシンをスケジュール設定するホストの機能に影響を及ぼすため、そういったプロセッサで共有されるリソースを最大限に活用できます。

ホスト電力管理ポリシー

ESXi では、パフォーマンスと消費電力との両立を調整するためにホスト ハードウェアが提供する、いくつかの電力管理機能を利用できます。電力管理ポリシーを選択することにより、これらの機能を ESXi でどのように使用するかを制御できます。

一般に、高パフォーマンスのポリシーを選択すると、絶対的なパフォーマンスは高まりますが、効率は低下します（1 ワットあたりのパフォーマンス）。低電力ポリシーを使用すると、絶対的なパフォーマンスは低くなりますが、効率は向上します。

ESXi では、5 つの電力管理ポリシーが提供されます。ホストが電力管理をサポートしない場合、または BIOS 設定でホスト オペレーティングシステムによる電力管理が許可されないことが指定されている場合、使用できるのは「サポート対象外」ポリシーのみです。

ホストのポリシーは、vSphere Client を使用して選択します。ポリシーを選択しない場合、ESXi ではデフォルトで [バランシング済み] が使用されます。

表 4-2. CPU 電力管理ポリシー

電力管理ポリシー	説明
サポート対象外	ホストは電力管理機能をサポートしません。または、電力管理が BIOS で有効になっていません。
高パフォーマンス	VMkernel は特定の電力管理機能を検出しますが、パワー キャッピング イベントまたは温度管理 イベント用に BIOS が要求しないかぎり、それらの機能を使用しません。
バランシング済み (デフォルト)	VMkernel は使用可能な電力管理機能を控えめに使用することで、パフォーマンスにほとんど影響を及ぼすことなくホストのエネルギー消費を低減します。
省電力	VMkernel は、パフォーマンスを低下させてもホストのエネルギー消費を軽減するために、使用可能な電力管理機能を積極的に使用します。
カスタム	VMkernel は、複数の詳細構成パラメータの値に基づいて電力管理ポリシーを適用します。これらのパラメータは、vSphere Client の [詳細設定] ダイアログ ボックスで設定できます。

CPU が低い周波数で実行されると、電圧も低くなるため、電力消費を削減できます。このタイプの電力管理は、通常、Dynamic Voltage and Frequency Scaling (DVFS) と呼ばれます。ESXi では、仮想マシンのパフォーマンスに影響を与えないように、CPU 周波数を調整します。

CPU がアイドル状態の際、ESXi はディープ停止状態 (C 状態) を利用できます。C 状態のディープ度が高いほど、CPU で使用される電力は少なくなりますが、CPU が稼働を再開するまでにかかる時間は長くなります。CPU がアイドル状態になると、ESXi は、アイドル状態の期間を予測し、適切な C 状態を選択するアルゴリズムを適用します。ディープな C 状態を使用しない電力管理ポリシーでは、ESXi は、アイドル状態の CPU に対し、ディープ度が最も低い停止状態 (C1) のみを使用します。

CPU 電力管理ポリシーの選択

ホストの CPU 電力管理ポリシーは、vSphere Client を使用して設定します。

開始する前に

ホストシステムの BIOS 設定で、オペレーティングシステムによる電力管理の制御が許可されていることを確認してください ([OS 制御] など)。

注意 システムにはプロセッサクロック制御 (PCC) テクノロジーを備えたものがあり、このテクノロジーによって、ホストの BIOS 設定で OS 制御モードが指定されない場合でも、ESXi によるホストシステムの電力管理が可能になります。このテクノロジーでは、ESXi は P 状態を直接的には管理しません。代わりに、ホストは BIOS と連動し、プロセッサのクロック速度を決定します。このテクノロジーをサポートしている HP システムには、Cooperative Power Management と呼ばれる BIOS 設定があり、これはデフォルトで有効になります。

ホストハードウェアでオペレーティングシステムによる電力管理が許可されていない場合、使用できるのは "サポート対象外" ポリシーのみです。(一部のシステムでは、"高パフォーマンス" ポリシーのみ使用できます)。

手順

- 1 vSphere Client のインベントリ パネルでホストを選択し、[構成] タブをクリックします。
- 2 ハードウェアの下の [電力管理] を選択し、[プロパティ] を選択します。
- 3 ホストの電力管理ポリシーを選択し、[OK] をクリックします。

この選択されたポリシーはホストの構成に保存され、起動時に再度使用されます。このポリシーはいつでも変更可能で、サーバを再起動する必要はありません。

ホストの電力管理のためのカスタム ポリシー パラメータの構成

ホストの電力管理のためにカスタム ポリシーを使用する場合、ESXi では、電力管理ポリシーはいくつかの詳細設定パラメータの値に基づきます。

開始する前に

電力管理ポリシーに [カスタム] を選択します。これは「[CPU 電力管理ポリシーの選択 \(P. 23\)](#)」に記載されているとおりです。

手順

- 1 vSphere Client インベントリでホストを選択し、[構成] タブをクリックします。
- 2 ソフトウェアの下で [詳細設定] を選択します。
- 3 左側のペインの [電源] をクリックします。

- 4 右側のペインで、カスタム ポリシーに影響する電力管理パラメータを編集できます。

カスタム ポリシーに影響する電力管理パラメータには、[(カスタム ポリシー項目)] で始まる説明が付けられています。その他すべての電力パラメータが、すべての電力管理ポリシーに影響します。

注意 デフォルトの電力管理パラメータの値は、balancing 済みポリシーとマッチします。

パラメータ	説明
Power.UsePStates	プロセッサがビジー状態のときに電力を抑えるには、ACPI P-states を使用します。
Power.MaxCpuLoad	指定したリアルタイムの割合を下回るビジー状態に CPU がある場合にのみ、CPU のパワーオンを抑えるには、P 状態を使用します。
Power.MinFreqPct	指定したフル CPU 速度の割合を下回る速度の P 状態を使用しません。
Power.UseStallCtr	キャッシュのミスなどのイベントを待機してプロセッサが頻繁に失速する場合に、ディープ度が高い P 状態を使用します。
Power.TimerHz	各 CPU がどの P 状態にあるべきかを ESXi が 1 秒間に再評価する回数を制御します。
Power.UseCStates	プロセッサがアイドル状態の場合に、ディープ度が高い ACPI C 状態 (C2 以下) を使用します。
Power.CStateMaxLatency	待ち時間がこの値を超えた場合に、C 状態を使用しません。
Power.CStateResidencyCoef	CPU がアイドル状態になったときに、待ち時間をこの値で乗算した値が、CPU がアイドル状態のままになる時間についてのホストの予測を下回る、ディープ度が最高の C 状態を選択します。より大きな値を指定するほど、ESXi は高いディープ度の C 状態の使用を抑え、より低い値を指定するほど、より積極的に使用します。
Power.CStatePredictionCoef	アイドル状態になる CPU が、その後アイドル状態のままになる時間の予測についての ESXi アルゴリズムのパラメータです。この値の変更は推奨されません。
Power.PerfBias	パフォーマンス電力バイアス ヒント (内部のみ) です。Intel プロセッサ上の MSR を、Intel 推奨値に設定します。Intel の推奨値は、パフォーマンスを高くする場合に 0、バランスをとる場合に 6、低電力の場合に 15 です。その他の値は定義されていません。

メモリ仮想化の基礎

メモリリソースを管理する前に、メモリリソースが ESXi によってどのように仮想化され、使用されるかを理解しておく必要があります。

VMkernel は、すべてのマシンメモリを管理します。VMkernel は、この管理対象マシンメモリの一部を自身専用にし、残りが、仮想マシン用に使用可能になります。仮想マシンは、マシンメモリを 2 つの目的で使用します。各仮想マシンは、自身のメモリを必要とし、仮想マシンモニタ (VMM) は、コードとデータ用にいくらかのメモリと動的オーバーヘッドメモリを必要とします。

仮想および物理メモリスペースは、ページと呼ばれるブロックに区別されます。物理メモリが一杯になった場合、物理メモリ内に存在しない仮想ページのデータは、ディスクに格納されます。プロセッサアーキテクチャに応じて、ページは通常 4 KB または 2 MB です。「[メモリの詳細属性 \(P. 99\)](#)」を参照してください。

この章では次のトピックについて説明します。

- [仮想マシンのメモリ \(P. 25\)](#)
- [メモリのオーバーコミット \(P. 26\)](#)
- [メモリの共有 \(P. 26\)](#)
- [ソフトウェアベースのメモリ仮想化 \(P. 26\)](#)
- [ハードウェアアシストによるメモリ仮想化 \(P. 27\)](#)

仮想マシンのメモリ

各仮想マシンは、構成されたサイズに基づいてメモリを消費し、加えて仮想化のためのオーバーヘッドメモリを消費します。

構成済みサイズは、仮想マシンの仮想化レイヤーで管理される構造です。これはゲスト OS に提供されるメモリの合計ですが、後述のリソース設定 (シェア、予約、制限) に依存し、仮想マシンに割り当てられる物理 RAM の量とは異なります。

たとえば、構成済みサイズが 1GB の仮想マシンについて考えます。ゲスト OS は起動するときに、物理メモリが 1GB の専用マシンで実行していると考えています。この仮想マシンに割り当てられる物理ホストメモリの実際の容量は、ESXi ホストのメモリリソース設定およびメモリの競合に依存します。場合によっては、仮想マシンに 1GB が完全に割り当てられません。別の場合には、より小さな割り当てを受けることがあります。実際の割り当てにかかわらず、ゲスト OS は、物理メモリが 1GB の専用マシンで実行している場合と同じ動作を続けます。

シェア 予約以外に指定可能な場合、ここで仮想マシンの相対的な優先順位を指定します。

予約 メモリがオーバーコミットされる場合でも、ホストが仮想マシン用に予約する物理メモリ容量の、保証された下限です。過大なページングがなく効率的に動作するのに十分なメモリを仮想マシンが確実に利用できるよう、予約のレベルを設定します。

仮想マシンは、予約分全体にアクセスしたあと、たとえアイドル状態になった場合でも、その量のメモリを保持することができ、そのメモリは解放されません。たとえば、一部のゲスト OS（Linux など）は、起動直後に構成済みメモリの全部にはアクセスできない場合があります。仮想マシンがその予約分全体にアクセスするまで、VMkernel はその予約の任意の未使用部分をほかの仮想マシンに割り当てることができます。しかし、ゲストのワークロードが増加し、ゲストの予約分全体が消費された場合、ゲストはそのメモリを保持できません。

制限

ホストが仮想マシンに割り当てることができる物理メモリ容量の上限です。仮想マシンのメモリ割り当ては、構成済みサイズでも暗黙的に制限されます。

オーバーヘッドメモリには、仮想マシンのフレームバッファと各種の仮想化データ構造のために予約されたスペースが含まれます。

メモリのオーバーコミット

システムは実行中の仮想マシンごとに、仮想マシンの予約（ある場合）と、仮想マシンの仮想化オーバーヘッド用の物理メモリを予約します。

ESXi ホストのメモリ管理手法のおかげで、仮想マシンは、物理マシン（ホスト）で利用可能なメモリより多いメモリを使用できます。たとえば、2GB のメモリを持つホストがある場合に、それぞれ 1GB のメモリを持つ 4 台の仮想マシンを実行できます。この場合、メモリがオーバーコミットされます。

オーバーコミットが有用なのは、一般に、仮想マシンには負荷の軽いものと負荷の重いものがあり、相対的な動作レベルが時間とともに変動するためです。

メモリの使用率を向上させるために、ESXi ホストは、アイドル状態の仮想マシンから、より多くのメモリを必要とする仮想マシンにメモリを移動します。予約またはシェアのパラメータを使用して、重要な仮想マシンにメモリを優先的に割り当てます。このメモリは、使用されていない場合にほかの仮想マシンで利用できます。

また、[\[メモリ圧縮\(P.36\)\]](#) で説明するように、メモリのオーバーコミット時に仮想マシンのパフォーマンスを向上させるため、ESXi ホストではメモリ圧縮がデフォルトで有効になっています。

メモリの共有

仮想マシンの間でメモリを共有する機会を提供するワークロードは、いくつもあります。

たとえば、いくつかの仮想マシンが、同じゲスト OS のインスタンスを実行していたり、同じアプリケーションまたはコンポーネントをロードしていたり、共通のデータを含んでいたりとすることがあります。ESXi システムは、専用のページ共有技法を使用して、メモリ ページの冗長コピーを安全に削除します。

メモリの共有により、複数の仮想マシンからなるワークロードが消費するメモリが、物理マシン上で実行する場合の消費量よりも少なくなることがよくあります。結果的に、システムは、より高いレベルのオーバーコミットを効率的にサポートできます。

メモリの共有によって節約されるメモリの量は、ワークロードの特性に依存します。ほとんど同一な多数の仮想マシンのワークロードでは、30 パーセントを超えるメモリが解放される場合がある一方、より多様なワークロードでは、メモリの 5 パーセント未満しか節約されないことがあります。

ソフトウェアベースのメモリ仮想化

ESXi は、特別なレベルのアドレス変換を追加することによってゲスト物理メモリを仮想化します。

- 各仮想マシンの VMM は、ゲスト OS の物理メモリ ページから、基盤となるマシン上の物理メモリ ページへのマッピングを保持します（VMware は、基盤となるホスト物理ページを「マシン」ページとして参照し、ゲスト OS の物理ページを「物理」ページとして参照します）。

各仮想マシンは、ゼロベースの連続したアドレス可能物理メモリスペースを参照します。各仮想マシンによって使用される、サーバの基盤となるマシンメモリは、必ずしも連続しません。

- VMM は、プロセッサ上の実際の MMU（メモリ管理ユニット）が仮想マシンによって直接更新されないように、ゲスト OS のメモリ管理構造を操作する仮想マシン命令を妨害します。
- ESXi ホストは、(VMM によって保持される) 物理からマシンへのページ マッピングを使用して最新に保たれるシャドウ ページ テーブルに、仮想とマシンとのページ マッピングを保持します。
- シャドウ ページ テーブルがプロセッサのページング ハードウェアによって直接使用されます。

このアドレス変換方法により、シャドウ ページ テーブルが設定されたあと、アドレス変換オーバーヘッドが追加されることなく、仮想マシン内での通常のメモリ アクセスが実行可能になります。プロセッサ上の TLB（変換索引バッファ）は、シャドウ ページ テーブルから読み取られた仮想とマシンとの直接マッピングをキャッシュするので、VMM によってオーバーヘッドが追加されることなく、メモリにアクセスできます。

パフォーマンスについての考慮事項

2つのページテーブルを使用した場合、次のようなパフォーマンスへの影響があります。

- 通常のゲストメモリアクセスの場合、オーバーヘッドはありません。
- 仮想マシン内のメモリをマッピングする追加の時間が必要です。これは、次のことを意味する場合があります。
 - 仮想マシンオペレーティングシステムが、仮想アドレスと物理アドレスとのマッピングを設定または更新している。
 - 仮想マシンオペレーティングシステムが、アドレススペース間で切り替えを行なっている（コンテキストスイッチ）。
- CPU の仮想化と同様に、メモリの仮想化オーバーヘッドはワークロードに依存します。

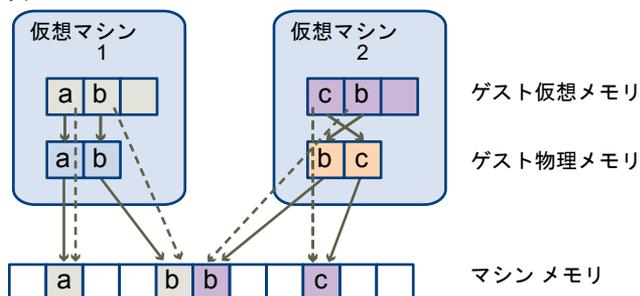
ハードウェアアシストによるメモリ仮想化

AMD SVM-V や Intel Xeon 5500 シリーズなどの一部の CPU は、ページテーブルの2つのレイヤーを使用することで、ハードウェアからメモリ仮想化をサポートする機能を提供しています。

ページテーブルの1番目のレイヤーには、ゲストの仮想メモリから物理メモリへの変換が格納されています。ページテーブルの2番目のレイヤーには、ゲストの物理メモリからマシンへの変換が格納されています。TLB（変換索引バッファ）は、プロセッサの MMU（メモリ管理ユニット）ハードウェアが維持する変換のキャッシュです。TLB ミスはこのキャッシュのミスであり、ハードウェアはメモリに（おそらく複数回）アクセスして必要な変換を検索する必要があります。TLB でゲストの仮想アドレスが参照できない場合、ハードウェアが両方のページテーブルを検索し、ゲストの仮想アドレスをホストの物理アドレスに変換します。

図は、ESXi でのメモリ仮想化の実装を示しています。

図 5-1. ESXi のメモリ マッピング



- ボックスはページを表し、矢印はそれぞれメモリ マッピングを表します。
- ゲスト仮想メモリからゲスト物理メモリへの矢印は、ゲスト OS 内のページテーブルが保持するマッピングを表します（仮想メモリから x86 アーキテクチャ プロセッサのリニア メモリへのマッピングは示していません）。
- ゲスト物理メモリからマシンメモリへの矢印は、VMM が保持するマッピングを表します。

- 点線の矢印は、ゲストの仮想メモリから、シャドウ ページ テーブル（これも VMM が保持）のマシン メモリへのマッピングを表します。仮想マシンを実行している基盤となるプロセッサは、シャドウ ページ テーブルのマッピングを使用します。

仮想化によって導入される特別なレベルのメモリ マッピングのため、ESXi は、すべての仮想マシン間でメモリを効率的に管理できます。仮想マシンの物理メモリの一部は、共有ページ、またはマッピング解除されるかスワップアウトされるページにマッピングされる場合があります。

ホストは仮想メモリ管理を行います。ゲスト OS が認識したり、ゲスト OS のメモリ管理サブシステムの干渉を受けたりすることはありません。

パフォーマンスについての考慮事項

ハードウェア アシストを使用すると、ソフトウェア メモリ仮想化のオーバーヘッドをなくすことができます。特に、ハードウェア アシストによって、シャドウ ページ テーブルをゲスト ページ テーブルと同期させておくために必要なオーバーヘッドがなくなります。ただし、ハードウェア アシストを使用すると、TLB ミスの遅延が大幅に増えます。結果として、ハードウェア アシストを使用することによってワークロードにメリットがあるかどうかは、ソフトウェアによるメモリ仮想化を使用した場合に発生するメモリ仮想化のオーバーヘッドに大きく依存します。ワークロードに含まれるページ テーブル アクティビティ（プロセス作成、メモリのマッピング、コンテキスト スイッチなど）が少ない場合、ソフトウェアの仮想化では、大きなオーバーヘッドは発生しません。逆に、ワークロードに大量のページ テーブル アクティビティが含まれている場合は、ハードウェア アシストを使用すると有益です。

メモリ リソースの管理

vSphere Client を使用して、メモリ割り当ての設定に関する情報を表示し、それらの設定を変更できます。メモリ リソースを効果的に管理するには、メモリ オーバーヘッド、アイドル メモリ税、および ESXi ホストがメモリをどのように解放するかについて、よく理解しておく必要もあります。

メモリ リソースを管理する場合、メモリ割り当てを指定できます。メモリ割り当てをカスタマイズしない場合、ESXi ホストは、ほとんどの状況で良好に機能するデフォルトを使用します。

メモリ割り当てを指定する方法は、いくつかあります。

- vSphere Client 通じて使用可能な属性と特殊な機能を使用します。vSphere Client のユーザー インターフェースを使用して ESXi ホストまたは vCenter Server システムに接続できます。
- 詳細設定を使用します。
- スクリプト化したメモリ割り当てに vSphere SDK を使用します。

この章では次のトピックについて説明します。

- [メモリ オーバーヘッドについて \(P. 29\)](#)
- [ESXi ホストによるメモリ割り当ての方法 \(P. 30\)](#)
- [アイドル仮想マシンのメモリ税 \(P. 31\)](#)
- [メモリの解放 \(P. 31\)](#)
- [スワップ ファイルの使用 \(P. 32\)](#)
- [ホスト キャッシュへのスワップ \(P. 34\)](#)
- [仮想マシン間でのメモリの共有 \(P. 35\)](#)
- [メモリ圧縮 \(P. 36\)](#)
- [メモリ使用量の測定とタイプの識別 \(P. 37\)](#)
- [メモリの信頼性 \(P. 38\)](#)

メモリ オーバーヘッドについて

メモリ リソースの仮想化には、関連するいくつかのオーバーヘッドがあります。

ESXi 仮想マシンは、2 種類のメモリ オーバーヘッドを被ることがあります。

- 仮想マシン内のメモリにアクセスするための追加の時間。
- ESXi ホストが自身のコードとデータ構造のために必要とする、各仮想マシンに割り当てられたメモリを超える余分のスペース。

ESXi のメモリ仮想化では、メモリ アクセスへの時間のオーバーヘッドはほとんど追加されません。プロセッサのページングハードウェアがページテーブル（ソフトウェアベースのアプローチ用のシャドウページテーブルまたはハードウェアアシストによるアプローチ用のネストしたページテーブル）を直接使用するので、仮想マシン内のほとんどのメモリアクセスは、アドレス変換のオーバーヘッドをせずに実行できます。

メモリ スペース オーバーヘッドには、2 つの構成要素があります。

- VMkernel が必要とする、システム全体の固定されたオーバーヘッド。
- 各仮想マシンの追加のオーバーヘッド

オーバーヘッド メモリには、仮想マシンのフレーム バッファと、シャドウ ページ テーブルなどの各種の仮想化データ構造のために予約されたスペースが含まれます。オーバーヘッド メモリは、仮想 CPU の数とゲスト OS 用に構成されたメモリに依存します。

ESXi は、メモリ共有などの最適化も提供して、基盤となるサーバ上で使用される物理メモリの量を減らします。これらの最適化により、オーバーヘッドによって占められるよりも多くのメモリを節約できます。

仮想マシン上のオーバーヘッド メモリ

仮想マシンをパワーオンするには、特定の量の使用可能なオーバーヘッド メモリが必要です。このオーバーヘッドの量を認識しておく必要があります。

次の表には、仮想マシンをパワーオンするために必要なオーバーヘッド メモリの量がリストされています。仮想マシンが実行された後、マシンで使用されるオーバーヘッド メモリ量はこの表にリストされている量と異なる場合があります。このサンプル値は VMX スワップおよびハードウェア MMU が有効になっている仮想マシンで収集されました (VMX スワップは、デフォルトで有効になっています)。

注意 表にリストされている値はオーバーヘッド メモリのサンプル値であり、考えられるすべての構成についての情報の提供を意図したものではありません。ホスト上のライセンス付き CPU の数、ゲスト OS がサポートする CPU の数に基づいて、仮想マシンに最大 32 の仮想 CPU を割り当てるように構成できます。

表 6-1. 仮想マシン上のサンプルのオーバーヘッド メモリ

メモリ (MB)	1 VCPU	2 VCPU	4 VCPU	8 VCPU
256	20.29	24.28	32.23	48.16
1024	25.90	29.91	37.86	53.82
4096	48.64	52.72	60.67	76.78
16384	139.62	143.98	151.93	168.60

ESXi ホストによるメモリ割り当ての方法

ホストは、メモリがオーバーコミットされていないかぎり、**制限** パラメータによって指定されたメモリを各仮想マシンに割り当てます。ESXi ホストは、指定された物理メモリ サイズを上回るメモリを仮想マシンに割り当てることはしません。

たとえば、1GB の仮想マシンには、デフォルトの制限（制限なし）またはユーザー指定の制限（たとえば 2GB）がある場合があります。このどちらの場合も、ESXi ホストは、1GB（仮想マシン用に指定された物理メモリ サイズ）を上回るメモリは割り当てません。

メモリがオーバーコミットされる場合、各仮想マシンには、[予約] によって指定された量と [制限] によって指定された量の間のメモリを割り当てられます。仮想マシンの予約を超えて仮想マシンに与えられるメモリの量は、通常、現在のメモリ負荷とともに変動します。

ホストは、仮想マシンに割り当てられたシェアの数と最新の作業セット サイズの概算に基づいて、各仮想マシンへの割り当てを決定します。

- シェア：ESXi ホストは、修正された比例的なシェアのメモリ割り当てポリシーを使用します。メモリ シェアは、使用可能な物理メモリの一部を仮想マシンに割り当てます。

- 作業セットサイズ：ESXi ホストは、仮想マシン実行時間の連続的な期間にわたってメモリの動作を監視することによって、仮想マシンの作業セットを見積もります。見積もりは、いくつかの期間にわたって平滑化されます。これには、作業セットサイズの拡大にすばやく対応し、作業セットサイズの縮小にゆっくりと対応する技法が使用されます。

この方法により、アイドル メモリを解放した仮想マシンがメモリをより有効に使用し始めた場合、共有ベースの割り当てすべてを短時間のうちに確実に獲得できます。

作業セットのサイズを見積もるために、デフォルトで 60 秒間、メモリの動作が監視されます。このデフォルトを変更するには、`Mem.SamplePeriod` 詳細設定を調整します。「[ホストの詳細属性の設定 \(P. 99\)](#)」を参照してください。

VMX スワップ ファイル

仮想マシン実行可能 (VMX) スワップ ファイルを使用すると、VMX 処理のために予約されるホストのオーバーヘッドメモリの量を大幅に削減できます。

注意 VMX スワップ ファイルは、ホスト キャッシュ機能へのスワップや通常のホスト レベルのスワップ ファイルとは関係がありません。

ESXi は、さまざまな目的で仮想マシンごとにメモリを予約します。仮想マシン モニタ (VMM) や仮想デバイスなどの特定のコンポーネントに必要なメモリは、仮想マシンのパワーオン時にすべて予約されます。しかし、VMX 処理のために予約されるオーバーヘッドメモリの一部は、スワップすることができます。VMX スワップ機能によって、VMX メモリの予約が大幅に削減されます (たとえば、仮想マシンごとに 50MB 以上だったのが、仮想マシンごとに 10MB 程度など)。これにより、ホストのメモリが割り当てられすぎているときに、残りのメモリをスワップアウトすることが可能になり、仮想マシンごとのオーバーヘッドメモリの予約量を削減できます。

VMX スワップメモリは、仮想マシンのパワーオン時に空きディスク領域が十分にあれば、ホストで自動的に作成されます。

アイドル仮想マシンのメモリ税

現在割り当てられているメモリのすべてを仮想マシンが有効に使用していない場合、ESXi は、使用中のメモリに対してより、アイドル メモリに対してより多くのコストを課します。これは、仮想マシンがアイドル メモリを蓄えないようにするのに役立ちます。

アイドル メモリ税は、累進的に課税されます。仮想マシンのアクティブなメモリに対するアイドル メモリの割合が大きくなるにつれて、実効税率も増加します (以前のバージョンの ESXi は階層リソース プールをサポートしておらず、仮想マシンのすべてのアイドル メモリには均一に課税されていました)。

アイドル メモリの税率は、`Mem.IdleTax` オプションで変更できます。このオプションを `Mem.SamplePeriod` 詳細属性とともに使用して、仮想マシンのターゲット メモリ割り当ての決定方法を制御します。「[ホストの詳細属性の設定 \(P. 99\)](#)」を参照してください。

注意 ほとんどの場合、`Mem.IdleTax` の変更は必要なく、適切でもありません。

メモリの解放

ESXi ホストは、仮想マシンからメモリを解放できます。

ホストは、予約によって指定された量のメモリを、仮想マシンに直接割り当てます。予約を超えるメモリは、ホストの物理リソースを使用して割り当てられるか、物理リソースを使用できない場合は、バルーニングやスワップなどの特別な方法を使用して処理されます。ホストは、2つの方法を使用して、仮想マシンに割り当てられたメモリの量を動的に拡大または縮小します。

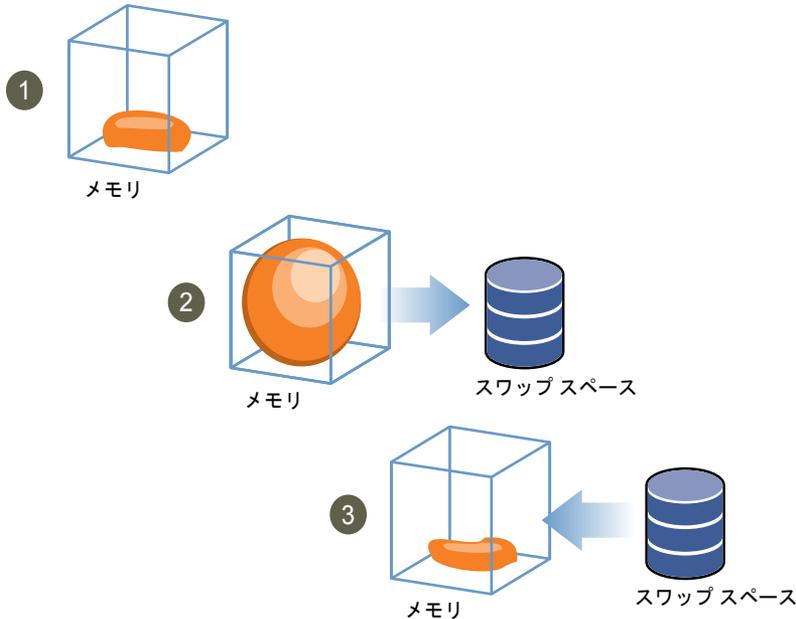
- ESXi システムは、仮想マシン内で実行中のゲスト OS にロードされたメモリ バルーン ドライバ (`vmmemctl`) を使用します。「[メモリ バルーン ドライバ \(P. 32\)](#)」を参照してください。
- ESXi システムは、ゲスト OS による関与なしに仮想マシンからサーバ スワップ ファイルにページングします。各仮想マシンには、独自のスワップ ファイルがあります。

メモリ バルーン ドライバ

メモリ バルーン ドライバ (`vmemctl`) は、サーバと協同して、ゲスト OS によって最も価値がないと見なされたページを解放します。

このドライバは、同様のメモリ制約下のネイティブシステムの動作と緊密に一致する、予測可能なパフォーマンスを提供する専用のバルーン技法を使用します。この技法は、ゲスト OS に対するメモリ プレッシャーを増減して、ゲスト独自のネイティブメモリ管理アルゴリズムをゲストに使用させます。メモリが十分ではない場合、ゲスト OS は、どのページを解放するかを決定し、必要に応じて、それらのページを自身の仮想ディスクにスワップします。

図 6-1. ゲスト OS のメモリ バルーン



注意 十分なスワップスペースを持つゲスト OS を構成する必要があります。ゲスト OS によっては、追加の制限事項があります。

必要な場合は、特定の仮想マシンについて `[sched.mem.maxmemctl]` パラメータを設定することによって、`vmemctl` が解放するメモリの量を制限できます。このオプションは、仮想マシンから解放できるメモリの最大量をメガバイト (MB) 単位で指定します。[「仮想マシンの詳細属性の設定 \(P. 102\)」](#) を参照してください。

スワップ ファイルの使用

スワップ ファイルの場所の指定、メモリがオーバーコミットされている場合のスワップスペースの予約、およびスワップ ファイルの削除が可能です。

ESXi ホストは、次のように `vmemctl` ドライバが使用できない場合、または反応しない場合、スワップを使用して仮想マシンからメモリを強制的に解放します。

- インストールされていない。
- 明示的に無効にされている。
- 実行されていない (たとえば、ゲスト OS の起動中)。
- 現在のシステム要求を十分に満たすほど迅速にメモリを解放することが、一時的にできない。
- 適切に機能しているが、最大バルーン サイズに達している。

標準の要求ページング技法は、仮想マシンにページが必要な場合、ページをスワップバックします。

スワップ ファイルの場所

デフォルトでは、スワップ ファイルは仮想マシンの構成ファイルと同じ場所に作成されます。

スワップ ファイルは、仮想マシンをパワーオンしたときに、ESXi ホストによって作成されます。スワップ ファイルを作成できない場合、仮想マシンをパワーオンできません。デフォルトを使用せずに、次のように操作することもできます。

- 仮想マシンごとの構成オプションを使用して、データストアを別の共有ストレージの場所に変更する。
- ホストのローカル スワップを使用して、データストアをホスト上にローカルで格納するように指定する。このように指定すると、ホスト レベルでスワップするため、SAN の容量を節約できます。ただし、これによって vSphere vMotion のパフォーマンスがわずかに低下する場合があります。その理由は、ソース ホスト上のローカル スワップ ファイルへスワップされたページは、ネットワークを介してターゲット ホストへ転送される必要があるからです。

DRS クラスタでのホストのローカル スワップの有効化

ホストのローカル スワップを使用すると、ホスト上にローカルに格納されているデータストアをスワップ ファイルの場所として指定できます。DRS クラスタで、ホストのローカル スワップを有効にすることができます。

手順

- 1 vSphere Client で、インベントリにあるクラスタを右クリックして、[設定の編集] を選択します。
- 2 表示されるクラスタの設定ダイアログ ボックスの左側のペインで、[スワップファイルの場所] をクリックします。
- 3 [ホストが指定するデータストアにスワップファイルを格納する] オプションを選択し、[OK] をクリックします。
- 4 vSphere Client インベントリで、クラスタ内のホストから 1 つを選択し、[構成] タブをクリックします。
- 5 ソフトウェアで、[仮想マシン スワップファイルの場所] を選択します。
- 6 使用するローカル データストアを選択して [OK] をクリックします。
- 7 クラスタ内の各ホストについて、[手順 4](#) から [手順 6](#) を繰り返します。

これで、ホストのローカル スワップが DRS クラスタで有効になりました。

スタンドアロン ホストでのホストのローカル スワップの有効化

ホストのローカル スワップを使用すると、ホスト上にローカルに格納されているデータストアをスワップ ファイルの場所として指定できます。スタンドアロン ホストで、ホストのローカル スワップを有効にすることができます。

手順

- 1 vSphere Client で、インベントリからホストを選択します。
- 2 [構成] タブをクリックします。
- 3 ソフトウェアで、[仮想マシン スワップファイルの場所] を選択します。
- 4 [スワップファイル データストアにスワップファイルを格納] を選択します。
- 5 リストからローカル データストアを選択し、[OK] をクリックします。

これで、ホストのローカル スワップがスタンドアロン ホストで有効になりました。

スワップ スペースとメモリのオーバーコミット

仮想マシンごとのスワップ ファイルで、予約されていない仮想マシン メモリ（これは予約されたメモリ サイズと構成済みメモリ サイズの差です）に対して、スワップ スペースを予約する必要があります。

このスワップ予約は、どのような状況下でも ESXi ホストが仮想マシン メモリを維持できるようにするのに必要です。実際には、ホスト レベルのスワップ スペースのごく一部分だけが使用されます。

パルニングによって生じるゲスト内スワップをサポートするために、ESXi でメモリをオーバーコミットする場合は、必ず、ゲスト OS にも十分なスワップスペースが存在することを確認してください。このゲストレベルのスワップスペースは、仮想マシンの構成済みメモリ サイズと仮想マシンの [予約] の差以上である必要があります。



注意 メモリがオーバーコミットされ、ゲスト OS に構成されたスワップスペースが不十分な場合、仮想マシン内のゲスト OS は起動に失敗します。

仮想マシンの障害を防ぐには、仮想マシン内のスワップスペースのサイズを拡大します。

- Windows ゲスト OS : Windows オペレーティングシステムでは、スワップスペースをページングファイルと呼びます。一部の Windows オペレーティングシステムは、十分な空きディスク領域がある場合、ページングファイルのサイズを自動的に拡大しようとします。

Microsoft Windows のドキュメントを参照するか、Windows のヘルプファイルで「ページングファイル」を検索します。仮想メモリ ページングファイルのサイズを変更する手順に従ってください。

- Linux ゲスト OS : Linux オペレーティングシステムでは、スワップスペースをスワップファイルと呼びます。スワップファイルの拡大については、次の Linux の man ページを参照してください。
 - `mkswap` : Linux スワップ領域を設定します。
 - `swapon` : ページングとスワップのためのデバイスとファイルを有効にします。

メモリが大量にあり、仮想ディスクが小さいゲスト OS (たとえば、8GB の RAM と 2GB の仮想ディスクを持つ仮想マシン) は、スワップスペースが不十分な場合の影響を受けやすくなります。

注意 スワップファイルは、シンプロビジョニングされた LUN 上に保存しないでください。シンプロビジョニングされた LUN 上に保存されたスワップファイルを使用する仮想マシンを実行した場合、スワップファイルの増大にエラーが起きる可能性があり、仮想マシンが終了する恐れがあります。

大きなスワップファイルを作成する場合 (100GB を超える場合など)、仮想マシンのパワーオンに要する時間が大幅に長くなる可能性があります。これを回避するには、大規模な仮想マシン用に予約を増やして設定します。

また、ホストのローカルスワップファイルを使用することで、コストの低いストレージにスワップファイルを配置することもできます。

スワップファイルの削除

ホストに障害が起きた場合、そのホストにスワップファイルを使用している実行中の仮想マシンがあるとスワップファイルは存在し続け、ディスクスペースを何ギガバイトも消費します。スワップファイルを削除すると、この問題を解消できます。

手順

- 1 障害のあったホスト上の仮想マシンを再起動します。
- 2 仮想マシンを停止します。

仮想マシンのスワップファイルが削除されます。

ホスト キャッシュへのスワップ

SSD (ソリッドステートドライブ) 上に作成されたデータストアは、ホスト キャッシュの領域の割り当てに使用できます。ホストは、ホスト キャッシュへのスワップ用に一定量の領域を確保します。

ホスト キャッシュは、ESXi が仮想マシン スワップファイルの書き込み戻しキャッシュとして使用する低待ち時間のディスク上のファイルから構成されます。このキャッシュは、このホスト上で実行されているすべての仮想マシンで共有されます。仮想マシン ページのホストレベルのスワップは、容量にかぎりがある可能性がある SSD 領域を有効活用します。

ホスト キャッシュへのスワップを使用することは、SSD が支援するデータストア上に通常のスワップ ファイルを配置することと同じではありません。ホスト キャッシュへのスワップを有効にした場合でも、ホストでは通常のスワップ ファイルを作成する必要があります。しかし、ホスト キャッシュへのスワップを使用した場合、ホストが通常のスワップ ファイルを配置するストレージの速度の重要性は低くなります。

ホスト キャッシュの構成ページを使用すると、ホストがホスト キャッシュのスワップに使用できるデータストアの容量を表示できます。ホスト キャッシュ構成ページのデータストアのリストには、SSD が支援するデータストアのみ表示されます。

ホスト キャッシュの構成

ホスト キャッシュに対して割り当て済み領域の割合を変更する、またはホスト キャッシュにスワップするホストの機能を無効にすることができます。

開始する前に

インベントリに SSD が支援するデータストアがある必要があります。

手順

- 1 vSphere Client で、インベントリからホストを選択します。
- 2 [構成] タブをクリックします。
- 3 ソフトウェアで、[ホスト キャッシュ構成] をクリックします。
- 4 リストでデータストアを選択し、[プロパティ] をクリックします。
- 5 ドライブ上のホスト キャッシュ割り当てのサイズを選択します。
- 6 ホストがデータストアごとにホスト キャッシュにスワップする機能を無効にするには、[ホスト キャッシュに領域を割り当て] チェック ボックスを選択解除します。
- 7 [OK] をクリックします。

仮想マシン間でのメモリの共有

多くの ESXi ワークロードでは、複数の仮想マシン間で（単一の仮想マシン内でも）メモリを共有する機会があります。

たとえば、いくつかの仮想マシンが、同じゲスト OS のインスタンスを実行していたり、同じアプリケーションまたはコンポーネントをロードしていたり、共通のデータを含んでいたりとすることがあります。このような場合、ホストは、専用の透過なページ共有技法を使用してメモリ ページの冗長コピーを安全に削除します。メモリの共有により、仮想マシンで実行しているワークロードが消費するメモリが、物理マシン上で実行する場合の消費量よりも少なくなることがよくあります。結果的に、より高いレベルのオーバーコミットを効率的にサポートできます。

Mem.ShareScanTime と **Mem.ShareScanGHZ** の詳細設定を使用すると、メモリ共有の機会を特定するためにシステムがメモリをスキャンする速度を制御できます。

sched.mem.pshare.enable オプション（デフォルトは [TRUE]）を [FALSE] に設定することで、個々の仮想マシンの共有を無効にすることもできます。[\[仮想マシンの詳細属性の設定 \(P. 102\)\]](#) を参照してください。

ESXi のメモリ共有は、時間の経過とともに共有の機会をスキャンするバックグラウンド動作として実行されます。節約されるメモリの量は、時間とともに変動します。ある程度一定のワークロードの場合、共有のすべての機会が利用されるまで、量は一般にゆっくりと増加します。

特定のワークロードについてのメモリ共有の有効性を判定するには、ワークロードを実行してみて、**resxtop** または **esxstop** を使用して実際の節約を観察します。その情報は、メモリ ページの対話モードの **PSHARE** フィールドにあります。

メモリ圧縮

ESXi はメモリ圧縮によるキャッシュ機能を備えているため、メモリのオーバーコミットメントの使用時に仮想マシンのパフォーマンスを向上させます。メモリ圧縮はデフォルトで有効になっています。ホストのメモリがオーバーコミット状態になると、ESXi は仮想ページを圧縮し、メモリに保存します。

圧縮メモリへのアクセスは、ディスクにスワップされたメモリへのアクセスよりも早いため、ESXi でメモリ圧縮を使用すると、パフォーマンスに大きな影響を与えることなくメモリをオーバーコミットできます。仮想ページでスワップが必要な場合、ESXi はまずページを圧縮しようとします。2KB 以下に圧縮できたページは仮想マシンの圧縮用キャッシュに保存されるため、ホストの容量が増加します。

vSphere Client の詳細設定ダイアログ ボックスで、圧縮キャッシュの最大サイズの設定、およびメモリ圧縮の無効化を行うことができます。

メモリ圧縮キャッシュの有効化または無効化

メモリ圧縮はデフォルトで有効になっています。vSphere Client の詳細設定ダイアログ ボックスを使用して、ホストのメモリ圧縮を有効または無効にできます。

手順

- 1 vSphere Client で、インベントリからホストを選択します。
- 2 [構成] タブをクリックします。
- 3 ソフトウェアの下で [詳細設定] を選択します。
- 4 左側のペインで [メモリ] を選択し、Mem.MemZipEnable を探します。
- 5 メモリ圧縮キャッシュを有効にするには 1 を、無効にするには 0 を入力します。
- 6 [OK] をクリックします。

メモリ圧縮キャッシュの最大サイズの設定

ホストの仮想マシン用に、メモリ圧縮キャッシュの最大サイズを設定できます。

圧縮キャッシュのサイズは、仮想マシンのメモリ サイズのパーセンテージで設定します。たとえば、仮想マシンのメモリ サイズが 1,000 MB のときに値 20 を入力した場合、ESXi では、仮想マシンの圧縮されたページを格納するのに、最大で 200 MB のホスト メモリを使用できます。

圧縮キャッシュのサイズを設定しない場合、ESXi ではデフォルト値の 10 パーセントを使用します。

手順

- 1 vSphere Client で、インベントリからホストを選択します。
- 2 [構成] タブをクリックします。
- 3 ソフトウェアの下で [詳細設定] を選択します。
- 4 左側のペインで [メモリ] を選択し、Mem.MemZipMaxPct を探します。
この属性の値によって、仮想マシンの圧縮キャッシュの最大サイズが決まります。
- 5 圧縮キャッシュの最大サイズを入力します。
この値は仮想マシンのサイズのパーセンテージで、5 ~ 100 パーセントである必要があります。
- 6 [OK] をクリックします。

メモリ使用量の測定とタイプの識別

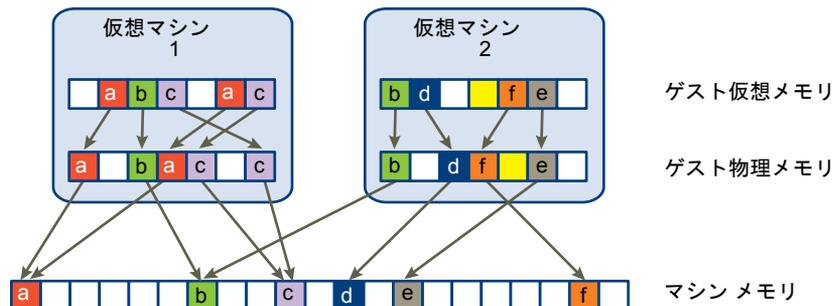
vSphere Client の [パフォーマンス] タブに、メモリ使用量の分析に使用できる、いくつかのメトリックが表示されます。

これらのメモリ メトリックには、ゲストの物理メモリを測定するものと、マシン メモリを測定するものがあります。たとえば、パフォーマンス メトリックを使用して検査できるメモリ使用量の 2 つのタイプは、ゲスト物理メモリとマシンメモリです。ゲスト物理メモリを測定するには、与えられたメモリ メトリック（仮想マシンの場合）または共有メモリ（ホストの場合）を使用します。しかし、マシンメモリを測定するには、消費されたメモリ（仮想マシンの場合）または一般共有メモリ（ホストの場合）を使用します。これらのメトリックが何を測定し、これらのメトリックをどのように解釈すべきかを知るためには、これらのタイプのメモリ使用量について、概念上の違いを理解することが重要です。

VMkernel はゲスト物理メモリをマシンメモリにマッピングしますが、必ずしも常に 1 対 1 でマッピングされるわけではありません。ゲスト物理メモリの複数の領域が、マシンメモリの同じ領域にマッピングされることもあれば（メモリ共有の場合）、ゲスト物理メモリの特定の領域がマシンメモリにマッピングされないこともあります（VMkernel がゲスト物理メモリをスワップアウトするかパルーニングする場合）。そうした状況では、個々の仮想マシンやホストについてのゲスト物理メモリの使用量およびマシンメモリの使用量の計算が異なります。

次図の例を考えてみましょう。これは、1 つのホストで実行中の 2 つの仮想マシンを示しています。各ブロックは 4KB のメモリを表しており、それぞれの色または文字は、ブロック上の異なるデータセットを表しています。

図 6-2. メモリ使用量の例



仮想マシンのパフォーマンス メトリックは、次のようにして判別できます。

- 仮想マシン 1 の与えられたメモリ（マシンメモリにマッピングされたゲスト物理メモリの量）は、マシンメモリへ向かう矢印を持つ、仮想マシン 1 のゲスト物理メモリのブロック数をカウントし、それに 4KB を乗算します。矢印を持つブロックは 5 つあるので、与えられたメモリは 20KB になります。
- [消費されたメモリ] は、仮想マシンに割り当てられたマシンメモリの量で、共有メモリから差し引かれる量を示しています。最初に、仮想マシン 1 のゲスト物理メモリからの矢印を持つ、マシンメモリ内のブロックの数をカウントします。そのようなブロックは 3 つありますが、1 ブロックは仮想マシン 2 と共有されています。したがって、2 つの完全なブロックと、3 番目のブロックの半分をカウントし、それに 4KB を乗算し、合計 10KB が消費されたメモリになります。

これら 2 つのメトリックの重要な違いは、[与えられたメモリ] がゲスト物理メモリ レベルで矢印を持つブロックの数をカウントし、[消費されたメモリ] はマシンメモリ レベルで矢印を持つブロックの数をカウントすることです。ブロックの数は、メモリ共有のために 2 つのレベル間で異なり、したがって、[与えられたメモリ] と [消費されたメモリ] に違いが出ます。これは問題にはならず、メモリが共有またはその他の解放の技法によって節約されていることを示しています。

ホストの共有メモリおよび一般共有メモリの判別でも、同様な結果が得られます。

- ホストの [共有メモリ] は、各仮想マシンの [共有メモリ] の合計です。これを計算するには、各仮想マシンのゲスト物理メモリに注目し、マシンメモリ ブロックを指す矢印を持ち、それ自体も複数の矢印で指されているブロックの数を数えます。例では、そのようなブロックが 6 つあるので、このホストの共有メモリは 24KB です。
- [一般共有メモリ] は、複数の仮想マシンによって共有されているマシンメモリの量です。これを判別するには、マシンメモリに注目し、複数の矢印で指されているブロックの数をカウントします。そのようなブロックは 3 つあるので、一般共有メモリは 12KB です。

[共有メモリ] は、ゲスト物理メモリに関するもので、矢印の起点に注目します。しかし、[一般共有メモリ] はマシンメモリに関するもので、矢印の宛先に注目します。

ゲスト物理メモリとマシンメモリを測定するメモリメトリックは、矛盾して見える場合があります。これらのメトリックは、実際には仮想マシンのメモリ使用量の異なる局面を測定しています。これらのメトリックの違いを理解することにより、これらのメトリックをより上手に利用して、パフォーマンスの問題を診断できます。

メモリの信頼性

メモリの信頼性は、**error insolation** と呼ばれ、ESXi により、実際に障害が発生した場合と同様、障害が発生している可能性がある場合と判断した場合にメモリの一部の使用を停止できる機能です。

特定のアドレスで、訂正済みエラーの報告が十分集まったら、ESXi はこのアドレスの使用を中止し、訂正済みエラーが未訂正エラーになることを防ぎます。

メモリの信頼性は、RAM 内に訂正済みエラーや未訂正エラーがあっても、VMkernel の信頼性を高めます。また、システムが、エラーを含む可能性のあるメモリページを使用することを防ぎます。

エラー隔離通知の訂正

メモリの信頼性を使用して、VMkernel はエラー隔離通知を受信するページの使用を中止します。

ユーザーが vSphere Client でイベントを受信するのは、VMkernel が修正不可のメモリエラーから復旧したとき、修正可能なエラーが多数あるために VMkernel がシステムメモリのかなりの割合を廃棄するとき、または廃棄できないページが多数ある場合です。

手順

- 1 ホストとしての使用を中止します。
- 2 仮想マシンを移行します。
- 3 テストを実行します。

ストレージ I/O リソースの管理

vSphere ストレージ I/O のコントロールによって、クラスタ全体でストレージ I/O の優先順位付けが可能となり、ワークロードの統合が強化され、過剰なプロビジョニングに伴うコストの削減にもつながります。

ストレージ I/O コントロールでは、ストレージ I/O リソースを処理するためのシェアと制限の構造を拡張できます。I/O が輻輳状態の間に仮想マシンに割り当てられるストレージ I/O の量を制御できます。こうすると、重要度の低い仮想マシンよりも多くの I/O リソースを、重要度の高い仮想マシンに確実に割り当てることができます。

データストアでのストレージ I/O コントロールを有効にすると、ESXi は、そのデータストアとの通信時にホストが監視するデバイスの待ち時間の監視を始めます。デバイスの待ち時間がしきい値を上回ると、そのデータストアは輻輳状態であると見なされ、そのデータストアにアクセスする各仮想マシンに、それぞれのシェアに応じた I/O リソースが割り当てられます。仮想マシンごとにシェアを設定します。必要に基づいて、それぞれの数を調整できます。

ストレージ I/O コントロールは、2 つの手順で構成します。

- 1 データストアに対してストレージ I/O コントロールを有効にします。
- 2 各仮想マシンに許可される、ストレージ I/O シェアの数と 1 秒あたりの I/O 処理数 (IOPS) の上限を設定します。デフォルトでは、すべての仮想マシンのシェアは標準 (1000) に、IOPS は制限なしに設定されます。

注意 ストレージ I/O コントロールは、ストレージ DRS 対応のデータストア クラスタで、デフォルトで有効です。

この章では次のトピックについて説明します。

- [ストレージ I/O コントロールの要件 \(P. 39\)](#)
- [ストレージ I/O コントロールのリソースのシェアおよび制限 \(P. 40\)](#)
- [ストレージ I/O コントロールのリソースのシェアおよび制限の設定 \(P. 41\)](#)
- [ストレージ I/O コントロールの有効化 \(P. 41\)](#)
- [ストレージ I/O コントロールのしきい値の設定 \(P. 42\)](#)

ストレージ I/O コントロールの要件

ストレージ I/O コントロールには、要件と制限事項がいくつかあります。

- ストレージ I/O コントロールが有効になっているデータストアは、単一の vCenter Server システムで管理される必要があります。
- ストレージ I/O コントロールは、ファイバチャネルに接続されたストレージ、iSCSI に接続されたストレージ、および NFS に接続されたストレージでサポートされます。Raw デバイスマッピング (RDM) はサポートされていません。
- ストレージ I/O コントロールは、複数のエクステンを持つデータストアはサポートしません。

- ストレージの自動階層化機能を持つアレイにパッキングされているデータストアでストレージ I/O コントロールを使用する前に、『VMware ストレージ/SAN 互換性ガイド』を参照して、自動的に階層化されたストレージアレイがストレージ I/O コントロールと互換性があると確認されているかどうかを調べてください。

ストレージの自動階層化機能とは、ユーザーが設定したポリシーおよび現在の I/O パターンに基づいて、LUN/ボリューム、または LUN/ボリュームの一部を別のタイプのストレージメディア（SSD、FC、SAS、SATA）に移行する、アレイ（またはアレイグループ）の機能です。これらの自動的な移行/階層化機能を持たないアレイについては、特別な証明は不要です。これには、異なるタイプのストレージメディア間で手でデータを移行する機能を提供するアレイも含まれます。

ストレージ I/O コントロールのリソースのシェアおよび制限

各仮想マシンに許可されるストレージ I/O シェアの数と、1 秒あたりの I/O 処理数（IOPS）の上限を割り当てます。あるデータストアでストレージ I/O の輻輳状態が検出されると、そのデータストアにアクセスする仮想マシンの I/O ワークロードは、各仮想マシンが持つシェアの比率に基づいて調整されます。

ストレージ I/O シェアは、メモリおよび CPU リソースの割り当てに使用されるものと似ています。これについては、『[リソース割り当てのシェア \(P. 11\)](#)』に記述されています。これらのシェアは、ストレージ I/O リソースの配分に関する仮想マシンの相対的な重要性を表します。リソースの競合がある場合、シェア値が高い仮想マシンはストレージアレイへのアクセスを優先的に割り当てられるため、通常は、スループットの向上と待ち時間の削減につながります。

ストレージ I/O リソースを割り当てるときに、仮想マシンに許可する IOPS を制限できます。デフォルトでは、IOPS は制限なしです。仮想マシンに仮想ディスクが 2 つ以上ある場合、それらのすべての仮想ディスクに対して制限を設定する必要があります。設定しないと、仮想マシンに対してこの制限は実行されません。この場合、仮想マシンに対する制限は、すべての仮想ディスクの制限を合計した値になります。

リソースの制限を設定する利点と問題点については、『[リソース割り当て制限 \(P. 12\)](#)』を参照してください。仮想マシンに設定する制限を IOPS ではなく 1 秒あたりのメガバイト（Mbps）で考えている場合は、その仮想マシンの一般的な I/O サイズに基づいて Mbps を IOPS に変換できます。たとえば、64KB I/O のバックアップアプリケーションを 10Mbps に制限する場合は、160 IOPS に設定します。

ストレージ I/O コントロールのシェアおよび制限の表示

データストアで実行しているすべての仮想マシンのシェアと制限を表示できます。この情報を表示すると、実行しているクラスタに関係なく、そのデータストアにアクセスしているすべての仮想マシンの設定を比較できます。

手順

- 1 vSphere Client のインベントリでデータストアを選択します。
- 2 [仮想マシン] タブをクリックします。

このタブには、そのデータストアで実行している各仮想マシンとそれに関連するシェアの値、IOPS の制限、およびデータストアシェアのパーセンテージが表示されます。

ストレージ I/O コントロールのシェアの監視

データストアにアクセスしている仮想マシンの I/O ワークロードを、ストレージ I/O コントロールがそれらのシェアに基づいてどのように処理しているかを監視するには、データストアの [パフォーマンス] タブを使用します。

データストアのパフォーマンス チャートでは、次の情報を監視できます。

- データストアの平均待ち時間および IOPS の合計
- ホスト間の待ち時間
- ホスト間のキューの深さ
- ホスト間の読み取り/書き込み IOPS
- 仮想マシンのディスク間の読み取り/書き込み待ち時間
- 仮想マシンのディスク間の読み取り/書き込み IOPS

手順

- 1 vSphere Client インベントリでデータストアを選択し、[パフォーマンス] タブをクリックします。
- 2 [ビュー] ドロップダウン メニューから、[パフォーマンス] を選択します。
詳細については、『vSphere の監視とパフォーマンス』ドキュメントを参照してください。

ストレージ I/O コントロールのリソースのシェアおよび制限の設定

仮想マシンに相対的なシェアの量を割り当てることで、重要度に基づいて仮想マシンにストレージ I/O リソースを割り当てることができます。

仮想マシンのワークロードが非常に似ている場合を除き、シェアは、1 秒あたりの I/O 処理数や MB 数で割り当てを指定するとはかぎりません。高いシェアの仮想マシンは、低いシェアの仮想マシンと比べて、ストレージ デバイスやデータストアでより多くの同時 I/O 処理を保留状態に保つことができます。ワークロードに応じて、2 台の仮想マシンのスループットがそれぞれ異なることがあります。

手順

- 1 vSphere Client インベントリ内で仮想マシンを選択します。
 - 2 [サマリ] タブをクリックし、[設定の編集] をクリックします。
 - 3 [リソース] タブをクリックし、[ディスク] を選択します。
 - 4 リストから仮想ハード ディスクを選択します。
 - 5 [シェア] 列をクリックして、仮想マシンに割り当てる相対的なシェアの量（低、標準、高）を選択します。
[カスタム] を選択するとユーザー定義のシェア値を入力できます。
 - 6 [制限 - IOPS] 列をクリックし、仮想マシンに割り当てるストレージ リソースの上限を入力します。
IOPS は、1 秒あたりの I/O 処理数です。デフォルトでは、IOPS は制限なしです。低（500）、標準（1000）、または高（2000）を選択します。または、[カスタム] を選択してユーザーが定義したシェアの値を入力できます。
 - 7 [OK] をクリックします。
- シェアおよび制限はホストおよびクラスタの [リソース割り当て] タブに反映されます。

ストレージ I/O コントロールの有効化

ストレージ I/O コントロールを有効にすると、ESXi はデータストアの待ち時間を監視し、平均待ち時間がしきい値を超えると、データストアに送られる I/O 負荷を調整します。

手順

- 1 vSphere Client のインベントリでデータストアを選択し、[構成] タブをクリックします。
- 2 [プロパティ] をクリックします。
- 3 ストレージ I/O コントロールで、[有効] チェック ボックスを選択します。
- 4 [閉じる] をクリックします。

[データストア] タブのストレージ I/O コントロール列に、そのデータストアのストレージ I/O コントロールが有効化されていることが示されます。

ストレージ I/O コントロールのしきい値の設定

データストアの輻輳しきい値とは、データストアに許容された待ち時間の上限で、この上限に達すると、ストレージ I/O コントロールはシェア値に応じて仮想マシンのワークロードに重み付けをし始めます。

ほとんどの環境では、しきい値の設定を調整する必要はありません。



注意 ストレージ I/O コントロールは、アレイ上の同一のスピンドルを共有するデータストアすべてで輻輳のしきい値が同一である場合を除いて、正常に機能しません。

輻輳しきい値の設定を変更する場合は、次の考慮事項に従って値を設定します。

- 通常、値が高いほど、スループットの合計が高くなり、隔離レベルが低くなる。全体の平均待ち時間がしきい値より高い場合を除いて、スロットルは発生しない。
- スループットの方が待ち時間より重要な場合、値を低くし過ぎないでください。たとえば、ファイバ チャネル ディスクの場合、20 ミリ秒より低い値にすると、最大のディスク スループットが小さくなる可能性があります。50 ミリ秒を超える非常に高い値を設定すると、スループット全体で大きなメリットは得られず、待ち時間が極めて長くなることがあります。
- 値が低いほどデバイスの待ち時間は短くなり、仮想マシンの I/O のパフォーマンスの隔離レベルが高くなります。隔離レベルが高いということは、より頻繁にシェア コントロールが実行されることを意味します。デバイスの待ち時間が短くなると、最も高いシェア値を持つ仮想マシンの I/O の待ち時間が短くなります。一方、シェアの少ない仮想マシンでは I/O の待ち時間は長くなります。
- 待ち時間がより重要な場合、20 ミリ秒よりも低い、非常に低い値を設定すると、デバイスの待ち時間は短くなり、I/O 間の隔離レベルが向上するが、データストアのスループットの合計が減少する可能性がある。

開始する前に

ストレージ I/O コントロールが有効になっていることを確認します。

手順

- 1 vSphere Client のインベントリでデータストアを選択し、[構成] タブをクリックします。
- 2 [プロパティ] をクリックします。
- 3 ストレージ I/O コントロールで、[有効] チェック ボックスを選択します。
- 4 (オプション) [詳細] をクリックして、データストアの輻輳のしきい値を編集します。
値は、10 ミリ秒から 100 ミリ秒の間にする必要があります。
- 5 (オプション) [リセット] をクリックすると、輻輳しきい値の設定はデフォルト値 (30 ミリ秒) に戻ります。
- 6 [OK] をクリックし、[閉じる] をクリックします。

リソース プールの管理

リソース プールは、リソースを柔軟に管理するための論理上の抽象概念です。リソース プールは階層にグループ化することができ、使用可能な CPU リソースとメモリ リソースを階層状にパーティショニングするために使用できます。

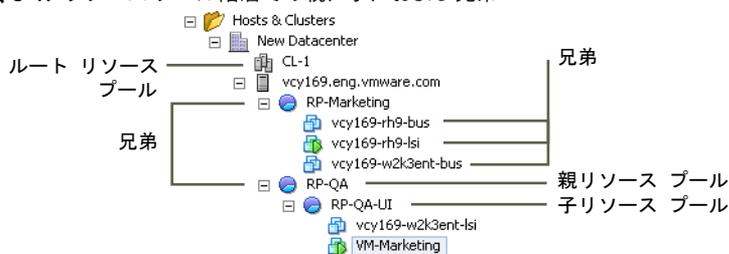
各スタンドアロン ホストと各 DRS クラスタには、(非表示の) ルート リソース プールがあり、そのホストまたはクラスタのリソースがグループ分けされています。ホスト (またはクラスタ) のリソースとルート リソース プールのリソースは常に同じであるため、ルート リソース プールは表示されません。

ユーザーは、ルート リソース プールの子リソース プール、またはユーザーが作成した任意の子リソース プールの子リソース プールを作成できます。それぞれの子リソース プールは、親のリソースの一部を保持し、これによって、連続した小さな単位のコンピュータ機能を表す子リソース プールの階層構造が形成されます。

リソース プールには、子リソース プール、仮想マシン、またはその両方を格納できます。共有リソースの階層を形成できます。より上位にあるリソース プールを親リソース プールと呼びます。同じレベルにあるリソース プールと仮想マシンを兄弟と呼びます。クラスタ自体は、ルート リソース プールを表します。子リソース プールを作成しない場合は、ルート リソース プールだけが存在します。

次の例の RP-QA は、RP-QA-UI の親リソース プールです。RP- マーケティングと RP-QA は兄弟です。RP- マーケティングのすぐ下にある 3 台の仮想マシンも兄弟です。

図 8-1. リソース プール階層での親、子、および兄弟



各リソース プールに対し、予約、制限、シェア、予約拡張の可否を指定します。リソース プールのリソースは、その子リソース プールと仮想マシンからも使用できます。

この章では次のトピックについて説明します。

- [リソース プールを使用する理由 \(P. 44\)](#)
- [リソース プールの作成 \(P. 45\)](#)
- [リソース プールの編集 \(P. 46\)](#)
- [リソース プールへの仮想マシンの追加 \(P. 46\)](#)
- [リソース プールからの仮想マシンの削除 \(P. 47\)](#)
- [リソース プールの削除 \(P. 47\)](#)
- [リソース プールのアドミッション コントロール \(P. 47\)](#)

リソース プールを使用する理由

リソース プールを使用すると、ホスト（またはクラスタ）のリソース管理を委任できます。リソース プールを使用して、クラスタ内のすべてのリソースを区分化する場合にメリットがあります。ホストまたはクラスタの直接の子として、複数のリソース プールを作成し、構成できます。その後、リソース プールの管理をほかの個人または組織に委任できます。

リソース プールを使用すると、次のメリットを得られる場合があります。

- 柔軟な階層型編成：必要に応じて、リソース プールを追加、削除、再編成し、リソース割り当てを変更します。
- プール間での分離とプール内での共有：最上位レベルの管理者は、リソース プールを部門レベルの管理者が使用できるようにすることができます。1つの部門のリソース プールの内部的な割り当ての変更が、ほかの無関係なリソース プールに不当な影響を及ぼすことはありません。
- アクセスコントロールと委任：最上位レベルの管理者が、リソース プールを部門レベルの管理者が使用できるようにした場合、部門レベルの管理者は、現在の設定（シェア、予約、制限）によってリソース プールに割り当てられるリソースの限度内で、すべての仮想マシンの作成と管理を実行できます。委任は、通常、権限の設定と結び付けて行われます。
- リソースとハードウェアとの分離：DRS が有効に設定されているクラスタを使用している場合、すべてのホストのリソースは、常にそのクラスタに割り当てられます。これは、リソースを提供する実際のホストと無関係に管理者がリソース管理を実行できることを意味します。たとえば3台の2GB ホストを2台の3GB ホストに置き換える場合、リソース割り当てを変更する必要はありません。

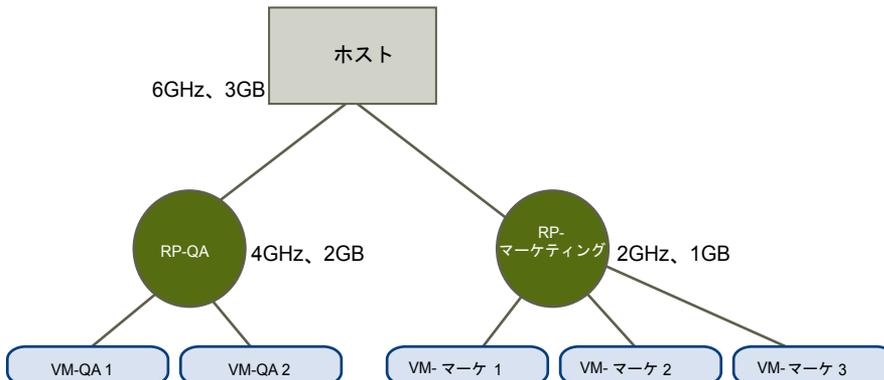
このように切り離して考えることで、管理者は個々のホストというより、コンピューティング能力全体に注意を払えるようになります。

- 多層サービスを実行する一連の仮想マシンの管理：リソース プール内で多層サービス用に仮想マシンをグループ化します。それぞれの仮想マシン上でリソースを設定する必要はありません。代わりに、それらの仮想マシンを含むリソース プール上で設定を変更することで、仮想マシンのセットに対するリソースの全体的な割り当てを制御できます。

たとえば、ホストにいくつかの仮想マシンがあるとします。マーケティング部門は3台の仮想マシンを使用し、QA 部門は2台の仮想マシンを使用します。QA 部門は、より多くのCPU とメモリを必要とするため、管理者は、グループごとに1つのリソース プールを作成します。QA 部門のユーザーは自動化されたテストを実行するため、管理者はQA 部門のプールの [CPU シェア] を [高] に設定し、マーケティング部門のプールを [標準] に設定します。CPU とメモリのリソースがより少ない2番目のリソース プールでも、マーケティングスタッフの負荷は軽いので十分です。QA 部門が割り当て分のすべてを使用していない場合、マーケティング部門はいつでも使用可能リソースを使用できます。

次の図の数値は、リソース プールに対する実際の割り当てを示しています。

図 8-2. リソース プールへのリソースの割り当て



リソース プールの作成

任意の ESXi ホスト、リソース プール、または DRS クラスタの子リソース プールを作成できます。

注意 ホストがクラスタに追加された場合、そのホストの子リソース プールは作成できません。クラスタが DRS に対して有効な場合、クラスタの子リソース プールを作成できます。

子リソース プールを作成する際、リソース プールの属性情報を入力するよう求められます。システムは、アドミッション コントロールを使用して、使用できないリソースが割り当てられないようにします。

開始する前に

vSphere Client は vCenter Server システムに接続されています。vSphere Client が直接、ホストに接続される場合は、リソース プールは作成できません。

手順

- 1 vSphere Client のインベントリで、リソース プールの親オブジェクト（ホスト、別のリソース プール、または DRS クラスタ）を選択します。
- 2 [ファイル] - [新規] - [リソース プール] を選択します。
- 3 リソース プールを識別するための名前を入力します。
- 4 CPU リソースおよびメモリ リソースの割り当て方法を指定します。

リソース プールの CPU リソースは、ホストがリソース プール用に予約する、保証された物理リソースです。通常、デフォルト値を受け入れ、ホストにリソース割り当てを任せます。

オプション	説明
シェア	親の合計リソースに対するこのリソース プールのシェアを指定します。兄弟のリソース プールは、予約と制限の範囲内で、相対的シェア値に従ってリソースを共有します。 <ul style="list-style-type: none"> ■ [低]、[標準]、または [高] を選択して、それぞれ 1:2:4 の比率でシェア値を指定します。 ■ 各仮想マシンに対して、比重に見合う特定のシェア値を指定するには、[カスタム] を選択します。
予約	リソース プールで確保されている CPU またはメモリの割り当てを指定します。デフォルトは 0 です。 <p>0 以外の予約は、親（ホストまたはリソース プール）の未予約のリソースから差し引かれます。リソースは、仮想マシンがリソース プールに関連付けられているかどうかに関係なく、予約済みとみなされます。</p>
拡張可能な予約	このチェック ボックスを選択すると（デフォルト）、アドミッション コントロール中に拡張可能な予約が考慮されます。 <p>このリソース プール内の仮想マシンをパワーオンすると仮想マシンの予約の合計がリソース プールの予約よりも大きくなる場合、リソース プールは親または先祖のリソースを使用できます。</p>
制限	リソース プールに割り当てる CPU またはメモリの上限を指定します。通常はデフォルト値（[制限なし]）を使用します。 <p>制限を指定するには、[制限なし] チェック ボックスを選択解除します。</p>

- 5 [OK] をクリックします。

リソース プールを作成したら、仮想マシンをリソース プールに追加できます。仮想マシンのシェアは、同じ親リソース プールを持つほかの仮想マシン（またはリソース プール）に対して相対的です。

例: リソース プールの作成

マーケティング部門と QA 部門の間で共有する必要がある、6GHz の CPU と 3GB のメモリを備えたホストがあると仮定します。また、リソースを均一に共有するのではなく、一方の部門 (QA) に高い優先順位を与えたいものとします。これは、各部門でリソース プールを作成し、[シェア] 属性を使用してリソース割り当ての優先順位を設定することで実現できます。

この例では、ESXi ホストを親リソースとするリソース プールの作成方法を説明します。

- 1 リソース プールの作成ダイアログ ボックスで、QA 部門のリソース プールの名前を入力します (例: RP-QA)。
- 2 RP-QA の CPU リソースとメモリ リソースについて、[シェア] を [高] に指定します。
- 3 2 番目のリソース プールの RP- マーケティングを作成します。
CPU とメモリの [シェア] は、[標準] のままにします。
- 4 [OK] をクリックします。

リソースの競合がある場合、RP-QA は 4GHz および 2GB のメモリを受け取り、RP- マーケティングは 2GHz および 1GB を受け取ります。競合がない場合は、どちらも前述の値を上回る割り当てを受け取ります。その後、これらのリソースはそれぞれのリソース プール内の仮想マシンで使用できます。

リソース プールの編集

リソース プールの作成後、CPU およびメモリのリソース設定を編集できます。

手順

- 1 vSphere Client で、インベントリにあるリソース プールを右クリックして、[設定の編集] を選択します。
- 2 [「リソース プールの作成 \(P. 45\)」](#) で説明されているように、選択したリソース プールのすべての属性を設定の編集ダイアログ ボックスで変更できます。
- 3 [OK] をクリックして、変更内容を保存します。

リソース プールへの仮想マシンの追加

仮想マシンを作成すると、作成プロセスの一部として、[新規仮想マシン] ウィザードでリソース プールの場所を指定できます。既存の仮想マシンをリソース プールに追加することもできます。

新しいリソース プールに移動した仮想マシンは、次のような状態になります。

- 仮想マシンの予約と制限は変更されません。
- 仮想マシンのシェアが [高]、[標準]、または [低] の場合、[シェア率] は、新規リソース プールで使用されているシェア数の合計に合わせて調整されます。
- 仮想マシンにカスタム シェアが割り当てられている場合は、そのシェア値が維持されます。

注意 シェアの割り当てはリソース プールに対して相対的なもので、仮想マシンをリソース プールに移動する場合は、新規リソース プール内の相対的な値と仮想マシンのシェアが一致するように、仮想マシンのシェアを手動で変更しなければならない場合があります。合計シェアに対して非常に大きい (または非常に小さい) 割合が仮想マシンに割り当てられると、警告が表示されます。

- [リソース割り当て] タブに表示される、リソース プールの予約済みおよび未予約の CPU とメモリのリソースに関する情報は、仮想マシンに関連付けられた予約 (ある場合) に合わせて変更されます。

注意 仮想マシンがパワーオフまたはサスペンド状態の場合、その仮想マシンを移動することはできませんが、リソース プール全体で使用可能なリソース (予約済みおよび未予約の CPU とメモリなど) は変更されません。

手順

- 1 vSphere Client で、インベントリから仮想マシンを選択します。
仮想マシンは、スタンドアロン ホスト、クラスタ、または別のリソース プールと関連付けることができます。
- 2 仮想マシン (複数可) をリソース プールにドラッグします。

仮想マシンがパワーオン状態で、ターゲットのリソース プールに仮想マシンの予約を確保するだけの十分な CPU またはメモリがない場合、仮想マシンの移動はアドミッション コントロールによって許可されないため、失敗します。エラー ダイアログ ボックスには、使用可能なリソースと要求されたリソースが表示されるので、調整することで問題が解決されるかどうかを判断できます。

リソース プールからの仮想マシンの削除

仮想マシンを別のリソース プールへ移動するか、仮想マシンを削除することにより、仮想マシンをリソース プールから削除できます。

仮想マシンをリソース プールから削除すると、リソース プールに関連付けられたシェアの総数が減り、残りの各シェアが、より多くのリソースを表すようになります。たとえば、シェアを [標準] に設定した 3 台の仮想マシンを含み、6GHz を割り当てられたプールがあるとします。仮想マシンが CPU バインドであったとすると、各仮想マシンは、均等に 2GHz を割り当てられます。仮想マシンの 1 台を別のリソース プールに移動した場合、残りの 2 台の仮想マシンは、それぞれ均等に 3GHz を割り当てられます。

手順

- 1 vSphere Client で、インベントリにあるクラスタを右クリックして、[設定の編集] を選択します。
- 2 次のいずれかの方法で、仮想マシンをリソース プールから削除します。
 - 仮想マシンを別のリソース プールへドラッグする。
仮想マシンを移動する前に仮想マシンをパワーオフする必要はありません。
 - 仮想マシンを右クリックし、[インベントリから削除] または [ディスクから削除] を選択する。
仮想マシンを完全に削除するには、事前に仮想マシンをパワーオフする必要があります。

リソース プールの削除

リソース プールをインベントリから削除できます。

手順

- 1 vSphere Client でリソース プールを右クリックし、[削除] を選択します。
確認のダイアログ ボックスが表示されます。
- 2 [はい] をクリックしてリソース プールを削除します。

リソース プールのアドミッション コントロール

リソース プール内の仮想マシンをパワーオンする場合、または子リソース プールの作成を試みる場合、システムは追加のアドミッション コントロールを実行して、リソース プールの制限に違反しないことを確認します。

仮想マシンをパワーオンしたり、リソース プールを作成したりする前に、vSphere Client の [リソース割り当て] タブを使用して、リソースが十分にあるかを確認します。CPU とメモリの [使用可能な予約] 値で、未予約のリソースが表示されます。

使用可能な CPU とメモリ リソースの計算法、およびアクションが実行されるかどうかは、[予約タイプ] によって決まります。

表 8-1. 予約タイプ

予約タイプ	説明
[固定]	選択されたリソース プールに予約のないリソースが十分にあるかどうかをシステムが確認します。ある場合、アクションを実行できます。ない場合、メッセージが表示され、アクションを実行することはできません。
[拡張可能] (デフォルト)	システムは、選択されたリソース プールおよび直接の親のリソース プール内で使用可能なリソースを考慮します。親のリソース プールでも [拡張可能な予約] オプションが選択されている場合は、その親のリソース プールからリソースを借りることができます。[拡張可能な予約] オプションが選択されているかぎり、リソースの借用は、現在のリソース プールの先祖から繰り返し発生します。このオプションを選択しておく、柔軟性が高まりますが、同時に保護が弱くなります。子リソース プールの所有者は、予想以上のリソースを予約することがあります。

事前構成された [予約] または [制限] の設定に違反することはできません。リソース プールを再構成するごとに、または仮想マシンをパワーオンするごとに、システムはすべてのパラメータを検証して、すべてのサービス レベルを保証できるようにします。

拡張可能な予約の例 1

この例では、拡張可能な予約を持つリソース プールの機能を示します。

管理者が、プール P を管理し、2 人の異なるユーザー（または 2 つの異なるグループ）用に 2 つの子リソース プール S1 と S2 を定義するとします。

管理者は、予約を持つ仮想マシンをユーザーがパワーオンしたいことを知っていますが、各ユーザーがどれだけの量を予約する必要があるかは知りません。S1 と S2 の予約を拡張可能にすると、管理者は、プール P の共通の予約をより柔軟に共有および継承できます。

拡張可能な予約を使用しない場合、管理者は、特定の量を明示的に S1 と S2 に割り当てる必要があります。このような特定の割り当ては、特に深いリソース プール階層において柔軟性がなくなる場合があり、リソース プール階層での予約設定が複雑になる場合があります。

拡張可能な予約では、厳密な分離がなくなります。つまり、S1 は、P の予約をすべて使い始めることができ、S2 が直接使用できるメモリまたは CPU がなくなります。

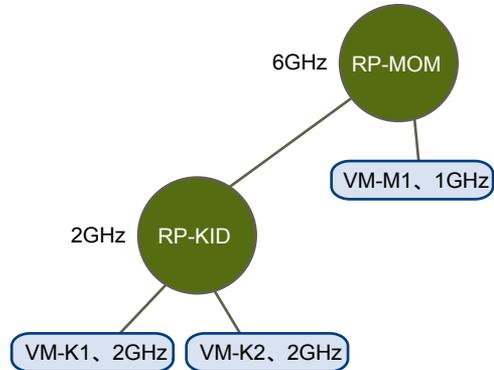
拡張可能な予約の例 2

この例では、拡張可能な予約を持つリソース プールの機能を示します。

図に示すように、次のシナリオを想定しています。

- 親プールの RP-MOM に、6GHz の予約と、1GHz を予約する実行中の 1 台の仮想マシン VM-M1 があります。
- 2GHz の予約を持ち、[拡張可能な予約] が選択された子リソース プール RP-KID を作成します。
- それぞれ 2GHz の予約を持つ 2 台の仮想マシン VM-K1 と VM-K2 を子リソース プールに追加し、これらの仮想マシンのパワーオンを試みます。
- VM-K1 は、(2GHz を持つ) RP-KID からリソースを直接予約できます。
- VM-K2 が使用できるローカル リソースはないので、VM-K2 は親リソース プール RP-MOM からリソースを借ります。RP-MOM は、6GHz マイナス 1GHz (仮想マシンによる予約) マイナス 2GHz (RP-KID による予約) を持ち、3GHz が未予約として残ります。3GHz が使用可能なので、2GHz の仮想マシンをパワーオンすることができます。

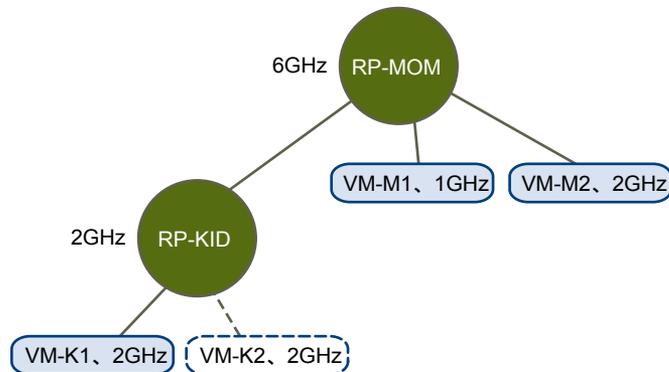
図 8-3. 拡張可能なリソース プールを使用したアドミッション コントロール：正常にパワーオンされた場合



次に、VM-M1 と VM-M2 を使用した別のシナリオについて考えます。

- 予約の合計が 3GHz である、RP-MOM 内の 2 台の仮想マシンをパワーオンします。
- この場合でも、2GHz はローカルで使用可能なので、RP-KID 内の VM-K1 をパワーオンすることができます。
- VM-K2 をパワーオンするとき、RP-KID には未予約の CPU 容量がないため、親を確認します。RP-MOM には未予約で使用可能な容量が 1GHz しかありません（RP-MOM 用に 5GHz が使用済みで、ローカル仮想マシンが 3GHz を予約し、RP-KID が 2GHz を予約しています）。その結果、2GHz の予約が必要な VM-K2 をパワーオンできません。

図 8-4. 拡張可能なリソース プールを使用したアドミッション コントロール：正常にパワーオンされなかった場合



DRS クラスタの作成

DRS クラスタは、ESXi ホストと、共有リソースおよび共有管理インターフェイスを持つ、関連付けられた仮想マシンの集まりです。クラスタレベルのリソース管理の利点を得る前に、DRS クラスタを作成する必要があります。

ある DRS クラスタにホストを追加すると、ホストのリソースはクラスタのリソースの一部になります。このリソースの集約に加えて、DRS クラスタを使用すると、クラスタ全体のリソース プールをサポートし、クラスタレベルのリソース割り当てポリシーを実施できます。次のクラスタレベルのリソース管理機能も使用できます。

ロード バランシング

クラスタ内にあるすべてのホストおよび仮想マシンの CPU およびメモリ リソースの配分と使用率を継続的に監視します。DRS は、これらのメトリックを、クラスタのリソース プールと仮想マシンの属性に設定すべき理想的なリソース使用率、現在の需要、および不均衡なターゲットと比較します。次に、それに応じて仮想マシンへの移行を実行（または推奨）します。[\[仮想マシンの移行 \(P.53\)\]](#) を参照してください。クラスタ内ではじめて仮想マシンをパワーオンしたときに、DRS は仮想マシンを適切なホストに配置するか、推奨を行なって、適切なロード バランシングを維持しようとします。[\[アドミSSION コントロールと初期配置 \(P.52\)\]](#) を参照してください。

電力管理

vSphere Distributed Power Management (DPM) の機能が有効になっている場合、DRS はクラスタレベルおよびホストレベルの容量と、クラスタの仮想マシンに必要な容量（最近の需要履歴を含む）とを比較します。十分な超過容量がある場合は、ホストの電源モードをスタンバイにするか、そうするよう推奨し、容量が必要な場合はホストをパワーオンするよう推奨します。ホストの電源状態に関する推奨に応じて、仮想マシンをホスト間で移行する必要があることがあります。[\[電力リソースの管理 \(P.67\)\]](#) を参照してください。

アフィニティ ルール

アフィニティ ルールを割り当てることで、クラスタ内のホスト上の仮想マシンの配置を制御できます。[\[DRS アフィニティ ルールの使用 \(P.71\)\]](#) を参照してください。

Enhanced vMotion Compatibility (EVC) が有効であるかどうかに応じて、クラスタ内の vSphere フォールト トレランス (vSphere FT) 仮想マシンを使用するときの DRS 動作は異なります。

表 9-1. vSphere FT 仮想マシンおよび EVC との DRS 動作

EVC	DRS (ロード バランシング)	DRS (初期配置)
有効化	有効 (プライマリ仮想マシンとセカンダリ仮想マシン)	有効 (プライマリ仮想マシンとセカンダリ仮想マシン)
無効	無効 (プライマリ仮想マシンとセカンダリ仮想マシン)	無効 (プライマリ仮想マシン) 完全自動化 (セカンダリ仮想マシン)

この章では次のトピックについて説明します。

- [アドミSSION コントロールと初期配置 \(P.52\)](#)
- [仮想マシンの移行 \(P.53\)](#)

- [DRS クラスタの要件 \(P. 55\)](#)
- [DRS クラスタの作成 \(P. 56\)](#)
- [仮想マシンのカスタム自動化レベルの設定 \(P. 57\)](#)
- [DRS の無効化 \(P. 58\)](#)

アドミッション コントロールと初期配置

DRS が有効なクラスタ内の単一の仮想マシンまたは仮想マシンのグループをパワーオンしようとするとき、vCenter Server はアドミッション コントロールを実行します。アドミッション コントロールによって、クラスタ内に仮想マシンをサポートするだけの十分なリソースがあるかどうかを確認されます。

単一の仮想マシンをパワーオンするため、またはグループのパワーオンの試行で仮想マシンのいずれかをパワーオンするために十分なリソースがクラスタにない場合、メッセージが表示されます。十分なリソースがある場合は、各仮想マシンに対して、DRS は仮想マシンを実行するホストの推奨を生成し、次のアクションのいずれかを行います。

- 配置の推奨を自動実行する。
- 配置の推奨を表示し、ユーザーは承諾するか変更するかを選択できる。

注意 スタンドアロン ホストまたは非 DRS クラスタの仮想マシンには、初期配置の推奨は表示されません。パワーオンしたときに、現在存在しているホストに配置されます。

1 台の仮想マシンのパワーオン

DRS クラスタで、単一仮想マシンをパワーオンし、初期配置の推奨を取得します。

単一仮想マシンをパワーオンする場合、2 種類の初期配置の推奨があります。

- パワーオンに必要な前処理はなく、1 台の仮想マシンがパワーオンされる。
ユーザーには、相互に排他的な仮想マシンの初期配置の推奨のリストが表示されます。1 つだけを選択できます。
- パワーオンに必要な前処理を行ってから、1 台の仮想マシンがパワーオンされる。

これらの操作には、スタンバイ モードのホストのパワーオン、別の仮想マシンのホスト間の移行が含まれます。この場合、推奨は複数行で表示され、必要な各操作が表示されます。ユーザーは、この推奨全体を受け入れるか、仮想マシンのパワーオンをキャンセルできます。

グループのパワーオン

複数の仮想マシンを同時にパワーオンできます (グループ パワーオン)。

グループのパワーオンの試行に選択する仮想マシンは、同じ DRS クラスタにある必要はありません。クラスタをまたがって選択できますが、同じデータ センターにある必要があります。また、DRS 以外のクラスタまたはスタンドアロン ホストにある仮想マシンを含めることもできます。これらは自動的にパワーオンされますが、初期配置の推奨には含まれません。

グループ パワーオンの初期配置に関する推奨は、クラスタ単位で提供されます。グループのパワーオンの試行に関する配置関連のすべてのアクションが自動モードの場合、初期配置の推奨は表示されず、仮想マシンがパワーオンされます。仮想マシンの配置関連のアクションのいずれかが手動モードの場合、すべての仮想マシン (自動モードの仮想マシンを含む) は手動でパワーオンされ、初期配置の推奨に含まれます。

パワーオンする仮想マシンが属する DRS クラスタごとに、必要なすべての前提条件が含まれた単一の推奨が作成されます (または、推奨は作成されません)。すべてのクラスタ固有の推奨が、[パワーオン推奨] タブにまとめて表示されます。

自動モードでないグループのパワーオンが試行され、初期配置に関する推奨の対象外である仮想マシン (スタンドアロン ホストまたは DRS 以外のクラスタにある仮想マシン) が含まれている場合、vCenter Server はこれらの仮想マシンを自動的にパワーオンしようとしています。パワーオンが成功すると、[パワーオン開始] タブに一覧表示されます。パワーオンに失敗した仮想マシンは、[パワーオン失敗] タブに一覧表示されます。

例: グループのパワーオン

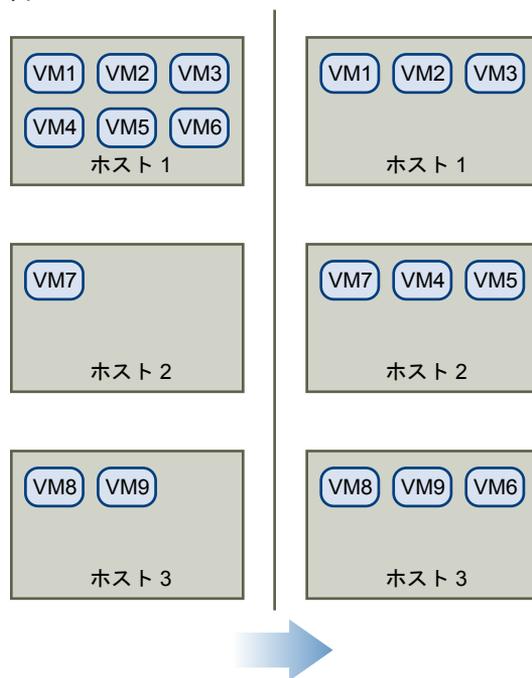
ユーザーが、グループのパワーオンを試行するために、同じデータセンターにある3台の仮想マシンを選択します。最初の2台の仮想マシン（VM1とVM2）は、同じDRSクラスタ（Cluster1）にあり、3番目の仮想マシン（VM3）はスタンドアロンホストにあります。VM1は自動モードで、VM2は手動モードです。このシナリオでは、VM1とVM2をパワーオンするアクションで構成されたCluster1の初期配置の推奨が（[パワーオン推奨]タブに）表示されます。VM3のパワーオン試行は自動的に行われ、成功した場合、[パワーオン開始]タブに表示されます。この試行が失敗した場合は、[パワーオン失敗]タブに一覧表示されます。

仮想マシンの移行

DRSが初期配置を実行するため、クラスタ間の負荷は均衡になりますが、仮想マシンの負荷やリソースの可用性が変化すると、クラスタがアンバランスになります。このようなアンバランスを調整するには、DRSが移行の推奨を作成します。

クラスタでDRSが有効な場合は、負荷をより均一に分散し、前述の不均衡を軽減できます。たとえば、この図の左側にある3台のホストは、不均衡です。ホスト1、ホスト2、ホスト3の容量が同一であり、すべての仮想マシンの構成と負荷（予約が設定されている場合はそれも含む）が同じであるとし、ホスト1には6台の仮想マシンがあるので、ホスト1のリソースは使用過剰になる場合があるのに対し、ホスト2とホスト3では、使用可能なリソースが十分にあります。DRSは、仮想マシンをホスト1からホスト2とホスト3に移行します（または移行を推奨します）。図の右側に、適切にロードバランシングされたあとのホストの構成が表示されます。

図 9-1. ロード バランシング



クラスタが不均衡になった場合、DRSはデフォルトの自動化レベルに応じて、推奨を行うか仮想マシンを移行します。

- クラスタまたはいずれかの仮想マシンが手動または一部自動化の場合、vCenter Serverはリソースのバランスをとるための自動的なアクションを実行しません。代わりに、移行の推奨が使用可能であることがサマリページに表示され、クラスタ内で最も効率的にリソースを使用できる変更の推奨がDRS推奨ページに表示されます。

- 関連するクラスタおよび仮想マシンが完全自動化の場合、vCenter Server は、実行中の仮想マシンを必要に応じてホスト間で移行して、クラスタ リソースが効率的に使用できるようにします。

注意 自動移行設定であっても、ユーザーが個々の仮想マシンを明示的に移行できますが、vCenter Server は、クラスタ リソースを最適化するためにそれらの仮想マシンをほかのホストに移動する場合があります。

デフォルトでは、自動化レベルはクラスタ全体に対して指定されます。個々の仮想マシンについてカスタム自動化レベルを指定することもできます。

DRS 移行のしきい値

DRS 移行のしきい値を使用すると、どの推奨を生成して適用するか（推奨に関与する仮想マシンが完全自動化モードの場合）、または表示するか（手動モードの場合）を指定できます。このしきい値は、ホスト（CPU およびメモリ）の負荷全体で許容可能なクラスタの不均衡の尺度でもあります。

しきい値スライダを移動して、[保守的] から [積極的] までの 5 つの設定のうち、1 つを使用できます。5 つの移行設定は、割り当てられている優先順位レベルに基づいて、推奨を生成します。設定ごとにスライダを右に移動すると、1 段階低い優先順位レベルを含めることができます。[保守的] 設定は、優先順位 1 の推奨（必須の推奨）のみを生成し、その右にある次のレベルは、優先順位 2 以上の推奨を生成します。以後、同様に生成し、[積極的] レベルは優先順位 5 以上の推奨を生成します（つまり、すべての推奨を生成します）。

移行の推奨ごとの優先順位レベルは、クラスタの負荷不均衡のメトリックを使用して計算されます。このメトリックは、vSphere Client にあるクラスタの [サマリ] タブに現在のホストの負荷標準偏差として表示されます。負荷不均衡が大きいほど、高い優先順位の移行の推奨が行われます。このメトリックと推奨の優先順位レベルの計算方法の詳細については、当社のナレッジ ベースの記事「Calculating the priority level of a VMware DRS migration recommendation (VMware DRS による移行の推奨の優先順位レベルの算出)」を参照してください。

推奨が優先順位レベルを取得したあと、そのレベルは設定した移行のしきい値と比較されます。優先順位レベルがしきい値の設定以下の場合、推奨が適用される（関連する仮想マシンが完全自動化モードの場合）か、ユーザーの確認のために表示されます（手動モードまたは一部自動化モードの場合）。

移行の推奨

デフォルトが手動または一部自動化モードのクラスタを作成した場合、vCenter Server は DRS 推奨ページに移行の推奨を表示します。

システムは、ルールを実施してクラスタのリソースのバランスをとるのに必要なだけの推奨を行います。各推奨には、移動する仮想マシン、現在の（ソース）ホストとターゲットホスト、および推奨の理由が含まれます。理由は次のいずれかです。

- CPU 負荷または予約を均衡化する。
- メモリ負荷または予約を均衡化する。
- リソース プールの予約に従う。
- アフィニティ ルールに従う。
- ホストがメンテナンス モードまたはスタンバイ モードに切り替えている最中である。

注意 vSphere Distributed Power Management の機能を使用している場合、DRS は移行の推奨のほかに、ホストの電源状態の推奨を提供します。

DRS クラスタの要件

クラスタ機能を正しく使用するには、DRS クラスタに追加するホストが、特定の要件を満たしている必要があります。

共有ストレージの要件

DRS クラスタには、一定の共有ストレージ要件があります。

管理対象ホストが共有ストレージを使用している必要があります。共有ストレージは、SAN 上にあるのが一般的ですが、NAS 共有ストレージを使用して実装することもできます。

他の共有ストレージの情報については、『vSphere ストレージ』ドキュメントを参照してください。

共有 VMFS ボリュームの要件

DRS クラスタには、一定の共有 VMFS ボリュームの要件があります。

共有 VMFS ボリュームを使用するように、すべての管理対象ホストを構成します。

- ソース ホストとターゲット ホストからアクセスできる VMFS ボリューム上に、全仮想マシンのディスクを設置します。
- VMFS ボリュームが、仮想マシンのすべての仮想ディスクを保存するのに十分なサイズであることを確認します。
- ソース ホストとターゲット ホスト上のすべての VMFS ボリュームがボリューム名を使用すること、およびすべての仮想マシンがそれらのボリューム名を使用して仮想ディスクを指定することを確認します。

注意 仮想マシン スワップ ファイルも、**.vmdk** 仮想ディスクと同様に、ソース ホストとターゲット ホストがアクセスできる VMFS 上に配置される必要があります。すべてのソース ホストとターゲット ホストが ESX Server 3.5 以降で、ホストのローカル スワップを使用している場合、この要件は適用されません。この場合は、非共有ストレージのスワップ ファイルを使用する vMotion がサポートされます。スワップ ファイルはデフォルトで VMFS 上に配置されますが、管理者は、仮想マシンの詳細構成オプションを使用してファイルの場所をオーバーライドできます。

プロセッサの互換性要件

DRS クラスタには、一定のプロセッサ互換性要件があります。

DRS の機能が制限されないようにするためには、クラスタ内のソース ホストとターゲット ホストのプロセッサの互換性を最大にする必要があります。

vMotion は、実行中の仮想マシンのアーキテクチャ状態を、基盤となる ESXi ホスト間で移行します。vMotion の互換性とは、ソース ホストのプロセッサのサスペンドに用いたのと同等の命令を使用して、ターゲット ホストのプロセッサが実行をレジュームできなければならないことを意味します。vMotion での移行に必要な互換性を保つためには、プロセッサのクロック速度とキャッシュ サイズは異なってもかまいませんが、プロセッサが同じクラスのベンダー (Intel と AMD) 製で、同じプロセッサ ファミリでなければなりません。

プロセッサ ファミリはプロセッサ ベンダーによって決められています。同じファミリ内の異なるプロセッサ バージョンは、プロセッサのモデル、ステッピング レベル、拡張機能を比較することで区別できます。

プロセッサ ベンダーが、同じプロセッサ ファミリ内で大幅なアーキテクチャ変更 (64 ビット拡張や SSE3 など) を行うことがあります。vMotion での正常な移行を保証できない場合、これらの例外を確認します。

vCenter Server には、vMotion で移行した仮想マシンがプロセッサの互換性要件を確実に満たすようにする機能が備わっています。これには、次の機能が含まれます。

- **Enhanced vMotion Compatibility (EVC)** : EVC を使用して、クラスタ内のホストの vMotion 互換性を確保できます。ホスト上の実際の CPU が異なる場合でも、EVC によって 1 つのクラスタ内のすべてのホストが確実に同じ CPU 機能セットを仮想マシンに提供するようになります。これにより、互換性のない CPU が原因で vMotion での移行が失敗することがなくなります。

EVC は、クラスタ設定ダイアログ ボックスで構成します。クラスタで EVC を使用するためには、クラスタ内のホストが特定の要件を満たす必要があります。EVC および EVC の要件については、『vCenter Server およびホスト管理』ドキュメントを参照してください。

- **CPU 互換性マスク** : vCenter Server は仮想マシンで使用できる CPU 機能をターゲット ホストの CPU 機能と比較して、vMotion での移行が可能かどうかを判断します。個々の仮想マシンに CPU 互換性マスクを適用することにより、特定の CPU 機能を仮想マシンから隠し、互換性のない CPU が原因で vMotion での移行が失敗する可能性を減らすことができます。

DRS クラスタの vMotion 要件

DRS クラスタには、一定の vMotion 要件があります。

DRS 移行の推奨を使用できるようにする場合は、クラスタ内のホストが vMotion ネットワークの一部である必要があります。ホストが vMotion ネットワーク内にない場合でも、DRS は初期配置の推奨を行うことができます。

vMotion 用に構成するためには、クラスタ内の各ホストが次の要件を満たしている必要があります。

- vMotion は、Raw ディスク、または MSCS (Microsoft Cluster Service) でクラスタリングしたアプリケーションの移行をサポートしていません。
- vMotion では、vMotion に対応したすべての管理対象ホスト間に、プライベートなギガビット イーサネット移行ネットワークが必要です。管理対象ホスト上で vMotion を有効にする場合は、管理対象ホストに固有のネットワーク ID オブジェクトを構成し、これをプライベートな移行用ネットワークに接続します。

DRS クラスタの作成

vSphere Client で[新規クラスタ] ウィザードを使用して、DRS クラスタを作成します。

開始する前に

クラスタを作成するのに特別なライセンスは必要ありませんが、クラスタで vSphere DRS (または vSphere HA) を有効にするにはライセンスが必要です。

手順

- 1 vSphere Client でデータ センターまたはフォルダを右クリックして、[新規クラスタ] を選択します。
- 2 [名前] テキスト ボックスでクラスタの名前を指定します。
この名前は、vSphere Client のインベントリ パネルに表示されます。
- 3 DRS 機能を有効にするには、[vSphere DRS] ボックスをクリックします。
[vSphere HA] をクリックして、vSphere HA 機能を有効にすることもできます。
- 4 [次へ] をクリックします。

- 5 DRS のデフォルトの自動化レベルを選択します。

自動化レベル	操作
手動	<ul style="list-style-type: none"> ■ 初期配置：推奨されるホストが表示されます。 ■ 移行：推奨が表示されます。
一部自動化	<ul style="list-style-type: none"> ■ 初期配置：自動です。 ■ 移行：推奨が表示されます。
完全自動化	<ul style="list-style-type: none"> ■ 初期配置：自動です。 ■ 移行：推奨が自動的に実行されます。

- 6 DRS の移行のしきい値を設定します。

- 7 [次へ] をクリックします。

- 8 クラスタのデフォルトの電力管理設定を指定します。

電力管理が有効になっている場合、vSphere DPM しきい値の設定を選択します。

- 9 [次へ] をクリックします。

- 10 適切と思われる場合は、EVC (Enhanced vMotion Compatibility) 機能を有効にし、この機能を操作するモードを選択します。

- 11 [次へ] をクリックします。

- 12 仮想マシンのスワップ ファイルの場所を選択します。

スワップファイルは、その仮想マシンと同じディレクトリに保存することも、ホストによって指定されたデータストアに保存することもできます (ホストのローカル スワップ)。

- 13 [次へ] をクリックします。

- 14 サマリ ページで選択したオプションのリストを確認します。

- 15 [終了] をクリックし、クラスタの作成を完了します。または [戻る] をクリックして戻り、クラスタの設定を修正します。

新規クラスタにはホストや仮想マシンは含まれていません。

ホストおよび仮想マシンをクラスタに追加するには、[「クラスタへのホストの追加\(P.59\)」](#) および [「クラスタからの仮想マシンの削除 \(P. 61\)」](#) を参照してください。

仮想マシンのカスタム自動化レベルの設定

DRS クラスタを作成したあとで、個々の仮想マシンの自動化レベルをカスタマイズして、クラスタのデフォルト自動化レベルをオーバーライドできます。

たとえば、完全に自動化されたクラスタ内の特定の仮想マシンに対して [手動] を選択したり、手動のクラスタ内の特定の仮想マシンに対して [一部自動化] を選択したりできます。

仮想マシンを [無効] に設定すると、vCenter Server でその仮想マシンの移行や、移行に関する推奨を行うことがなくなります。これを「登録先ホストへの仮想マシンの固定化」と言います。

注意 クラスタの EVC (Enhanced vMotion Compatibility) を有効にしていない場合、フォールトトレランス対応の仮想マシンは DRS を無効に設定します。画面には表示されますが、自動化モードを割り当てることはできません。

手順

- 1 vSphere Client で、インベントリにあるクラスタを右クリックして、[設定の編集] を選択します。
- 2 vSphere DRS の左側のペインで、[仮想マシンのオプション] を選択します。
- 3 [各仮想マシンの自動化レベルを有効にする] チェック ボックスを選択します。

- 4 (オプション) 個々の仮想マシンのオーバーライドを一時的に無効にするには、[各仮想マシンの自動化レベルを有効にする] チェック ボックスを選択解除します。
 チェック ボックスをもう一度選択すると、仮想マシンの設定は元に戻ります。
- 5 (オプション) クラスタ内のすべての vMotion アクティビティを一時的にサスペンドするには、クラスタを手動モードに切り替え、[各仮想マシンの自動化レベルを有効にする] チェック ボックスを選択解除します。
- 6 1 つ以上の仮想マシンを選択します。
- 7 [自動化レベル] 列をクリックし、ドロップダウン メニューから自動化レベルを選択します。

オプション	説明
手動	配置と移行の推奨が表示されますが、推奨が手動で適用されるまで実行はされません。
完全自動化	配置と移行の推奨が自動的に実行されます。
一部自動化	初期配置が自動的に実行されます。移行の推奨は表示されますが、実行はされません。
無効	vCenter Server では、仮想マシンの移行や、移行のための推奨は行われません。

- 8 [OK] をクリックします。

注意 ほかの当社製品または機能 (vSphere vApp および vSphere Fault Tolerance など) が、DRS クラスタ内の仮想マシンの自動化レベルをオーバーライドする場合があります。詳細については、個別の製品ドキュメントを参照してください。

DRS の無効化

クラスタの DRS をオフにできます。

DRS を無効化すると、DRS を再度オンにしたときに、クラスタのリソース プールの階層およびアフィニティ ルールが再構築されません。したがって、DRS を無効にすると、リソース プールはクラスタから削除されます。リソース プールが失われないようにするには、DRS を無効にするのではなく、DRS 自動化レベルを手動に変更し (また、仮想マシンによるオーバーライドを無効にして)、DRS をサスペンドしてください。これにより、自動の DRS アクションは実行されませんが、リソース プールの階層は保存されます。

手順

- 1 vSphere Client のインベントリでクラスタを選択します。
- 2 右クリックして [設定の編集] を選択します。
- 3 左側のパネルで、[全般] を選択し、[vSphere DRS をオンにする] チェック ボックスを選択解除します。
- 4 [OK] をクリックして DRS をオフにします。

DRS クラスタを使用したリソース管理

DRS クラスタを作成したあとで、それをカスタマイズしたり、それを使用してリソースを管理したりできます。

DRS クラスタおよびそれに含まれるリソースをカスタマイズするには、アフィニティ ルールを構成し、ホストおよび仮想マシンを追加または削除します。クラスタの設定とリソースを定義したら、そのクラスタが有効なクラスタであり、有効であり続けるようにする必要があります。ユーザーは有効な DRS クラスタを使用して電力リソースを管理し、vSphere HA と相互運用できます。

この章では次のトピックについて説明します。

- [クラスタへのホストの追加 \(P. 59\)](#)
- [クラスタへの仮想マシンの追加 \(P. 60\)](#)
- [クラスタからの仮想マシンの削除 \(P. 61\)](#)
- [クラスタからのホストの削除 \(P. 61\)](#)
- [DRS クラスタの妥当性 \(P. 62\)](#)
- [電力リソースの管理 \(P. 67\)](#)
- [DRS アフィニティ ルールの使用 \(P. 71\)](#)

クラスタへのホストの追加

クラスタにホストを追加する手順は、同じ vCenter Server によって管理されているホスト（管理対象ホスト）と、そのサーバによって管理されていないホストとで異なります。

ホストを追加すると、そのホストにデプロイされている仮想マシンはクラスタの一部になり、DRS は、一部の仮想マシンをクラスタ内の別のホストへ移行するよう推奨する場合があります。

クラスタへの管理対象ホストの追加

すでに vCenter Server によって管理されているスタンドアロン ホストを DRS クラスタに追加すると、そのホストのリソースがクラスタに関連付けられます。

ユーザーは、既存の仮想マシンとリソース プールをクラスタのルート リソース プールと関連付けるか、それともリソース プール階層を移動するかを決定できます。

注意 ホストに子のリソース プールまたは仮想マシンがない場合、ホストのリソースがクラスタに追加されますが、最上位レベルのリソース プールを持つリソース プール階層は作成されません。

手順

- 1 インベントリまたはリストのビューからホストを選択します。
- 2 ホストをターゲット クラスタ オブジェクトにドラッグします。

- 3 ホストの仮想マシンとリソース プールをどうするかを選択します。
 - [このホストの仮想マシンをクラスタのルート リソース プールに入れます]

vCenter Server はホストの既存のリソース プールをすべて削除し、ホストの階層内にあるすべての仮想マシンは、ルートに接続されます。シェア割り当てはリソース プールに対して相対的なので、リソース プール階層を破棄するこのオプションを選択したあとで、手動で仮想マシンのシェアを変更する必要があることがあります。
 - [このホストの仮想マシンとリソース プール用のリソース プールを作成します]

vCenter Server は、クラスタの直接の子になる最上位レベルのリソース プールを作成し、ホストのすべての子をその新規リソース プールに追加します。この新しい最上位レベルのリソース プールの名前を指定できます。デフォルトは、[<host_name> から移動] です。

ホストがクラスタに追加されます。

管理されていないホストのクラスタへの追加

管理されていないホストをクラスタに追加できます。現在、そのクラスタと同じ vCenter Server システムによって管理されておらず、vSphere Client に認識されていないホストを追加できます。

手順

- 1 ホストの追加先のクラスタを選択して、右クリック メニューから [ホストの追加] を選択します。
- 2 ホスト名、ユーザー名、およびパスワードを入力して、[次へ] をクリックします。
- 3 概要情報を確認して、[次へ] をクリックします。
- 4 ホストの仮想マシンとリソース プールをどうするかを選択します。
 - [このホストの仮想マシンをクラスタのルート リソース プールに入れます]

vCenter Server はホストの既存のリソース プールをすべて削除し、ホストの階層内にあるすべての仮想マシンは、ルートに接続されます。シェア割り当てはリソース プールに対して相対的なので、リソース プール階層を破棄するこのオプションを選択したあとで、手動で仮想マシンのシェアを変更する必要があることがあります。
 - [このホストの仮想マシンとリソース プール用のリソース プールを作成します]

vCenter Server は、クラスタの直接の子になる最上位レベルのリソース プールを作成し、ホストのすべての子をその新規リソース プールに追加します。この新しい最上位レベルのリソース プールの名前を指定できます。デフォルトは、[<host_name> から移動] です。

ホストがクラスタに追加されます。

クラスタへの仮想マシンの追加

次の 3 つの方法で、仮想マシンをクラスタに追加できます。

- クラスタにホストを追加すると、そのホスト上のすべての仮想マシンがクラスタに追加されます。
- 仮想マシンを作成すると、[新規仮想マシン] ウィザードは、その仮想マシンを配置する場所を尋ねてきます。スタンドアロン ホストまたはクラスタを選択でき、そのホストまたはクラスタの内部にある任意のリソース プールを選択できます。
- [仮想マシンの移行] ウィザードを使用して、仮想マシンをスタンドアロン ホストからクラスタに、またはあるクラスタから別のクラスタに移行できます。このウィザードを開始するには、仮想マシン オブジェクトをクラスタ オブジェクトの上へドラッグするか、仮想マシン名を右クリックして、[移行] を選択します。

注意 クラスタ内のリソース プールへ仮想マシンを直接ドラッグできます。この場合、[仮想マシンの移行] ウィザードが起動しますが、リソース プール選択ページは表示されません。リソース プールがリソースを制御するので、クラスタ内のホストに直接移行することはできません。

クラスタからの仮想マシンの削除

仮想マシンをクラスタから削除できます。

次の 2 つの方法で、仮想マシンをクラスタから削除できます。

- ホストをクラスタから削除すると、ほかのホストに移行していないパワーオフ状態の仮想マシンもすべて削除されます。ホストを削除できるのは、ホストがメンテナンス モードになっているか、切断されている場合だけです。DRS クラスタからホストを削除した場合、クラスタは、オーバーコミットされるために黄色で表示されることがあります。
- [仮想マシンの移行]ウィザードを使用して、仮想マシンをクラスタからスタンドアロン ホストに、またはあるクラスタから別のクラスタに移行できます。このウィザードを開始するには、仮想マシンオブジェクトをクラスタ オブジェクトの上へドラッグするか、仮想マシン名を右クリックして、[移行] を選択します。

仮想マシンが DRS クラスタのルール グループのメンバーである場合、vCenter Server は、移行の続行を許可する前に警告を表示します。この警告は、従属仮想マシンの移行が自動的に実行されないことを示します。移行を続行するには、警告を了承する必要があります。

クラスタからのホストの削除

DRS クラスタからホストを削除すると、リソース プールの階層や仮想マシンに影響を与え、また無効なクラスタを作成する恐れがあります。影響を受けるオブジェクトを考慮してから、ホストを削除します。

- リソース プールの階層：ホストをクラスタから削除すると、ホストをクラスタに追加した時点で DRS クラスタを使用してホストのリソース プールを移動した場合でも、ホストが保持するのはルートリソース プールだけになります。この場合、階層はクラスタで保持されます。ホスト固有のリソース プール階層を作成できます。

注意 最初にホストをメンテナンス モードにすることで、クラスタからホストを確実に削除できます。メンテナンス モードにせず、クラスタからホストを削除する前にホストを切断すると、ホストはクラスタ階層を反映したりリソース プールを保持します。

- 仮想マシン：クラスタから削除する前にホストをメンテナンス モードにしておく必要があり、ホストをメンテナンス モードにするには、パワーオン状態の仮想マシンをすべてそのホストから移行する必要があります。また、ホストをメンテナンス モードに切り替えることを要求する場合、そのホストにあるパワーオフ状態のすべての仮想マシンを、クラスタ内の別のホストに移行するか尋ねられます。
- 無効なクラスタ：クラスタからホストを削除すると、クラスタで使用可能なリソースが減少します。クラスタ内のすべての仮想マシンとリソース プールの予約を満たすのに十分なリソースがまだクラスタにある場合、クラスタは、リソース割り当てを調整して、減少後のリソース量を反映させます。クラスタに、すべてのリソース プールの予約を満たすのに十分なリソースはないが、すべての仮想マシンの予約を満たす十分なリソースはある場合は、アラームが発行され、クラスタが黄色で表示されます。DRS は、実行を続けます。

ホストをメンテナンス モードに切り替える

メモリの増設など、ホストの保守作業を行う必要がある場合は、ホストをメンテナンス モードにします。ホストは、ユーザーの要求があった場合のみ、メンテナンス モードを開始または終了します。

メンテナンス モードに切り替えるホストで実行中の仮想マシンは、別のホストに（手動で、または DRS によって自動的に）移行するか、シャットダウンする必要があります。ホストは、稼働中のすべての仮想マシンがパワーオフされるか別のホストへ移行されるまで、[メンテナンス モードに切り替えています] と表示された状態になります。メンテナンス モードのホストで仮想マシンをパワーオンしたり、そのホストに仮想マシンを移行したりすることはできません。

ホスト上に実行中の仮想マシンがなくなると、ホストのアイコンが変化して [メンテナンス中] が組み込まれ、ホストのサマリ パネルに新しい状態が表示されます。メンテナンス モードの間は、そのホストでは、仮想マシンをデプロイしたり、パワーオンしたりすることはできません。

注意 DRS は、要求されたモードにホストが切り替わると vSphere HA のフェイルオーバー レベルに違反することになる場合、メンテナンス モードまたはスタンバイ モードに切り替わるホストから仮想マシンが移行することを推奨（完全自動モードの場合は実行）しません。

手順

- 1 vSphere Client インベントリで、ホストを右クリックして [メンテナンス モードに切り替える] を選択します。
 - ホストが、一部自動化されているか手動の DRS クラスタの一部の場合、ホストで実行中の仮想マシン用移行推奨リストが表示されます。
 - ホストが、自動化された DRS クラスタの一部になっている場合、メンテナンス モードに切り替わると、仮想マシンは別のホストに移行します。
- 2 必要に応じて、[推奨の適用] をクリックします。

[メンテナンス モードの終了] を選択するまで、ホストはメンテナンス モードです。

クラスタからのホストの削除

ホストをクラスタから削除できます。

手順

- 1 vSphere Client で、インベントリ内のホストを右クリックし、[メンテナンス モードに切り替える] を選択します。
- 2 ホストがメンテナンス モードのとき、そのホストを別のインベントリの場所（最上位レベルのデータ センター、または別のクラスタ）にドラッグしてください。

ホストをクラスタから削除後、次のタスクを実行できます。

- ホストを vCenter Server から除去する：ホストを右クリックし、[削除] を選択します。
- そのホストをスタンドアロン ホストとして vCenter Server の下で稼働させる：ホストを右クリックし、[メンテナンス モードの終了] を選択します。
- ホストを他のクラスタにドラッグします。

ホストを移動すると、そのホストのリソースがクラスタから削除されます。ホストのリソース プール階層をクラスタに移動した場合、その階層はクラスタで保持されます。

スタンバイ モードの使用

ホスト マシンがスタンバイ モードになると、パワーオフされます。

通常、電力の使用を最適にするために、ホストは vSphere DPM 機能によってスタンバイ モードになります。手動でホストをスタンバイ モードにすることもできます。ただし、次に実行するときに、DRS が変更を取り消す（または取り消すように推奨する）ことがあります。ホストを強制的にオフのままにするには、メンテナンス モードにして、パワーオフします。

DRS クラスタの妥当性

vSphere Client には、DRS クラスタが有効か、オーバーコミットされているか（黄色）、あるいは無効か（赤）が表示されます。

DRS クラスタは、いくつかの理由でオーバーコミットまたは無効になります。

- クラスタは、ホストに障害が起きた場合にオーバーコミットされる可能性があります。

- クラスタは、vCenter Server が使用不能で、ホストに直接接続した vSphere Client を使用して仮想マシンをパワーオンした場合に、無効になります。
- クラスタは、仮想マシンのフェイルオーバーの最中にユーザーが親リソース プール上で予約を減らした場合に、無効になります。
- vCenter Server が使用不能なときに、ホストに接続した vSphere Client を使用してホストまたは仮想マシンに変更を加えた場合、それらの変更は有効です。しかし、vCenter Server が再び使用可能になると、クラスタ要件がもはや満たされていないためにクラスタが赤または黄色で表示されることがあります。

クラスタの妥当性のシナリオについて考える場合、次の用語を理解する必要があります。

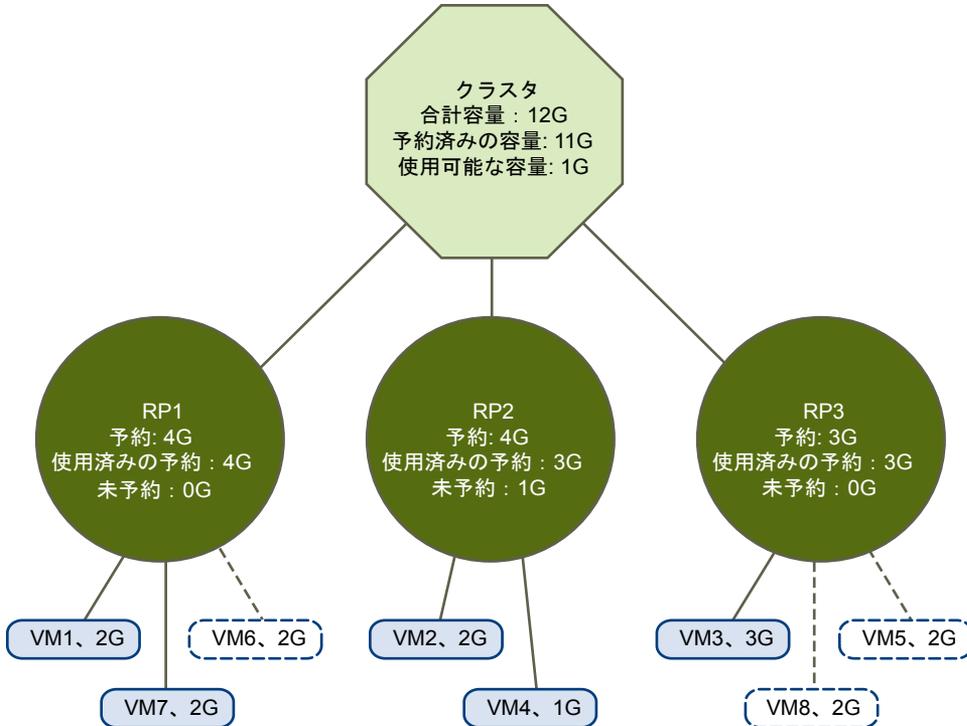
予約	ユーザーが入力した、固定で保証されたリソース プールの割り当て。
使用済みの予約	それぞれの子リソース プールの予約または使用済みの予約（いずれか大きな方）の合計。再帰的に加算されます。
予約なし	負でない値で、リソース プールのタイプによって異なります。 <ul style="list-style-type: none"> ■ 拡張不能なリソース プール：予約 - 使用済み予約 ■ 拡張可能なリソース プール（予約 - 使用済み予約） + 先祖のリソース プールから借用できる未予約リソース

有効な DRS クラスタ

有効なクラスタには、すべての予約を満たして、実行中のすべての仮想マシンをサポートするのに十分なリソースがあります。

次図に、固定リソース プールを持つ有効なクラスタの例と、その CPU およびメモリ リソースの計算方法を示します。

図 10-1. 固定リソース プールを持つ有効なクラスタ



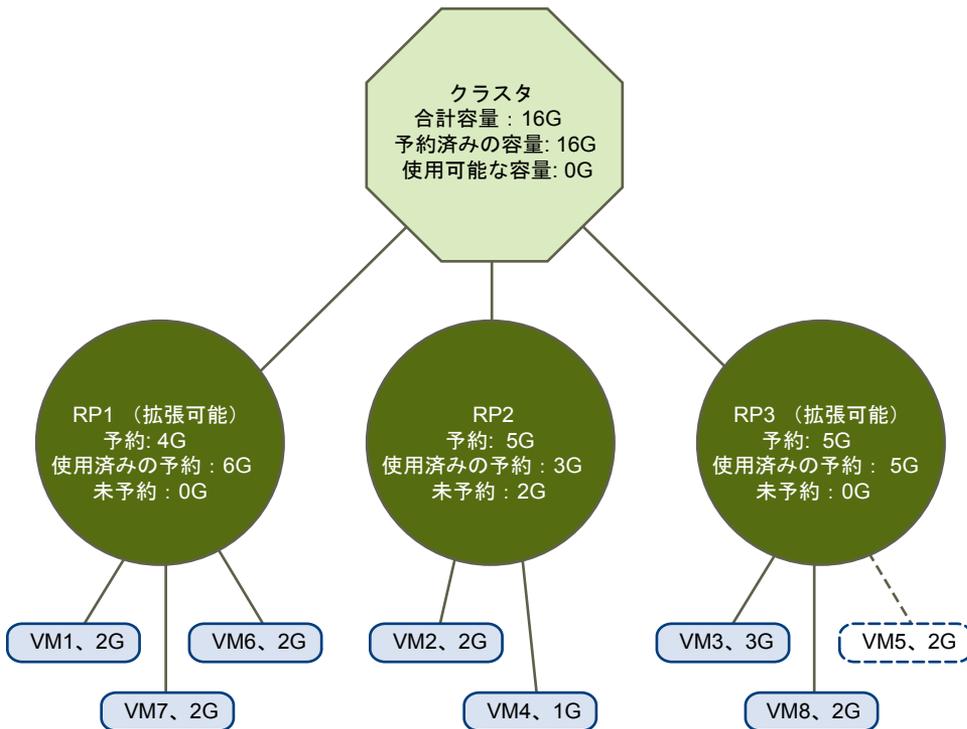
このクラスタの特性は次のとおりです。

- クラスタのリソースは、合計 12GHz です。
- それぞれタイプが [固定]（[拡張可能な予約] が選択されていない）のリソース プールが 3 つあります。

- 3つのリソース プールをあわせた予約の合計は、11GHz (4+4+3GHz) です。合計は、クラスタの [予約済みの容量] フィールドに示されます。
- RP1 は、4GHzの予約を使用して作成されています。それぞれ2GHzの2台の仮想マシン (VM1 および VM7) がパワーオンされます ([使用済みの予約] : 4GHz)。さらに仮想マシンをパワーオンするリソースは、残っていません。VM6 は、パワーオンされていないものとして表示されます。VM6 は予約を消費しません。
- RP2 は、4GHzの予約を使用して作成されています。1GHzと2GHzの2台の仮想マシンがパワーオンされています ([使用済みの予約] : 3GHz)。1GHzが未予約で残っています。
- RP3 は、3GHzの予約を使用して作成されています。3GHzを持つ1台の仮想マシンがパワーオンされています。さらに仮想マシンをパワーオンするのに使用可能なリソースはありません。

次図に、予約タイプ [拡張可能] を使用した一部のリソース プール (RP1 および RP3) を持つ有効なクラスタの例を示します。

図 10-2. 拡張可能なリソース プールを持つ有効なクラスタ



有効なクラスタを次のように構成できます。

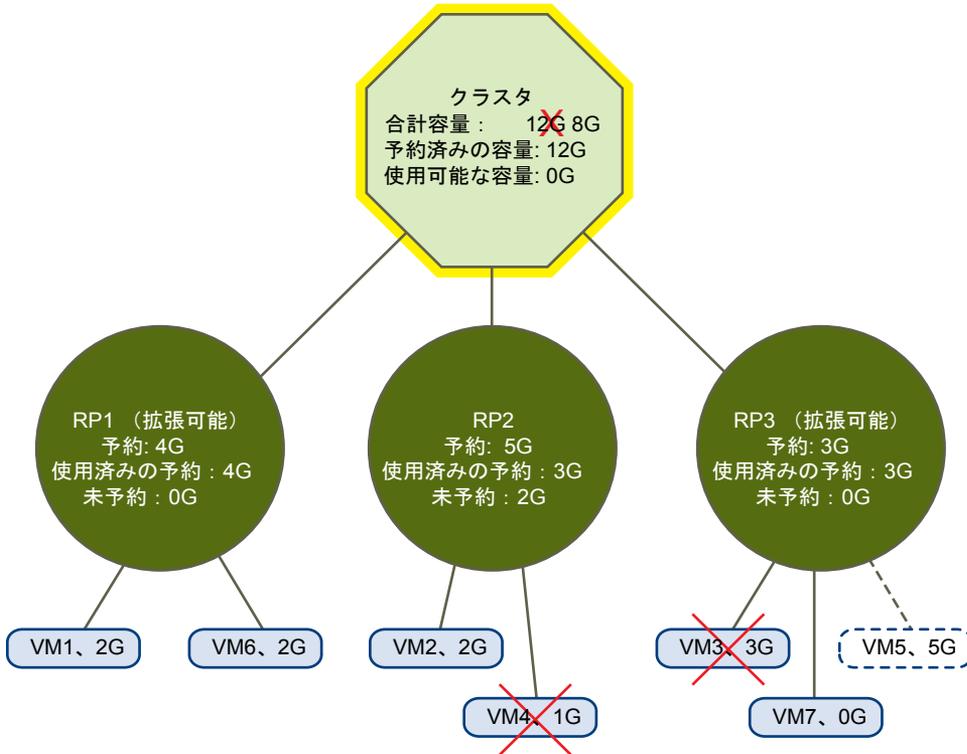
- クラスタのリソースは、合計 16GHz です。
 - RP1 および RP3 はタイプ [拡張可能] で、RP2 はタイプ固定です。
 - 3つのリソース プールを統合した使用済みの予約の合計は、16GHz です (RP1 の 6GHz、RP2 の 5GHz、および RP3 の 5GHz)。16GHz は、最上位レベルのクラスタの [予約済みの容量] として表示されます。
 - RP1 は、4GHzの予約を使用して作成されています。それぞれ2GHzの3台の仮想マシンがパワーオンされています。これらの仮想マシンのうちの2台 (たとえば、VM1 と VM7) は、RP1 の予約を使用でき、3台目の仮想マシン (VM6) は、クラスタのリソース プールから予約を使用できません。(このリソース プールのタイプが [固定] の場合、追加の仮想マシンのパワーオンはできません)。
 - RP2 は、5GHzの予約を使用して作成されています。1GHzと2GHzの2台の仮想マシンがパワーオンされています ([使用済みの予約] : 3GHz)。2GHzが未予約で残っています。
- RP3 は、5GHzの予約を使用して作成されています。3GHzと2GHzの2台の仮想マシンがパワーオンされています。このリソース プールのタイプが [拡張可能] であっても、親の余分なリソースがすでに RP1 によって使用されているので、追加の2GHzの仮想マシンをパワーオンすることはできません。

オーバーコミットされた DRS クラスタ

クラスタがオーバーコミット（黄色）で表示されるのは、リソース プールと仮想マシンのツリーが内部的に整合性があるが、子リソース プールによって予約されるすべてのリソースをサポートする容量がクラスタにない場合です。

ホストが使用不能になると、そのすべての仮想マシンが使用不能になるので、実行中のすべての仮想マシンをサポートするのに十分なリソースが常に存在するようになります。通常、クラスタは、クラスタの容量が突然減少した場合、たとえば、クラスタ内のホストが使用不能になった場合に黄色になります。クラスタが黄色にならないように、十分な追加クラスタ リソースを残すことを推奨します。

図 10-3. 黄色のクラスタ



この例では、次のようになります。

- クラスタのリソースは、それぞれ 4GHz の 3 台のホストから合計 12GHz です。
- 3 つのリソース プールが合計で 12GHz を予約します。
- 統合した 3 つのリソース プールによって使用される予約の合計は、12GHz (4+5+3 GHz) です。これは、クラスタ内の [予約済みの容量] として表示されます。
- 4GHz ホストの 1 台が使用不能になり、合計リソースが 8GHz に減ります。
- 同時に、障害の起きたホスト上で実行されていた VM4 (1GHz) および VM3 (3GHz) は、すでに実行されていません。
- クラスタは、合計で 6GHz を必要とする仮想マシンを実行中です。クラスタではまだ 8GHz が使用可能であり、これは、仮想マシンの要件を満たすのに十分です。

リソース プールの 12GHz の予約を満たすことはすでにできないので、クラスタは黄色で表示されます。

無効な DRS クラスタ

DRS が有効に設定されているクラスタは、ツリーが内部的に整合性がない、すなわち、リソース制約が遵守されていない場合に無効（赤）になります。

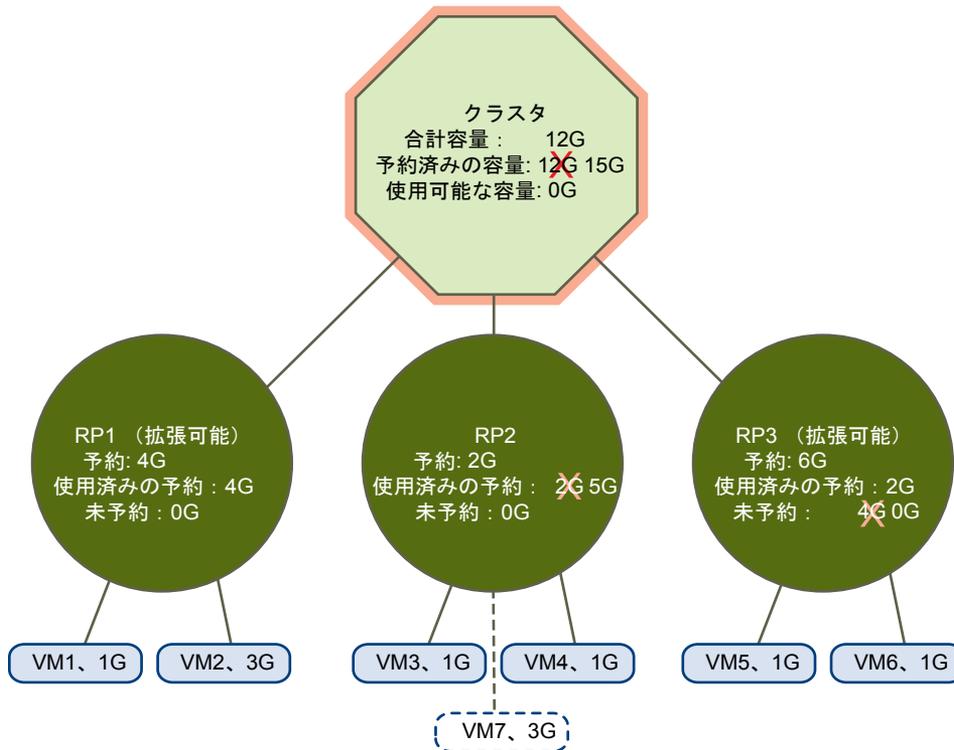
クラスタ内のリソースの合計量は、クラスタが赤になるかどうかに影響を与えません。クラスタは、ルートレベルに十分なリソースが存在しても、子レベルで不整合があると、赤で表示される場合があります。

赤の DRS クラスタの問題は、1 台以上の仮想マシンをパワーオフするか、十分なリソースがあるツリーの一部に仮想マシンを移動するか、あるいは赤い部分のリソース プール設定を編集することによって解決できます。一般的に、リソースの追加は、黄色の状態のときにのみ役立ちます。

仮想マシンのフェイルオーバーの最中にリソース プールを再構成した場合も、クラスタが赤で表示されることがあります。フェイルオーバー中の仮想マシンは、切断され、親リソース プールによって使用される予約の計算に含まれません。フェイルオーバーが完了する前に、親リソース プールの予約を減らすことができます。フェイルオーバーが完了したら、仮想マシンのリソースが親リソース プールに再びチャージされます。プールの使用量が新しい予約よりも多くなった場合、クラスタは赤になります。

ユーザーがリソース プール 2 の下で、3GHz の予約で仮想マシンを（サポートされていない方法で）起動した場合、次の図に示すようにクラスタは赤になります。

図 10-4. 赤のクラスタ



電力リソースの管理

vSphere DPM (Distributed Power Management) の機能を使用すると、クラスタのリソース使用状況に応じてホストをパワーオンおよびパワーオフするため、クラスタの電力消費量を削減できます。

vSphere DPM は、クラスタにあるすべての仮想マシンのメモリ リソース、および CPU リソースの度重なる需要を監視し、この需要と、クラスタのすべてのホストで使用可能なリソース キャパシティの総量とを比較します。余剰キャパシティが十分にある場合、vSphere DPM は 1 台以上のホストをスタンバイ モードにし、それらの仮想マシンを別のホストに移行したあとでパワーオフします。逆に、容量が不足しそうな場合、DRS はホストのスタンバイ モードを終了し (パワーオンし)、vMotion を使用して仮想マシンをそれらのホストに移行します。これらの計算を行う場合、vSphere DPM は現在の需要だけではなく、ユーザーが指定した仮想マシンのリソース予約についても考慮します。

注意 ESXi ホストは、vCenter Server が管理するクラスタ内で実行されている場合を除き、自動的にスタンバイ モードを終了することはできません。

vSphere DPM では、次の 3 つの電力管理プロトコルを使用して、ホストのスタンバイ モードを終了させることができます。IPMI (Intelligent Platform Management Interface)、Hewlett-Packard iLO (Integrated Lights-Out)、または WOL (Wake-On-LAN) です。各プロトコルで独自のハードウェア サポートおよび構成が必要です。ホストがこれらのプロトコルのいずれもサポートしていない場合、vSphere DPM はそのホストをスタンバイ モードにできません。ホストが複数のプロトコルをサポートしている場合に使用される順番は、IPMI、iLO、WOL です。

注意 初回のパワーオンを行わずに、スタンバイ モードのホストを切断したり、DRS クラスタの外に移動したりしないでください。vCenter Server はそのホストを再びパワーオンできなくなります。

vSphere DPM 用の IPMI 設定または iLO 設定の構成

IPMI はハードウェア レベルの仕様であり、Hewlett-Packard iLO は組み込みのサーバ管理テクノロジーです。どちらも、コンピュータをリモートで監視および制御するためのインターフェイスを記述および提供します。

各ホストで次の手順を行う必要があります。

開始する前に

IPMI および iLO は、ハードウェア制御機能にアクセスするためのゲートウェイを提供するためにハードウェア Baseboard Management Controller (BMC) を必要とします。また、これらはそのインターフェイスにシリアル接続または LAN 接続を使用してリモート システムからアクセスできるようにします。ホスト自体がパワーオフされている場合でも、BMC はパワーオンされた状態です。適切に有効にすると、BMC はリモート パワーオン コマンドに応答できます。

IPMI または iLO を Wake プロトコルとして使用する場合は、BMC を構成する必要があります。BMC の構成手順は、モデルによって異なります。詳細については、ベンダーのドキュメントを参照してください。IPMI を使用する場合は、BMC LAN チャネルが常時使用可能で、オペレータ特権コマンドを許可するように構成する必要もあります。一部の IPMI システムでは、「IPMI over LAN」を有効にする場合は、BIOS 内でこれを構成し、特定の IPMI アカウントを指定する必要があります。

IPMI のみを使用している vSphere DPM は、MD5 認証およびプレーンテキスト ベースの認証をサポートしていますが、MD2 ベースの認証はサポートされていません。MD5 がサポートされており、オペレータ ロールでそれを使用可能であることをホストの BMC が報告している場合、vCenter Server は MD5 を使用します。または、プレーンテキスト ベースの認証がサポートされており、使用可能であることを BMC が報告している場合、プレーンテキスト ベースの認証を使用します。MD5 認証もプレーンテキスト認証も使用できない場合、ホストで IPMI を使用できません。vCenter Server は Wake-on-LAN を使用しようとします。

手順

- 1 vSphere Client のインベントリでホストを選択します。
- 2 [構成] タブをクリックします。
- 3 [電力管理] をクリックします。

- 4 [プロパティ] をクリックします。
- 5 次の情報を入力します。
 - BMC アカウントのユーザー名とパスワード (このユーザー名は、リモート側からホストをパワーオンできる必要があります)。
 - BMC に関連付けられている NIC の IP アドレス (ホストの IP アドレスとは異なるため)。この IP アドレスは、静的アドレスまたはリース期間が無制限の DHCP アドレスにします。
 - BMC に関連付けられている NIC の MAC アドレス。
- 6 [OK] をクリックします。

vSphere DPM 用の Wake-On-LAN のテスト

vSphere DPM 機能に対する Wake-On-LAN (WOL) の使用は、この機能を VMware ガイドラインに従って構成し、テストが成功した場合、完全にサポートされます。クラスタの vSphere DPM をはじめて有効にする前に、または、vSphere DPM を使用しているクラスタに追加された任意のホストで、次の手順を実行する必要があります。

開始する前に

WOL をテストする前に、クラスタが前提条件を満たしていることを確認します。

- クラスタに少なくとも 2 台の ESX 3.5 (または ESX 3i バージョン 3.5) 以降のホストが含まれている必要があります。
- 各ホストの vMotion ネットワーク リンクが正しく機能している必要があります。vMotion ネットワークは、ルータによって分離された複数のサブネットではなく、単一の IP サブネットであることが必要です。
- 各ホスト上の vMotion NIC は、WOL をサポートしている必要があります。WOL のサポートを確認するには、はじめに VMkernel ポートに対応する物理ネットワーク アダプタの名前を突き止めます。vSphere Client のインベントリパネルでホストを選択し、[構成] タブを選択して、[ネットワーク] をクリックします。この情報を取得したあと、[ネットワーク アダプタ] をクリックし、そのネットワーク アダプタに対応する登録情報を探します。関連するアダプタの [Wake on LAN 対応] 列が「はい」になっているはずです。
- ホスト上の各 NIC の WOL 互換ステータスを表示するには、vSphere Client のインベントリパネルでホストを選択し、[構成] タブを選択してから、[ネットワーク アダプタ] をクリックします。[Wake on LAN 対応] 列で、NIC が「はい」になっている必要があります。
- WOL 対応の各 vMotion NIC が接続されているスイッチ ポートは、リンク速度が固定速度 (1,000 Mb/s など) ではなく、自動ネゴシエートするように設定されている必要があります。多くの NIC は、ホストがパワーオフ状態のときに 100 Mb/s 以下に切り替えることができる場合のみ、WOL をサポートします。

これらの前提条件を確認したあと、vSphere DPM のサポートに WOL を使用する予定の各 ESXi ホストをテストします。これらのホストをテストするときは、vSphere DPM 機能がクラスタに対して無効になっていることを確認します。



注意 WOL を Wake プロトコルとして使用する vSphere DPM クラスタに追加されるすべてのホストをテストし、テストに合格しないホストは電力管理の使用を無効にしてください。これをしなかった場合、vSphere DPM はあとでパワーオンに戻すことができないホストをパワーオフする可能性があります。

手順

- 1 vSphere Client のホストの [サマリ] タブで、[スタンバイ モードへの切り替え] コマンドをクリックします。
この操作により、ホストはパワーオフされます。
- 2 ホストのスタンバイ モードの終了を試行するには、ホストの [サマリ] タブで [パワーオン] コマンドをクリックします。
- 3 ホストが正常にパワーオンしたかどうかを観察します。

- 4 スタンバイ モードを正常に終了できないホストがある場合、クラスタの設定ダイアログ ボックスのホスト オプション ページでそのホストを選択し、[電力管理] 設定を無効に変更します。

これを実行すると、vSphere DPM はそのホストをパワーオフする候補と見なさなくなります。

DRS クラスタ用の vSphere DPM の有効化

各ホスト上で使用している Wake プロトコルに必要な構成手順またはテスト手順を実行したあと、vSphere DPM を有効にすることができます。

電力管理自動化レベル、しきい値、およびホスト レベルのオーバーライドを構成します。これらの設定は、クラスタの設定ダイアログ ボックスの [電力管理] で構成されます。

[スケジュール設定タスク：クラスタ電源設定の変更]ウィザードを使用して、クラスタの DPM を有効または無効にするスケジュール設定タスクを作成することもできます。

注意 DRS クラスタ内のホストに USB デバイスが接続されている場合は、そのホストについて DPM を無効にする必要があります。無効にしないと、DPM によって、そのホストとサーバがオフになり、そのデバイスとそのデバイスを使用していた仮想マシンとの接続が切れることがあります。

自動化レベル

vSphere DPM によって作成されたホストの電源状態および移行の推奨が自動的に実行されるかどうかは、その機能について選択された電力管理自動化レベルによって決まります。

自動化レベルは、クラスタの設定ダイアログ ボックスの [電力管理] で構成します。

注意 電力管理自動化レベルは、DRS 自動化レベルと同じものではありません。

表 10-1. 電力管理の自動化レベル

オプション	説明
オフ	機能は無効で、推奨は作成されません。
手動	ホストの電源操作および関連する仮想マシンの移行に関する推奨が行われますが、それが自動的に実行されることはありません。これらの推奨は、vSphere Client のクラスタの [DRS] タブに表示されます。
自動	関連する仮想マシンの移行をすべて自動的に実行できる場合、ホストの電源操作は自動的に実行されます。

vSphere DPM しきい値

vSphere DPM 機能によって生成される電源状態（ホストのパワーオンまたはオフ）の推奨には、優先順位 1 の推奨から優先順位 5 の推奨までの優先順位が割り当てられます。

これらの優先順位の評価は、DRS クラスタ内で検出された高いまたは低い使用率と、ホストの電源状態の意図的な変更から予想される改善の程度に基づいています。優先順位 1 の推奨は必須である一方、優先順位 5 の推奨はわずかな改善のみをもたらします。

しきい値は、クラスタの設定ダイアログ ボックスの [電力管理] で構成します。vSphere DPM しきい値のスライダを右へ 1 レベル移動するごとに、1 つ下のレベルの優先順位を、自動的に実行される推奨セットに含めたり、または手動で実行する推奨として表示したりできます。保守の設定では、vSphere DPM は優先順位 1 の推奨のみを作成し、その右にある次のレベルは、優先順位 2 以上の推奨を作成します。以後、同様に作成し、積極的レベルは優先順位 5 以上の推奨（つまり、すべての推奨）を作成します。

注意 DRS しきい値と vSphere DPM しきい値は、基本的に独立しています。ユーザーは移行の積極性と、そのそれぞれが提供するホストの電源状態の推奨を区別できます。

ホスト レベルのオーバーライド

DRS クラスタで vSphere DPM を有効にすると、デフォルトでクラスタのすべてのホストが vSphere DPM の自動化レベルを継承します。

クラスタの設定ダイアログ ボックスのホスト オプション ページを選択し、[電力管理] 設定をクリックすることで、各ホストにあわせてこのデフォルト値をオーバーライドできます。この設定を次のオプションに変更できます。

- 無効
- 手動
- 自動

注意 スタンバイ モードの終了テストに失敗したためにホストの電力管理設定が無効になっている場合は、この設定を変更しないでください。

vSphere DPM を有効にして実行したあと、それが正しく機能しているかどうかを確認できます。確認するには、クラスタの設定ダイアログ ボックスのホスト オプション ページ、および各クラスタの [ホスト] タブに表示される、各ホストの [前回のスタンバイ終了] 情報を参照します。このフィールドには、vCenter Server が前回、ホストのスタンバイ モードを終了させようと試みたときのタイム スタンプと、それが成功または失敗のどちらであったかが表示されます。そのような試みが行われなかった場合、このフィールドには実行しないと表示されます。

注意 [前回のスタンバイ終了] フィールドの時刻は、vCenter Server イベント ログから生成されます。このログが消去された場合、時刻は実行しないにリセットされます。

vSphere DPM の監視

vCenter Server でイベント ベースのアラームを使用し、vSphere DPM を監視できます。

vSphere DPM の使用時に発生する可能性がある最も深刻なエラーは、DRS クラスタがホストの容量を必要としたときに、ホストがスタンバイ モードを終了できないことです。このエラーが発生していないかどうかを監視するために、事前構成された [スタンバイの終了エラー] アラームを vCenter Server で使用できます。vSphere DPM がホストのスタンバイ モードを終了できない場合 (vCenter Server イベント **DrsExitStandbyModeFailedEvent**)、このアラームを構成して、管理者にアラート E メールを送信するか、SNMP トラップを使用して通知を送信できます。デフォルトでは、このアラームは vCenter Server がそのホストに正常に接続できるようになると消去されます。

vSphere DPM のアクティビティを監視するために、次に示す vCenter Server イベントのアラームも作成できます。

表 10-2. vCenter Server イベント

イベント タイプ	イベント名
スタンバイ モードに切り替えています (ホストをパワーオフしようとしている)	DrsEnteringStandbyModeEvent
正常にスタンバイ モードに入りました (ホストのパワーオフが成功した)	DrsEnteredStandbyModeEvent
スタンバイ モードを終了しています (ホストをパワーオンしようとしている)	DrsExitingStandbyModeEvent
正常にスタンバイ モードを終了しました (パワーオンが成功した)	DrsExitedStandbyModeEvent

アラームの作成と編集の詳細については、『vSphere の監視とパフォーマンス』ドキュメントを参照してください。

vCenter Server 以外の監視ソフトウェアを使用しており、そのソフトウェアが物理ホストの予想外のパワーオフ時にアラームを起動する場合は、vSphere DPM がホストをスタンバイ モードにするときに、誤ったアラームが生成される可能性があります。そのようなアラームを受信しないようにする場合は、ベンダーと協力し、vCenter Server に統合したバージョンの監視ソフトウェアを導入してください。また、vCenter Server 自体を監視ソリューションとして使用することもできます。vSphere 4.x 以降では、vCenter Server は vSphere DPM を本質的に認識でき、そうした誤ったアラームを起動しないからです。

DRS アフィニティ ルールの使用

アフィニティ ルールを使用することで、クラスタ内のホストへの仮想マシンの配置を制御できます。

次の 2 つのタイプのルールを作成できます。

- 仮想マシンのグループとホストのグループとの間にアフィニティまたは非アフィニティを指定するために使用します。アフィニティ ルールは、選択した仮想マシンの DRS グループのメンバーが、指定したホストの DRS グループのメンバー上で実行できる、または実行する必要があることを指定します。非アフィニティ ルールは、選択した仮想マシンの DRS グループのメンバーが、指定したホストの DRS グループのメンバー上で実行できないことを指定します。

このタイプのルールの作成と使用については、[「仮想マシンとホスト間のアフィニティ ルール \(P. 73\)」](#) を参照してください。

- 個々の仮想マシン間のアフィニティまたは非アフィニティを指定するために使用します。アフィニティを指定したルールにより、DRS は指定された仮想マシンを、パフォーマンス上の理由などから同一ホスト上に保持しようとします。非アフィニティ ルールを使用すると、DRS は指定された仮想マシンを別々に配置しようとします。たとえば、1 台のホストに問題が発生しても、両方の仮想マシンを失うことはありません。

このタイプのルールの作成と使用については、[「仮想マシン間のアフィニティルール\(P.72\)」](#) を参照してください。

アフィニティ ルールの追加または編集時に、クラスタの現在の状態がルールに違反している場合、システムは操作を続行して違反を修正しようとします。手動、または部分的に自動化された DRS クラスタの場合、ルールの達成とロード バランシングに基づいた移行の推奨が表示され、許可が求められます。ルールを達成する必要はありませんが、ルールを達成するまで、それに対応する推奨が表示されます。

有効なアフィニティ ルールに、DRS で修正できない違反があるかどうかを確認するには、クラスタの [DRS] タブを選択し、[障害] をクリックします。現在違反があるすべてのルールについて、それに対応する障害がこのページに表示されます。障害を読んで、DRS がこの特定のルールに応じることができない理由を判断してください。ルール違反があると、ログ イベントも生成されます。

注意 仮想マシン間、および仮想マシンとホスト間のアフィニティ ルールは、個々のホストの CPU アフィニティ ルールとは異なります。

ホストの DRS グループの作成

仮想マシンとホスト間のアフィニティ ルールは、仮想マシンの DRS グループとホストの DRS グループとの間のアフィニティ（または非アフィニティ）関係を定義します。リンクするためのルールを作成する前に、これら両方のグループを作成する必要があります。

手順

- 1 vSphere Client で、インベントリにあるクラスタを右クリックして、[設定の編集] を選択します。
- 2 [vSphere DRS] の [クラスタ設定] ダイアログ ボックスの左側のペインで、[DRS グループ マネージャ] を選択します。
- 3 ホストの DRS グループ セクションで、[追加] をクリックします。
- 4 DRS グループ ダイアログ ボックスで、グループの名前を入力します。
- 5 左側のペインで、ホストを選択し、[>>] をクリックして、グループに追加します。必要なホストがすべて追加されるまで、この手順を繰り返します。
右側のペインで、グループからホストを選択し、[<<] をクリックして、削除することもできます。
- 6 [OK] をクリックします。

次に進む前に

ホストの DRS グループを使用して、仮想マシンとホスト間のアフィニティ ルールを作成できます。これにより、適切な仮想マシンの DRS グループとのアフィニティ（または非アフィニティ）関係を定義します。

[「仮想マシンの DRS グループの作成 \(P. 72\)」](#)

[「仮想マシンとホスト間のアフィニティ ルールの作成 \(P. 73\)」](#)

仮想マシンの DRS グループの作成

アフィニティ ルールは、DRS グループ間のアフィニティ（またはアンチアフィニティ）関係を規定します。リンクするためのルールを作成する前に、DRS グループを作成する必要があります。

手順

- 1 vSphere Client で、インベントリにあるクラスタを右クリックして、[設定の編集] を選択します。
- 2 [vSphere DRS] の [クラスタ設定] ダイアログ ボックスの左側のペインで、[DRS グループ マネージャ] を選択します。
- 3 仮想マシンの DRS グループ セクションで、[追加] をクリックします。
- 4 DRS グループ ダイアログ ボックスで、グループの名前を入力します。
- 5 左側のペインで、ホストを選択し、[>>] をクリックして、グループに追加します。必要なホストがすべて追加されるまで、この手順を繰り返します。
右側のペインで、グループからホストを選択し、[<<] をクリックして、削除することもできます。
- 6 [OK] をクリックします。

次に進む前に

[「ホストの DRS グループの作成 \(P. 71\)」](#)

[「仮想マシンとホスト間のアフィニティ ルールの作成 \(P. 73\)」](#)

[「仮想マシン間のアフィニティ ルールの作成 \(P. 72\)」](#)

仮想マシン間のアフィニティ ルール

仮想マシン間のアフィニティ ルールは、選択した個々の仮想マシンを同じホスト上で実行するか、別のホスト上に維持するかを指定します。このタイプのルールは、選択した個々の仮想マシン間のアフィニティまたは非アフィニティ関係を形成するのに使用します。

アフィニティ ルールが作成されると、DRS は、指定された仮想マシンを同じホスト上にまとめて維持します。パフォーマンス上の理由などで、この処理が必要になります。

非アフィニティ ルールの場合、DRS は、指定された仮想マシンを分離します。特定の仮想マシンをそれぞれ異なる物理ホスト上に常に維持する必要がある場合に、このルールを使用できます。その場合、1 台のホストに問題が発生しても、すべての仮想マシンにリスクが及ぶことはありません。

仮想マシン間のアフィニティ ルールの作成

クラスタ設定ダイアログ ボックスで仮想マシン間のアフィニティ ルールを作成し、選択した個々の仮想マシンを同じホスト上で実行するか、別々のホスト上で実行するかを指定できます。

注意 vSphere HA のフェイルオーバー ホストの指定アドミッション コントロール ポリシーを使用して、複数のフェイルオーバー ホストを指定する場合、仮想マシン間のアフィニティ ルールはサポートされません。

手順

- 1 vSphere Client で、インベントリにあるクラスタを右クリックして、[設定の編集] を選択します。

- 2 [vSphere DRS] のクラスタ設定ダイアログ ボックスの左側のペインで、[ルール] を選択します。
- 3 [追加] をクリックします。
- 4 ルール ダイアログ ボックスで、ルールの名前を入力します。
- 5 [タイプ] メニューから、[仮想マシンの包括] または [仮想マシンの分割] を選択します。
- 6 [追加] をクリックします。
- 7 ルールを適用する仮想マシンを少なくとも 2 台選択し、[OK] をクリックします。
- 8 [OK] をクリックします。

仮想マシン間のアフィニティ ルールの競合

仮想マシン間のアフィニティ ルールは複数作成して使用できますが、その場合、ルール同士が競合する可能性があります。

仮想マシン間の 2 つのアフィニティ ルールが競合する場合、両方のルールを有効にすることはできません。たとえば、1 つのルールは 2 台の仮想マシンをまとめておくと、もう 1 つのルールでは同じ 2 台の仮想マシンを分離したままにする場合、両方のルールを有効にすることはできません。適用するルールを 1 つ選択し、それと競合するルールは無効にするか、削除します。

仮想マシン間の 2 つのアフィニティ ルールが競合する場合、古い方のルールが優先され、新しいルールは無効になります。DRS は有効なルールを満たすだけで、無効なルールは無視します。DRS は、アフィニティ ルールの違反よりも、非アフィニティ ルールの違反を防ぐことを優先させます。

仮想マシンとホスト間のアフィニティ ルール

仮想マシンとホスト間のアフィニティ ルールは、選択した仮想マシン DRS グループのメンバーが、特定のホスト DRS グループのメンバー上で実行できるかどうかを指定します。

個々の仮想マシンの間にアフィニティ (または非アフィニティ) を指定する仮想マシン間のアフィニティ ルールとは異なり、仮想マシンとホスト間のアフィニティ ルールは、仮想マシンのグループとホストのグループとの間にアフィニティ関係を指定します。アフィニティ ルールには、「必須」ルール ([must]) で指定) と「必須でない」ルール ([should] で指定) があります。

仮想マシンとホスト間のアフィニティ ルールには、次のコンポーネントがあります。

- 1 つの仮想マシン DRS グループ
- 1 つのホスト DRS グループ
- ルールが必須 ([must]) か必須でない ([should]) か、アフィニティ ([run on]) か非アフィニティ ([not run on]) かの指定。

仮想マシンとホスト間のアフィニティ ルールはクラスタ ベースであるため、ルールに含まれている仮想マシンとホストが、すべて同じクラスタ内にある必要があります。仮想マシンがクラスタから削除された場合、あとでそのクラスタに戻したとしても、その仮想マシンは DRS グループの関連付けを失います。

仮想マシンとホスト間のアフィニティ ルールの作成

クラスタ設定ダイアログ ボックスで仮想マシンとホスト間のアフィニティ ルールを作成し、選択した仮想マシンの DRS グループのメンバーが、指定したホストの DRS グループのメンバー上で実行できるかどうかを指定します。

開始する前に

仮想マシンとホスト間のアフィニティ ルールを適用する、仮想マシンの DRS グループおよびホストの DRS グループを作成します。

手順

- 1 vSphere Client で、インベントリにあるクラスタを右クリックして、[設定の編集] を選択します。
- 2 vSphere DRS の [クラスタ設定] ダイアログ ボックスの左側のペインで、[ルール] を選択します。

- 3 [追加] をクリックします。
- 4 ルール ダイアログ ボックスで、ルールの名前を入力します。
- 5 [タイプ] メニューで、[仮想マシンからホストへ] を選択します。
- 6 ルールを適用する仮想マシンの DRS グループおよびホストの DRS グループを選択します。
- 7 ルールの仕様を選択します。
 - [グループ内のホスト上で実行する必要があります]。仮想マシン グループ 1 の仮想マシンはホスト グループ A のホスト上で実行する必要があります。
 - [グループ内のホスト上で実行します]。仮想マシン グループ 1 の仮想マシンはホスト グループ A のホスト上で実行しますが、それが必須ではありません。
 - [グループ内のホスト上では実行しない必要があります]。仮想マシン グループ 1 の仮想マシンはホスト グループ A のホスト上では実行しない必要があります。
 - [グループ内のホスト上では実行しません]。仮想マシン グループ 1 の仮想マシンはホスト グループ A のホスト上で実行しませんが、実行する場合もあります。
- 8 [OK] をクリックします。

仮想マシンとホスト間のアフィニティ ルールの使用

仮想マシンとホスト間のアフィニティ ルールを使用して、仮想マシンのグループとホストのグループの間にアフィニティ 関係を指定します。仮想マシンとホスト間のアフィニティ ルールを使用する際には、そのルールが最も役立つタイミング、ルール間の競合を解決する方法、および必須アフィニティ ルール設定時の注意事項の重要性を認識しておく必要があります。

仮想マシンとホスト間のアフィニティ ルールが役立つ使用事例の 1 つは、仮想マシンで実行しているソフトウェアにライセンスの制限がある場合です。そのような仮想マシンを 1 つの DRS グループに配置し、必要なライセンスを持つホストマシンだけを含むホスト DRS グループ上での実行を必須にするルールを作成できます。

注意 仮想マシンで実行しているソフトウェアのライセンス要件やハードウェア要件に基づいて、仮想マシンとホスト間のアフィニティ ルールを作成する場合は、グループが適切に設定されているようにする必要があります。ルールでは、仮想マシンで実行されているソフトウェアの監視は行いません。また、どの ESXi ホストにどのような VMware 以外のライセンスがあるかについても認識しません。

仮想マシンとホスト間のアフィニティ ルールを複数作成する場合、それらのルールのランク付けは行われず、同等に適用されます。これがルール間の相互作用に与える影響に注意してください。たとえば、1 台の仮想マシンが 2 つの DRS グループに属し、DRS グループのそれぞれに異なる必須ルールが適用されている場合、その仮想マシンは、各ルールに表されるホスト DRS グループの両方に属するホスト上でのみ実行できます。

仮想マシンとホスト間のアフィニティ ルールを作成するときに、そのルールがほかのルールに関連して機能できるかどうかはチェックされません。このため、使用しているほかのルールと競合するルールを作成してしまう可能性があります。仮想マシンとホスト間の 2 つのアフィニティ ルールが競合する場合、古い方のルールが優先され、新しいルールは無効になります。DRS は有効なルールを満たすだけで、無効なルールは無視します。

DRS、vSphere HA、および vSphere DPM は、必須アフィニティ ルール（仮想マシン DRS グループのホスト DRS グループに対する「must run on」または「must not run on」ルール）の違反につながるアクションは実行しません。クラスタの機能に悪影響が生じる可能性があるため、このタイプのルールを使用するときには注意が必要です、仮想マシンとホスト間のアフィニティ ルールを不適切に使用した場合、クラスタが断片化され、DRS、vSphere HA、および vSphere DPM の機能が阻害される可能性があります。

必須アフィニティ ルールの違反につながる場合、いくつかのクラスタ機能は実行されません。

- DRS は、ホストをメンテナンス モードにするために仮想マシンを退避させません。
- DRS は、仮想マシンのパワーオンや仮想マシンのロード バランシングを行いません。
- vSphere HA は、フェイルオーバーを実行しません。

- vSphere DPM は、ホストをスタンバイ モードにして電力管理を最適化することをしません。

これらの問題を回避するため、必須アフィニティ ルールを複数作成するときには注意してください。または、仮想マシンとホスト間の、必須ではないアフィニティ ルール（仮想マシン DRS グループのホスト DRS グループに対する「should run on」または「should not run on」ルール）を使用することを検討します。各仮想マシンが関連付けられているクラスタ内のホストの数が十分であり、1 台のホストが失われても、仮想マシンを実行できるホストがなくなることを確認します。DRS、vSphere HA、および vSphere DPM が適切に機能するように、必須でないルールに違反することは可能です。

注意 仮想マシンとホスト間のアフィニティ ルールに仮想マシンが違反したときに起動される、イベント ベースのアラームを作成できます。vSphere Client で、仮想マシン用の新しいアラームを追加し、イベント トリガーとして [仮想マシンが仮想マシン/ホストアフィニティ ルールに違反しています] を選択します。アラームの作成と編集の詳細については、『vSphere の監視とパフォーマンス』ドキュメントを参照してください。

データストア クラスタの作成

データストア クラスタは、リソースと管理インターフェイスが共有されたデータストアの集まりです。データストア クラスタとデータストアの関係は、クラスタとホストの関係と似ています。データストア クラスタを作成すると、vSphere ストレージ DRS を使用してストレージ リソースを管理できます。

注意 データストア クラスタは、vSphere API でストレージ ポッドとも呼ばれます。

データストア クラスタにデータストアを追加すると、データストアのリソースはそのデータストア クラスタのリソースの一部になります。ホストのクラスタと同様に、データストア クラスタを使用してストレージ リソースを集約することで、データストア クラスタ レベルでリソース割り当てポリシーをサポートすることができます。次のリソース管理機能も各データストア クラスタで使用できます。

領域使用率のロード バランシング	領域使用のしきい値を設定することができます。データストアの領域使用量がしきい値を超えると、ストレージ DRS によって推奨が生成されるか、Storage vMotion 移行が実行されて、データストア クラスタ全体で領域使用量のバランスが調整されます。
I/O 待ち時間のロード バランシング	I/O 待ち時間のしきい値を設定して、ボトルネックを回避できます。データストア上の I/O 待ち時間がしきい値を超えると、ストレージ DRS によって推奨が生成されるか、Storage vMotion 移行が実行されて、I/O の高負荷が軽減されます。
非アフィニティ ルール	仮想マシン ディスク用の非アフィニティ ルールを作成できます。たとえば、特定の仮想マシンの仮想ディスクは別々のデータストアに配置する必要があります。デフォルトでは、仮想マシンのすべての仮想ディスクが同じデータストアに配置されます。

この章では次のトピックについて説明します。

- [初期配置と実行中のバランシング \(P. 78\)](#)
- [ストレージの移行の推奨 \(P. 78\)](#)
- [データストア クラスタの作成 \(P. 78\)](#)
- [ストレージ DRS の有効化と無効化 \(P. 79\)](#)
- [データストア クラスタの自動化レベルの設定 \(P. 79\)](#)
- [ストレージ DRS の積極性レベルの設定 \(P. 80\)](#)
- [データストア クラスタの要件 \(P. 81\)](#)
- [データストアのデータストア クラスタへの追加と削除 \(P. 82\)](#)

初期配置と実行中のバランシング

ストレージ DRS では、初期配置と実行中のバランシングの推奨が、ストレージ DRS 対応のデータストア クラスタ内のデータストアに対して提供されます。

初期配置は、仮想マシン ディスクを配置するデータストア クラスタ内のデータストアが、ストレージ DRS で選択されるときに発生します。これは、仮想マシンが作成またはクローン作成されるとき、仮想マシン ディスクが別のデータストア クラスタに移行されるとき、または既存の仮想マシンにディスクを追加するときに発生します。

初期配置の推奨は、領域の制約に従って、領域および I/O ロード バランシングの目標を基準に作成されます。これらの目標は、1 つのデータストアのオーバー プロビジョニング、ストレージ I/O のボトルネック、および仮想マシンのパフォーマンス上の影響のリスクを最低限にすることを意図しています。

ストレージ DRS は、構成された頻度（デフォルトでは 8 時間ごと）で起動する、またはデータストア クラスタ内の 1 つ以上のデータストアがユーザー構成可能な領域使用率のしきい値を超えたときに起動します。ストレージ DRS が起動すると、各データストアの領域使用率および I/O 待ち時間の値が、しきい値と照合して確認されます。I/O 待ち時間について、ストレージ DRS では、1 日の過程で測定された I/O 待ち時間の 90 パーセントの値が、しきい値との照合に使用されます。

ストレージの移行の推奨

vCenter Server では、手動での自動化モードがあるデータストア クラスタのストレージ DRS 推奨ページに、移行の推奨が表示されます。

このシステムでは、ストレージ DRS ルールの適用と、データストア クラスタの領域と I/O リソースのバランスの調整に必要なだけの推奨が提供されます。各推奨には、仮想マシン名、仮想ディスク名、データストア クラスタ名、ソース データストア、ターゲット データストア、および推奨の理由が含まれます。

- データストアの領域使用量のバランス調整
- データストア I/O ロードのバランス調整

ストレージ DRS は、次の状況で移行に必須の推奨を行います。

- データストアの空き領域がなくなった。
- 非アフィニティまたはアフィニティ ルールに違反している。
- データストアがメンテナンス モードを開始しようとしており、回避させる必要がある。

さらに、データストアの領域がなくなりかかっている場合や、領域や I/O ロード バランシングのために調整する必要がある場合に、オプションの推奨が行われます。

ストレージ DRS では、領域のバランシングのためにパワーオフ状態またはパワーオン状態の仮想マシンの移動が検討されます。ストレージ DRS では、スナップショットがあるパワーオフ状態の仮想マシンもこれらの検討に含まれます。

データストア クラスタの作成

データストア クラスタ リソースはストレージ DRS を使用して管理できます。

手順

- 1 vSphere Client インベントリのデータストアおよびデータストア クラスタ ビューで、データ センター オブジェクトを右クリックし、[新規データストア クラスタ] を選択します。
- 2 表示される指示に従って、[データストア クラスタの作成] ウィザードを完了します。

ストレージ DRS の有効化と無効化

ストレージ DRS を使用することで、データストア クラスタの統合されたリソースを管理できます。ストレージ DRS が有効化されている場合、データストア クラスタ内のデータストア全体で領域と I/O リソースのバランスを調整するための仮想マシンのディスク配置および移行に関する推奨が提供されます。

ストレージ DRS を有効にすると、以下の機能が有効になります。

- データストア クラスタ内のデータストア間における領域のロード バランシング。
- データストア クラスタ内のデータストア間における I/O のロード バランシング。
- 領域および I/O ワークロードに基づく仮想ディスクの初期配置。

データストア クラスタ設定ダイアログ ボックスのストレージ DRS の有効化チェック ボックスで、これらすべてのコンポーネントが一度に有効化または無効化されます。必要に応じて、ストレージ DRS の I/O 関連の機能を、領域のバランシング機能とは別に無効にすることができます。

データストア クラスタ上でストレージ DRS を無効にした場合、ストレージ DRS 設定は保存されます。ストレージ DRS を有効にしたときに、データストア クラスタに対する設定は、ストレージ DRS が無効にされた時点の設定に戻ります。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
- 2 [全般] をクリックします。
- 3 [ストレージ DRS をオンにします] を選択し、[OK] をクリックします。
- 4 (オプション) ストレージ DRS の I/O 関連の機能のみを無効にするには、領域関連のコントロールを有効にしたまま、次の手順を実行します。
 - a [SDRS ランタイム ルール] を選択します。
 - b [SDRS 推奨に対して I/O メトリックを有効にします] チェック ボックスを選択解除します。
- 5 [OK] をクリックします。

データストア クラスタの自動化レベルの設定

データストア クラスタの自動化レベルによって、ストレージ DRS からの配置および移行の推奨が自動的に適用されるかどうかが決まります。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
- 2 [SDRS 自動化] を選択します。
- 3 自動化レベルを選択します。

デフォルトの自動化レベルは手動です。

オプション	説明
自動化なし (手動モード)	配置と移行の推奨が表示されますが、推奨が手動で適用されるまで実行はされません。
完全自動化	配置と移行の推奨が自動的に実行されます。

- 4 [OK] をクリックします。

ストレージ DRS の積極性レベルの設定

使用領域と I/O 待ち時間のしきい値を指定することで、ストレージ DRS の積極性が決まります。

ストレージ DRS は、データストア クラスタ内のデータストアのリソース使用情報を収集します。vCenter Server では、データストア上の仮想ディスクの配置用の推奨を生成するため、この情報が使用されます。

データストア クラスタに低い積極性レベルを設定した場合、ストレージ DRS では、明らかに必要な場合にのみ、Storage vMotion の移行が推奨されます。たとえば、I/O ロード、領域使用率、またはそれらのバランスの不均衡が高い場合です。データストア クラスタに高い積極性レベルを設定した場合、ストレージ DRS では、データストア クラスタが領域または I/O ロードのバランス調整によって利益を受けるときはいつでも、移行が推奨されます。

vSphere Client では、ストレージ DRS に対する積極性レベルの設定に、次のしきい値を使用できます。

領域使用率 vSphere Client に設定したしきい値をデータストアの領域使用率の割合が上回ったときに、ストレージ DRS で推奨が生成される、または移行が実行されます。

I/O 待ち時間 データストアについて 1 日に測定された I/O 待ち時間の 90 パーセントの値が、しきい値を上回ったときに、ストレージ DRS で推奨が生成される、または移行が実行されます。

また、詳細オプションを設定し、ストレージ DRS の積極性レベルをさらに構成することもできます。

領域使用率差 このしきい値は、ソースとターゲットで使用されている領域の差が最低限になるようにします。たとえば、データストア A で使用された領域が 82% で、データストア B が 79% の場合、差は 3 になります。しきい値が 5 の場合、ストレージ DRS はデータストア A からデータストア B への移行推奨を行いません。

I/O ロード バランシングの起動間隔 この時間の経過後、ストレージ DRS で I/O ロードの調整が実行されます。

I/O 不均衡のしきい値 この値を下げると、I/O ロード バランシングの積極性が低下します。ストレージ DRS は、0 と 1 の間の I/O 公平性のメトリックを算出します。1 は、最も公平な分散を表します。I/O ロード バランシングは、算出されたメトリックが 1 より小さい場合 (I/O 不均衡のしきい値 / 100) のみ実行されます。

ストレージ DRS ランタイム ルールの設定

ストレージ DRS トリガーを設定し、データストア クラスタの詳細オプションを構成します。

手順

- 1 (オプション) [SDRS 推奨に対して I/O メトリックを有効にします] チェック ボックスを選択または選択解除し、I/O メトリックを含めるかどうかを決めます。

このオプションを無効にすると、ストレージ DRS 推奨を行うときに vCenter Server は I/O メトリックを考慮しなくなります。このオプションを無効にすると、ストレージ DRS の以下の要素が無効になります。

- データストア クラスタ内のデータストア間における I/O のロード バランシング。
- I/O ワークロードに基づく仮想ディスクの初期配置。初期配置は、領域のみに基づきます。

2 (オプション) ストレージ DRS のしきい値を設定します。

使用領域と I/O 待ち時間のしきい値を指定することで、ストレージ DRS の積極性レベルを設定します。

- 使用されている領域のスライダを使用して、ストレージ DRS が起動されるまでに許容される消費領域の最大のパーセンテージを示します。データストアでの領域使用量がしきい値を上回ると、ストレージ DRS は、推奨を行い、移行を実行します。
- I/O 待ち時間のスライダを使用して、ストレージ DRS が起動されるまでに許容される最大 I/O 待ち時間を示します。待ち時間がしきい値を上回ると、ストレージ DRS は、推奨を行い、移行を実行します。

注意 データストア クラスタのストレージ DRS I/O 待ち時間のしきい値は、ストレージ I/O コントロールの輻輳のしきい値以下である必要があります。

3 (オプション) 詳細オプションを構成します。

- ソースとターゲットの使用率の差が次の場合、推奨が行われます。スライダを使用して、領域使用率差のしきい値を指定します。使用率は、使用量 x 100/キャパシティです。

このしきい値は、ソースとターゲットで使用されている領域の差が最低限になるようにします。たとえば、データストア A で使用された領域が 82% で、データストア B が 79% の場合、差は 3 になります。しきい値が 5 の場合、ストレージ DRS はデータストア A からデータストア B への移行推奨を行いません。

- I/O 負荷の評価間隔：ストレージ DRS が領域と I/O ロード バランシングを評価する頻度を指定します。
- I/O 不均衡のしきい値：スライダを使用して、I/O ロード バランシングの積極性を示します。この値を下げると、I/O ロード バランシングの積極性が低下します。ストレージ DRS は、0 と 1 の間の I/O 公平性のメトリックを算出します。1 は、最も公平な分散を表します。I/O ロード バランシングは、算出されたメトリックが 1 より小さい場合 (I/O 不均衡のしきい値 / 100) のみ実行されます。

4 [次へ] をクリックします。

データストア クラスタの要件

データストア クラスタ機能を正しく使用するには、データストア クラスタと関連付けられたデータストアおよびホストが、特定の要件を満たしている必要があります。

データストア クラスタを作成するときは、次のガイドラインに従ってください。

- データストア クラスタには、類似した、または交換可能なデータストアを格納する必要があります。
 - 1 つのデータストア クラスタに、サイズおよび I/O キャパシティが異なるデータストアや、異なるアレイおよびベンダーからのデータストアを混在させることができます。しかし、1 つのデータストア クラスタ内に、次のタイプのデータストアを共存させることはできません。
 - NFS および VMFS のデータストアを同じデータストア クラスタ内に組み合わせることはできません。
 - 複製されたデータストアは、同じストレージ DRS 対応のデータストア クラスタ内に、複製でないデータストアと組み合わせることはできません。
- データストア クラスタ内のデータストアに接続されるすべてのホストは、ESXi 5.0 以降である必要があります。データストア クラスタ内のデータストアが ESX/ESXi 4.x 以前のホストと接続されている場合、ストレージ DRS は実行されません。
- 複数のデータ センター全体で共有されているデータストアは、データストア クラスタに含めることはできません。
- ベスト プラクティスとして、ハードウェア アクセラレーション機能が有効化されたデータストアは、ハードウェア アクセラレーション機能が有効化されていないデータストアと同じデータストア クラスタ内に含めないでください。データストア クラスタ内のデータストアは、ハードウェア アクセラレーション機能対応の動作を保証するためには、同種のクラスタである必要があります。

データストアのデータストア クラスタへの追加と削除

既存のデータストア クラスタでのデータストアの追加および削除は、それらを vSphere Client インベントリにドラッグして行います。

データストア クラスタには、vSphere Client インベントリ内のホストにマウントされた任意のデータストアを追加できます。このとき次の例外があります。

- データストアに接続されるすべてのホストは、ESXi 5.0 以降である必要があります。
- データストアは、vSphere Client の同じインスタンス内にある複数のデータ センター内に含めることはできません。

データストア クラスタからデータストアを削除した場合、そのデータストアは vSphere Client インベントリ内に残り、ホストからアンマウントされません。

データストア クラスタを使用したストレージリソースの管理

12

データストア クラスタを作成したあとで、それをカスタマイズしたり、それを使用してストレージ I/O および領域使用率リソースを管理したりできます。

この章では次のトピックについて説明します。

- [ストレージ DRS メンテナンス モードの使用 \(P. 83\)](#)
- [ストレージ DRS 推奨の適用 \(P. 84\)](#)
- [仮想マシンのストレージ DRS 自動レベルの変更 \(P. 85\)](#)
- [ストレージ DRS の勤務時間外スケジュールの設定 \(P. 86\)](#)
- [ストレージ DRS 非アフィニティ ルール \(P. 87\)](#)
- [ストレージ DRS 統計の削除 \(P. 89\)](#)
- [ストレージ vMotion のデータストア クラスタとの互換性 \(P. 90\)](#)

ストレージ DRS メンテナンス モードの使用

データストアを使用停止にして保守作業を行うために、メンテナンス モードに切り替えます。データストアは、ユーザーの要求があった場合のみ、メンテナンス モードを開始または終了します。

ストレージ DRS 対応データストア クラスタ内のデータストアは、メンテナンス モードを使用できます。スタンドアロン データストアはメンテナンス モードにすることはできません。

メンテナンス モードになっているデータストア上の仮想ディスクは、手動またはストレージ DRS を使用して、別のデータストアに移行する必要があります。データストアをメンテナンス モードに切り替えようとする時、[配置の推奨] タブに移行の推奨リスト、仮想ディスクを移行できる同じデータストア クラスタ内のデータストアが表示されます。vCenter Server では、[障害] タブに、移行できないディスクのリストとその理由が表示されます。ストレージ DRS のアフィニティまたは非アフィニティのルールにより、ディスクを移行できない場合は、[メンテナンスで関係ルールを無視する] オプションを有効にできます。

すべての仮想ディスクが移行されるまで、データストアは [メンテナンス モードに切り替えています] の状態のままになります。

データストアをメンテナンス モードに切り替える

データストアのサービスを停止する必要がある場合は、データストアをストレージ DRS メンテナンス モードに切り替えることができます。

開始する前に

ストレージ DRS は、メンテナンス モードを開始したデータストアを含むデータストア クラスタで有効にされる。

データストアには、CD-ROM イメージ ファイルは保存されていない。

データストア クラスタに少なくとも 2 つのデータストアがある。

手順

- 1 vSphere Client のインベントリでデータストア クラスタ内のデータストアを右クリックし、[SDRS メンテナンス モードへの切り替え] を選択します。

データストアのメンテナンス モード移行に関する推奨リストが表示されます。

- 2 (オプション) [配置の推奨] タブで、適用したくない推奨を選択解除します。

注意 ディスクをすべて取り外さないと、データストアをメンテナンス モードに切り替えることはできません。推奨を選択解除した場合は、影響を受ける仮想マシンを手動で移動する必要があります。

- 3 必要に応じて、[推奨の適用] をクリックします。

vCenter Server は Storage vMotion を使用して、ソース データストアからターゲット データストアに仮想ディスクを移行します。データストアはメンテナンス モードに切り替わります。

データストアアイコンがすぐに更新されず、データストアの現在の状態が反映されないことがあります。アイコンをすぐに更新するには、[更新] をクリックします。

メンテナンス モードのためにストレージ DRS アフィニティルールを無視

ストレージ DRS アフィニティまたは非アフィニティ ルールにより、データストアがメンテナンス モードを開始できなくなる可能性があります。データストアをメンテナンス モードに切り替えるときに、これらのルールを無視できます。

データストア クラスタに対し、メンテナンスでアフィニティ ルールを無視するオプションを有効にした場合、データストアのメンテナンス モードへの切り替えを妨げるストレージ DRS のアフィニティ ルールまたは非アフィニティのルールが、vCenter Server で無視されます。

ストレージ DRS ルールは、退避の推奨についてのみ無効になります。領域およびロード バランシングの推奨や初期配置の推奨を作成するときのルールが、vCenter Server によって阻害されることはありません。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
- 2 データストア クラスタの編集ダイアログ ボックスの右側のペインで、[SDRS 自動化] を選択します。
- 3 [詳細オプション] をクリックします。
- 4 [IgnoreAffinityRulesForMaintenance] を選択します。
- 5 オプションを有効にするには、値の列に「1」と入力します。

オプションを無効にするには、「0」と入力します。

- 6 [OK] をクリックします。

メンテナンスでアフィニティ ルールを無視するオプションがデータストア クラスタに適用されます。

ストレージ DRS 推奨の適用

ストレージ DRS は、データストア クラスタ内のすべてのデータストアのリソース使用情報を収集します。ストレージ DRS はこの情報を使用して、データストア クラスタ内のデータストアへの仮想マシン ディスク配置に関する推奨を生成します。

ストレージ DRS の推奨は、vSphere Client データストア ビューの [ストレージ DRS] タブに表示されます。データストアをストレージ DRS メンテナンス モードにしようとしたときにも、推奨が表示されます。ストレージ DRS 推奨を適用すると、vCenter Server は Storage vMotion を使用して仮想マシンのディスクをデータストア クラスタ内の他のデータストアに移行して、リソースのバランスを調整します。

推奨のサブセットを適用するには、[提案された DRS 推奨のオーバーライド] チェック ボックスを選択し、各推奨を選択して適用します。

表 12-1. ストレージ DRS 推奨

ラベル	説明
優先順位	推奨の優先順位レベル (1 ~ 5)。(デフォルトでは非表示。)
推奨	ストレージ DRS が推奨するアクション。
理由	アクションが必要な理由。
移行前の領域使用率 % (ソース) および (ターゲット)	移行前にソースとターゲットのデータストアで使用される領域の割合。
移行後の領域使用率 % (ソース) および (ターゲット)	移行後にソースとターゲットのデータストアで使用される領域の割合。
移行前の I/O 待ち時間 (ソース)	移行前のソース データストア上の I/O 待ち時間の値。
移行前の I/O 待ち時間 (ターゲット)	移行前のターゲット データストア上の I/O 待ち時間の値。

ストレージ DRS 推奨の更新

ストレージ DRS の移行に関する推奨は、vSphere Client の [ストレージ DRS] タブに表示されます。これらの推奨は、ストレージ DRS を実行することで更新できます。

開始する前に

1 つ以上のデータストア クラスタが vSphere Client インベントリ内に存在する必要があります。

データストア クラスタ用のストレージ DRS を有効化します。[ストレージ DRS] タブは、ストレージ DRS が有効になっている場合のみ表示されます。

手順

- 1 vSphere Client データストア ビューで、データストア クラスタを選択して、[ストレージ DRS] タブをクリックします。
- 2 [推奨] ビューを選択し、右上隅の [ストレージ DRS の実行] リンクをクリックします。

推奨が更新されます。最終更新日タイムスタンプに、ストレージ DRS の推奨の更新時が表示されます。

仮想マシンのストレージ DRS 自動レベルの変更

データストア クラスタ全体に渡る自動化レベルは、仮想マシンごとにオーバーライドできます。デフォルトの仮想ディスク アフィニティ ルールをオーバーライドすることもできます。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
- 2 [仮想マシン設定] を選択します。
- 3 仮想マシンを選択します。
- 4 自動化レベルの列で、仮想マシンの自動化レベルを選択します。

オプション	説明
デフォルト (手動)	配置と移行の推奨が表示されますが、推奨が手動で適用されるまで実行はされません。
完全自動化	配置と移行の推奨が自動的に実行されます。
無効	vCenter Server では、仮想マシンの移行や、移行のための推奨は行われません。

- 5 [VMDK の包括] の列でチェック ボックスを選択解除して、デフォルトの VMDK アフィニティをオーバーライドします。
[\[VMDK アフィニティ ルールのオーバーライド \(P. 89\)\]](#) を参照してください。
- 6 [OK] をクリックします。

ストレージ DRS の勤務時間外スケジュールの設定

スケジュール設定タスクを作成して、データストア クラスタのストレージ DRS 設定を変更することで、完全自動化データストア クラスタの移行をできるだけオフピーク時間中に行えるようにすることができます。

スケジュール設定タスクを作成して、データストア クラスタの自動化レベルおよび積極性レベルを変更することができます。たとえば、パフォーマンスが重視される場合には、ピーク時にストレージ DRS の実行を抑制するように構成して、ストレージ移行の発生を最小限に抑えることができます。ピーク時以外には、ストレージ DRS をより積極的に実行し、より頻繁に起動されるようにすることができます。

開始する前に

ストレージ DRS を有効にします。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
 - 2 データストア クラスタの編集ダイアログ ボックスで、[SDRS スケジュール設定] をクリックします。
 - 3 [追加] をクリックします。
 - 4 タスクを実行する時刻を入力して日にちを選択します。
 - 5 [次へ] をクリックします。
 - 6 タスク用の開始設定を指定します。
 - a 開始設定の説明を入力します。
 たとえば、「**Change SDRS Configuration**」と入力します。
 - b 自動化レベルを選択します。
 - c ストレージ DRS 推奨の I/O メトリックを無効にするには、このチェック ボックスをオンにします。
 ストレージ DRS 推奨で I/O メトリックを無効にすると、I/O メトリックはストレージ DRS 推奨の一部またはデータストア クラスタの自動移行としてみなされなくなります。
 - d 使用領域のしきい値を設定します。
 使用されている領域のスライダを使用して、ストレージ DRS が起動されるまでに許容される消費領域の最大のパーセンテージを示します。データストアでの領域使用量がしきい値を上回ると、ストレージ DRS は、推奨を行い、移行を実行します。
 - e I/O 待ち時間のしきい値を設定します。
 I/O 待ち時間のスライダを使用して、ストレージ DRS が起動されるまでに許容される最大 I/O 待ち時間を示します。待ち時間がしきい値を上回ると、ストレージ DRS は、推奨を行い、移行を実行します。

注意 データストア クラスタのストレージ DRS I/O 待ち時間のしきい値は、ストレージ I/O コントロールの輻輳のしきい値以下である必要があります。

 - f I/O 不均衡のしきい値を設定します。
 I/O 不均衡のしきい値のスライダを使用して、I/O ロード バランシングの積極性を示します。I/O 負荷の不均衡のレベルがしきい値を上回ると、ストレージ DRS は、推奨を行い、移行を実行します。
- 7 [次へ] をクリックします。

- 8 タスク用の終了設定を指定します。
- ストレージ DRS 設定をタスク前の構成にリストアするには、[設定のリストア] チェック ボックスをオンにします。
 - タスク前の構成以外の設定を指定する場合は、[設定のリストア] チェック ボックスをオフにします。
- 9 終了準備の完了ページの内容を確認して、[終了] をクリックします。

指定した時間に、スケジュール設定タスクが実行されます。

ストレージ DRS 非アフィニティ ルール

ストレージ DRS 非アフィニティ ルールを作成して、データストア クラスタ内の同じデータストアに配置すべきではない仮想ディスクを指定することができます。デフォルトでは、仮想マシンの仮想ディスクはすべて、同じデータストア内で格納されます。

非アフィニティ ルールを作成すると、データストア クラスタ内の関連する仮想ディスクに適用されます。非アフィニティ ルールは初期配置およびストレージ DRS 推奨による移行の際に適用されますが、移行がユーザーによって開始される場合は適用されません。

注意 非アフィニティ ルールは、データストア クラスタ内のデータストアに保存された CD-ROM ISO イメージ ファイルには適用されず、ユーザー 定義の場所に保存されたスワップファイルにも適用されません。

仮想マシン間の非アフィニティ ルール どの仮想マシンを同じデータストアに格納できないようにするかを指定します。「[仮想マシン間非アフィニティ ルールの作成 \(P. 88\)](#)」を参照してください。

仮想マシン内の非アフィニティ ルール 特定の仮想マシンに関連付けられた仮想ディスクのうち、どの仮想ディスクを別のデータストアに格納すべきかを指定します。「[仮想マシン内非アフィニティ ルールの作成 \(P. 88\)](#)」を参照してください。

仮想ディスクをデータストア クラスタから除外すると、アフィニティ ルールまたは非アフィニティ ルールはそのディスクに適用されなくなります。

既存のアフィニティ ルールまたは非アフィニティ ルールのあるデータストア クラスタに仮想ディスク ファイルを移動する場合、以下の動作が適用されます。

- データストア クラスタ B に仮想マシン内のアフィニティ ルールがある場合。仮想ディスクをデータストア クラスタ A からデータストア クラスタ B に移動すると、データストア クラスタ A 内の特定の仮想マシンの仮想ディスクに適用されていたルールは適用されなくなります。仮想ディスクは、データストア クラスタ B 内の仮想マシン内のアフィニティ ルールの対象となります。
- データストア クラスタ B に仮想マシン間の非アフィニティ ルールがある場合。仮想ディスクをデータストア クラスタ A からデータストア クラスタ B に移動すると、データストア クラスタ A 内の特定の仮想マシンの仮想ディスクに適用されていたルールは適用されなくなります。仮想ディスクは、データストア クラスタ B 内の仮想マシン間の非アフィニティ ルールの対象となります。
- データストア クラスタ B に仮想マシン内の非アフィニティ ルールがある場合。仮想ディスクをデータストア クラスタ A からデータストア クラスタ B に移動すると、データストア クラスタ B 内の仮想マシン内の非アフィニティ ルールは特定の仮想マシンの仮想ディスクに適用されません。これは、ルールがデータストア クラスタ B 内の指定の仮想ディスクのみを対象とするためです。

注意 ストレージ DRS ルールにより、データストアがメンテナンス モードを開始できなくなる可能性があります。[メンテナンス] オプションに対して [アフィニティ ルールの無視] を有効にすると、メンテナンス モードのためにストレージ DRS ルールを無視できます。

仮想マシン間非アフィニティ ルールの作成

非アフィニティ ルールを作成して、特定の仮想マシンのすべての仮想ディスクは別々のデータストアに存在する必要があることを示すことができます。このルールは個々のデータストア クラスタに適用されます。

データストア クラスタ内の仮想マシン間非アフィニティ ルールに参加する仮想マシンを、データストア クラスタの仮想マシン内アフィニティ ルールと関連付ける必要があります。仮想マシンは、仮想マシン内アフィニティ ルールにも準拠する必要があります。

仮想マシンが仮想マシン間非アフィニティ ルールの対象となる場合は、次の動作が適用されます。

- ストレージ DRS は、仮想マシンの仮想ディスクをルールに従って配置します。
- ストレージ DRS は、データストアをメンテナンス モードにするなどの強制的な理由で移行する場合でも、ルールに従って vMotion を使用して仮想ディスクを移行します。
- 仮想マシンの仮想ディスクがルールに違反している場合、ストレージ DRS は移行推奨を行ってエラーを修正します。推奨を行ってエラーを修正できない場合は、その違反をエラーとして報告します。

デフォルトでは、仮想マシン間非アフィニティ ルールは定義されません。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
- 2 [データストア クラスタの編集] ダイアログ ボックスの左側のペインで、[ルール] を選択します。
- 3 [追加] をクリックします。
- 4 ルールの名前を入力します。
- 5 [タイプ] メニューで [仮想マシン非アフィニティ] を選択します。
- 6 [追加] をクリックします。
- 7 [仮想マシンの選択] をクリックします。
- 8 少なくとも 2 台の仮想マシンを選択し、[OK] をクリックします。
- 9 [OK] をクリックしてルールを保存します。

仮想マシン内非アフィニティ ルールの作成

仮想マシンの VMDK 非アフィニティ ルールを作成して、どの仮想ディスクを別のデータストアに格納する必要があるかを示すことができます。

VMDK 非アフィニティ ルールは、ルールが定義された仮想マシンに適用されます。すべての仮想マシンに適用されるわけではありません。ルールは、互いに分離する仮想ディスクのリストとして表示されます。

1 台の仮想マシンに対して仮想マシン内非アフィニティ ルールと仮想マシン内アフィニティ ルールを設定しようとすると、vCenter Server で最後に定義されたルールが拒否されます。

仮想マシンが VMDK 非アフィニティ ルールの対象となる場合は、次の動作が適用されます。

- ストレージ DRS は、仮想マシンの仮想ディスクをルールに従って配置します。
- ストレージ DRS は、データストアをメンテナンス モードにするなどの強制的な理由で移行する場合でも、ルールに従って vMotion を使用して仮想ディスクを移行します。
- 仮想マシンの仮想ディスクがルールに違反している場合、ストレージ DRS は移行推奨を行ってエラーを修正します。推奨を行ってエラーを修正できない場合は、その違反をエラーとして報告します。

デフォルトでは、仮想マシン内非アフィニティ ルールは定義されません。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
- 2 [データストア クラスタの編集] ダイアログ ボックスの左側のペインで、[ルール] を選択します。
- 3 [追加] をクリックします。
- 4 ルールの名前を入力します。
- 5 [タイプ] メニューで [VMDK 非アフィニティ] を選択します。
- 6 [追加] をクリックします。
- 7 [仮想マシンの選択] をクリックします。
- 8 仮想マシンを選択し、[OK] をクリックします。
- 9 ルールが適用される仮想ディスクを少なくとも 2 つ選択し、[OK] をクリックします。
- 10 [OK] をクリックしてルールを保存します。

VMDK アフィニティ ルールのオーバーライド

VMDK アフィニティ ルールは、特定の仮想マシンと関連付けられているデータストア クラスタ内のすべての仮想ディスクが、データストア クラスタ内の同じデータストアに置かれていることを示します。このルールは個々のデータストア クラスタに適用されます。

VMDK アフィニティ ルールは、データストア クラスタ内のすべての仮想マシンに対してデフォルトで有効になっています。データストア クラスタまたは個々の仮想マシンに対するこのデフォルト設定をオーバーライドできます。

VMDK アフィニティ ルールの適用対象となる仮想マシンは、次のように動作します。

- ストレージ DRS は、仮想マシンの仮想ディスクをルールに従って配置します。
- ストレージ DRS は、データストアをメンテナンス モードにするなどの強制的な理由で移行する場合でも、ルールに従って vMotion を使用して仮想ディスクを移行します。
- 仮想マシンの仮想ディスクがルールに違反している場合、ストレージ DRS は移行推奨を行ってエラーを修正します。推奨を行ってエラーを修正できない場合は、その違反をエラーとして報告します。

ストレージ DRS 対応のデータストア クラスタにデータストアを追加すると、そのデータストア上に仮想ディスクを持つすべての仮想マシンで、他のデータストアにも仮想ディスクがある場合は、VMDK アフィニティ ルールが無効にされます。

手順

- 1 vSphere Client のインベントリでデータストア クラスタを右クリックし、[設定の編集] を選択します。
- 2 [仮想マシン設定] をクリックします。
- 3 仮想マシンの [VMDK の包括] チェック ボックスを選択解除します。
- 4 [OK] をクリックします。

ストレージ DRS 統計の削除

ストレージ DRS の問題を診断するには、手動でストレージ DRS を実行する前に、ストレージ DRS 統計情報を削除できます。

重要 ストレージ DRS 統計情報を削除するオプションを有効化した場合、このオプションを無効にしないかぎり、統計情報はストレージ DRS が実行されるたびに削除されます。ストレージ DRS の問題の診断後は、常にこのオプションを無効にしてください。

開始する前に

データストア クラスタ用のストレージ DRS を有効化します。

手順

- 1 **ClearIoStatsOnSdrsRun** オプションを有効にします。
 - a vSphere Client でデータストア クラスタを右クリックし、[設定の編集] を選択します。
 - b [SDRS 自動化レベル] を選択し、[詳細オプション] をクリックします。
 - c オプションテキスト ボックスに、「**ClearIoStatsOnSdrsRun**」 と入力します。
 - d 対応する値テキスト ボックスに、「**1**」 と入力します。
 - e [OK] をクリックし、[OK] を再度クリックして設定ダイアログ ボックスを閉じます。
- 2 vSphere Client のインベントリで、データストア クラスタを選択します。
- 3 [ストレージ DRS] タブをクリックし、[DRS の実行] をページの右上隅で選択します。
vSphere Client インベントリ内のすべてのデータストア クラスタにあるすべてのデータストアおよび仮想ディスクについての現在のストレージ DRS 統計情報が削除されますが、新しい統計情報は収集されません。
- 4 **ClearIoStatsOnSdrsRun** フラグ値を **0** に変更して無効にします。
- 5 ストレージ DRS を再度実行します。
ストレージ DRS が通常どおりに実行されます。新しい設定が有効になるまで、数時間かかります。

ストレージ vMotion のデータストア クラスタとの互換性

データストア クラスタには、特定の vSphere Storage vMotion[®] 要件があります。

- ホストは Storage vMotion をサポートしているバージョンの ESXi を実行する必要があります。
- ホストには、ソース データストアとターゲット データストアの両方に書き込みアクセスがある必要があります。
- ホストに Storage vMotion に対応するだけの十分な空きメモリ リソースがある必要があります。
- ターゲット データストアに十分なディスク領域がある必要があります。
- ターゲット データストアがメンテナンス モードである、またはメンテナンス モードを開始するところではありません。

NUMA システムと ESXi の併用

ESXi は、NUMA (Non-Uniform Memory Access) をサポートするサーバアーキテクチャで、Intel と AMD Opteron のプロセッサのメモリ アクセス最適化をサポートします。

ESXi における NUMA のスケジュール設定の実行方法と VMware NUMA アルゴリズムの機能を理解すれば、NUMA 制御を指定して、仮想マシンのパフォーマンスを最適化できます。

この章では次のトピックについて説明します。

- [NUMA とは \(P. 91\)](#)
- [ESXi の NUMA スケジュール設定の機能 \(P. 92\)](#)
- [VMware NUMA 最適化のアルゴリズムと設定 \(P. 93\)](#)
- [NUMA アーキテクチャでのリソース管理 \(P. 94\)](#)
- [仮想 NUMA の使用 \(P. 94\)](#)
- [NUMA 制御の指定 \(P. 96\)](#)

NUMA とは

NUMA システムは、複数のシステムバスを持つ高度なサーバプラットフォームです。単一のシステムイメージ内で多数のプロセッサを利用でき、コストパフォーマンスに優れています。

この 10 年間、プロセッサのクロック速度は劇的に上昇しました。しかし、マルチギガヘルツ CPU は、その処理能力を効果的に使用するために、大量のメモリバンド幅とともに提供されなければなりません。科学計算アプリケーションなどのメモリ集約型のワークロードを実行する単一 CPU でさえ、メモリバンド幅によって制約される場合があります。

この問題は、対称型マルチプロセッシング (SMP) システムで深刻になり、同じシステムバス上で多くのプロセッサがバンド幅を奪い合うこととなります。一部のハイエンドシステムでは、高速データバスの構築によってこの問題を解決しようとしています。しかし、そのような解決策は高価であり、スケーラビリティが制限されます。

NUMA は、高性能接続を使用してコスト効率の高いいくつかの小型のノードをリンクする別の方法です。各ノードには、小型の SMP システムのように、プロセッサとメモリが含まれます。ただし、高度なメモリコントローラにより、ノードはほかのすべてのノード上のメモリを使用でき、単一のシステムイメージを作成します。プロセッサ自身のノード内にはメモリ (リモートメモリ) にプロセッサがアクセスする場合、ローカルメモリのアクセスよりも低速な NUMA 接続を通じてデータを転送する必要があります。テクノロジーの名前が示唆するように、メモリアクセスの時間は一定ではなく、メモリの場所およびメモリにアクセスするノードによって異なります。

オペレーティングシステムについての課題

NUMA アーキテクチャは、単一のシステムイメージを提供するので、特別な最適化を行わずにオペレーティングシステムを実行できることがよくあります。

リモート メモリ アクセスの遅延が大きいと、データがローカル ノードに転送されるのを常に待つため、プロセッサの使用率が低いままになることがあり、メモリ バンド幅の要求の高いアプリケーションにとって NUMA 接続がボトルネックになることがあります。

さらに、そのようなシステムでのパフォーマンスは、変動が大きくなる場合があります。たとえば、1 回のベンチマーク実行においてローカルに配置されたメモリがアプリケーションにあるが、以後の実行ではそのすべてのメモリがリモート ノード上に配置される場合、パフォーマンスが変動します。この現象により、キャパシティ プランニングが困難になることがあります。

一部のハイエンド UNIX システムは、コンパイラとプログラミング ライブラリ内で NUMA 最適化のサポートを提供します。このサポートでは、ソフトウェア開発者が、最適なパフォーマンスのためにプログラムをチューニングおよび再コンパイルする必要があります。1 つのシステムを最適化しても、同じシステムの次の世代において適切に機能することは保証されません。ほかのシステムでは、アプリケーションを実行するノードを管理者が明示的に決定することが許可されています。これは、メモリの 100 パーセントがローカルであることを要求する特定のアプリケーションにとっては許容可能かもしれませんが、管理の負担が発生し、ワークロードが変化した場合にノード間で不均衡が発生することがあります。

理想的なのは、修正を行うことなくただちにアプリケーションが利益を得られるように、システム ソフトウェアが透過的な NUMA サポートを提供することです。システムは、管理者が常に介入する必要がなく、ローカル メモリを最大限に使用し、プログラムをインテリジェントにスケジュール設定する必要があります。最後に、公平性またはパフォーマンスを阻害することなく、状態の変化に適切に対応する必要があります。

ESXi の NUMA スケジュール設定の機能

ESXi は、高度な NUMA スケジューラを使用して、プロセッサの負荷とメモリの局所性またはプロセッサの負荷バランスを動的に均衡させます。

- 1 NUMA スケジューラによって管理される各仮想マシンは、ホーム ノードに割り当てられます。ホーム ノードは、システム リソース割り当てテーブル (SRAT) に示されたとおりにプロセッサとローカル メモリを含む、システムの NUMA ノードの 1 つです。
- 2 メモリが仮想マシンに割り当てられる場合、ESXi ホストは、ホーム ノードから優先的にメモリを割り当てます。仮想マシンの仮想 CPU は、メモリの局所性を最大限にするため、ホーム ノード上での実行に制限されます。
- 3 NUMA スケジューラは、システム負荷の変化に対応するために、仮想マシンのホーム ノードを動的に変更できません。NUMA スケジューラは、プロセッサの負荷不均衡を減らすために、新しいホーム ノードに仮想マシンを移行することがあります。このためにより多くのメモリがリモートになる場合があるので、NUMA スケジューラは、仮想マシンのメモリを動的に新しいホーム ノードに移行して、メモリの局所性を向上させることがあります。NUMA スケジューラは、ノード間で仮想マシンをスワップすると全体的なメモリの局所性が高まる場合、そのようなスワップを行うこともあります。

一部の仮想マシンは、ESXi の NUMA スケジューラによって管理されません。たとえば、仮想マシンのプロセッサまたはメモリ アフィニティを手動で設定する場合、NUMA スケジューラがその仮想マシンを管理できないことがあります。NUMA スケジューラによって管理されない仮想マシンであっても、正常に動作します。ただし、それらの仮想マシンは ESXi の NUMA 最適化の利点を得られません。

ESXi における NUMA のスケジュール設定とメモリ配置のポリシーは、すべての仮想マシンを透過的に管理できるため、管理者は、ノード間で仮想マシンのバランスをとることの複雑さを明示的に処理する必要はありません。

最適化は、ゲスト OS のタイプにかかわらずにシームレスに機能します。ESXi は、NUMA ハードウェアをサポートしない仮想マシン (Windows NT 4.0 など) にも、NUMA サポートを提供します。結果的に、レガシー オペレーティングシステムであっても新しいハードウェアを利用できます。

1 つのハードウェア ノード上で使用可能な物理プロセッサ コアの数よりも多くの仮想プロセッサを持つ仮想マシンは、自動的に管理できます。NUMA スケジューラは、複数の NUMA ノードを利用することで、そのような仮想マシンに対応します。つまり、そのような仮想マシンを複数の NUMA クライアントに分割し、それぞれの NUMA クライアントに 1 つのノードに割り当ててから、通常の、分割されていないクライアントとしてスケジューラが管理します。これにより、局所性の高い、メモリを多用するワークロードのパフォーマンスを強化できます。この機能の動作の構成については、[「仮想マシンの詳細属性 \(P. 102\)」](#) を参照してください。

ESXi 5.0 では、仮想 NUMA トポロジのゲスト OS への公開のサポートが導入されました。仮想 NUMA 制御の詳細については、[「仮想 NUMA の使用 \(P. 94\)」](#) を参照してください。

VMware NUMA 最適化のアルゴリズムと設定

このセクションでは、アプリケーションのパフォーマンスを最大化しながらリソース保証を維持するために ESXi が使用するアルゴリズムと設定について説明します。

ホーム ノードと初期配置

仮想マシンがパワーオンされると、ESXi がその仮想マシンにホーム ノードを割り当てます。仮想マシンはホーム ノード内のプロセッサ上でのみ実行され、新しく割り当てられるメモリもホーム ノードから得られます。

仮想マシンのホーム ノードが変化しないかぎり、仮想マシンはローカル メモリだけを使用して、ほかの NUMA ノードへのリモート メモリ アクセスに伴うパフォーマンス上のペナルティを回避します。

仮想マシンがパワーオンされると初期ホーム ノードが割り当てられ、NUMA ノード間の CPU とメモリ負荷全体のバランスが保たれます。大規模な NUMA システムではノード間の待ち時間に大きな幅があるので、ESXi は起動時にノード間の待ち時間を確認し、単一の NUMA ノードよりワイドな仮想マシンを初期配置するときにこの情報を使用します。これらのワイド仮想マシンは、メモリ アクセスの待ち時間を最小限にするために、たがいに近い NUMA ノードに配置されます。

初期配置だけを考慮する方法でも、ベンチマーク構成などのように、システムが実行中のあいだ変化しない、1 つのワークロードのみを実行するシステムには通常、十分です。しかし、この方法では、ワークロードが変化するデータセンタークラスのシステムに対して、良好なパフォーマンスと公平さを保証できません。そのため、初期配置に加え、ESXi 5.0 では、NUMA ノード間での仮想 CPU とメモリの動的移行を行い、CPU バランスの向上とメモリの局所性増加を図っています。

動的ロード バランシングとページの移行

ESXi は、従来の初期配置方法を動的な再バランシング アルゴリズムと統合します。システムは、定期的に（デフォルトでは 2 秒ごとに）さまざまなノードの負荷を調べて、仮想マシンをあるノードから別のノードに移動することによって負荷を再バランシングするべきかどうかを判定します。

公平性またはリソース資格に違反しないでパフォーマンスを向上させるために、この計算では、仮想マシンとリソースプールのリソース設定が考慮されます。

リバランサは、適切な仮想マシンを選択して、そのホーム ノードを負荷が最小のノードに変更します。可能な場合、リバランサは、すでに一部のメモリがターゲット ノード上に配置されている仮想マシンを移動します。そこから先は（仮想マシンが再び移動されないかぎり）、仮想マシンがその新しいホーム ノード上にメモリを割り当てて、新しいホーム ノード内のプロセッサ上でのみ実行されます。

再バランシングは、公平性を維持して、すべてのノードが完全に使用されるようにするための効果的な解決策です。リバランサは、メモリをほとんどまたはまったく割り当てなかったノードに仮想マシンを移動しなければならない場合があります。その場合、仮想マシンは、大量のリモート メモリ アクセスに伴うパフォーマンス上のペナルティを被ります。ESXi は、仮想マシンの元のノードから新しいホーム ノードへメモリを透過に移行することによって、このペナルティを排除できます。

- 1 システムは、元のノード上のページ（連続するメモリの 4KB）を選択し、そのデータをターゲット ノード内のページにコピーします。
- 2 システムは、ターゲット ノード上のページを以後のすべての参照に使用してリモート メモリ アクセスのペナルティを排除するように、仮想マシン監視レイヤーとプロセッサのメモリ管理ハードウェアを使用して仮想マシンのメモリビューをシームレスに再マッピングします。

仮想マシンが新しいノードに移動するとき、ESXi ホストはすぐに、この方法によるメモリの移行を開始します。仮想マシンのリモート メモリの残りがほとんどない場合やターゲット ノードの使用可能な空きメモリがほとんどない場合は特に、ESX/ESXi ホストは、この速度を管理してシステムに過度な負担をかけないようにします。このメモリ移行アルゴリズムは、仮想マシンが短期間だけ新しいノードに移動される場合は ESXi ホストがメモリを不必要に移動しないことも保証します。

初期配置、動的再バランシング、およびインテリジェントなメモリ移行が共同で動作する場合、ワークロードの変化があっても、NUMA システム上での良好なメモリ パフォーマンスがそれらの動作によって保証されます。主要なワークロードの変化が発生した場合、たとえば、新しい仮想マシンが起動された場合、システムは再調整に時間をかけて、仮想マシンとメモリを新しい場所に移行します。短期間（通常は数秒または数分）でシステムは再調整を完了し、安定状態に達します。

NUMA 用に最適化された透過的なページ共有

仮想マシン間でメモリを共有する機会を提供する ESXi ワークロードは、いくつもあります。

たとえば、いくつかの仮想マシンが、同じゲスト OS のインスタンスを実行していたり、同じアプリケーションまたはコンポーネントをロードしていたり、共通のデータを含んでいたりとすることがあります。このような場合、ESXi システムは、専用の透過的なページ共有技法を使用してメモリ ページの冗長コピーを安全に削除します。メモリの共有により、仮想マシンで実行しているワークロードが消費するメモリが、物理マシン上で実行する場合の消費量よりも少なくなることがよくあります。結果的に、より高いレベルのオーバーコミットを効率的にサポートできます。

ESXi システムの透過的なページ共有も、NUMA システム上で使用するために最適化されています。NUMA システムでは、ページはノードごとに共有されるので、各 NUMA ノードは、数多く共有されたページの独自のローカル コピーを持ちます。仮想マシンは、共有ページを使用する場合、リモート メモリにアクセスする必要がありません。

注意 デフォルトの動作は、以前の ESX および ESXi のすべてのバージョンと同じです。

NUMA アーキテクチャでのリソース管理

さまざまなタイプの NUMA アーキテクチャでリソース管理を行うことができます。

マルチコア システムが広く普及したことにより、NUMA アーキテクチャの人気は一層高まりつつあります。このアーキテクチャが、メモリを多用するワークロードのパフォーマンスを強化できることがその理由です。最近の Intel および AMD のシステムには、いずれも NUMA サポートがプロセッサに組み込まれています。さらに、IBM Enterprise X アーキテクチャなどの従来の NUMA システムもあり、これらでは Intel および AMD のプロセッサが、専用のチップセットサポートを備えた NUMA の動作で拡張されます。

通常、NUMA の動作は BIOS 設定を使用して有効または無効にできます。たとえば、AMD Opteron ベースの HP ProLiant サーバでは、BIOS のノード インタリーピングを有効にすることによって、NUMA を無効にできます。NUMA が有効な場合、BIOS でシステム リソース割り当てテーブル (SRAT) が構築され、このテーブルは、最適化に使用される NUMA 情報を ESXi が生成するために使用されます。スケジュール設定の公平性を確保するため、NUMA の最適化は、NUMA ノードあたりのコア数が少なすぎる、または全体のコア数が少なすぎるシステムに対しては有効になりません。この動作は、`numa.rebalancecorestotal` と `numa.rebalancecoresnode` の各オプションを使用して変更できます。

仮想 NUMA の使用

vSphere 5.0 では、仮想 NUMA トポロジのゲスト OS への公開のサポートが導入されました。これにより、ゲスト OS の活用と NUMA の最適化の適用を介してパフォーマンスを改善できます。

仮想 NUMA トポロジは、ハードウェアバージョン 8 仮想マシンに使用でき、仮想 CPU の数が 8 を超えた場合にデフォルトで有効化されます。また、詳細構成オプションを使用することで、仮想 NUMA トポロジを手動で調整できます。

仮想 NUMA トポロジは、vSphere Client の 2 つの設定を使用して調整できます。これらは仮想ソケット数と、仮想マシンに対するソケットあたりのコア数です。ソケットあたりのコア数 (`cpuid.coresPerSocket`) が 1 を超え、仮想マシン内の仮想コア数が 8 を超える場合、仮想 NUMA ノードのサイズは、仮想ソケット サイズと一致します。ソケットあたりのコア数が 1 以下の場合、仮想 NUMA ノードは、仮想マシンがパワーオンされる最初の物理ホストのトポロジと一致するように作成されます。

仮想 CPU の数とメモリの使用量が比例して増大する場合、デフォルトの値を使用できます。非常に大量のメモリを消費する仮想マシンの場合、次のいずれかの方法でデフォルトの値をオーバーライドできます。

- 仮想 CPU の数を、その数の仮想 CPU が使用されていなくても増やします。[\[仮想 CPU 数の変更 \(P. 95\)\]](#) を参照してください。

- 仮想 NUMA トポロジと、物理 NUMA トポロジへのマッピングを制御する詳細オプションを使用します。[[仮想 NUMA 制御 \(P.95\)](#)] を参照してください。

仮想 CPU 数の変更

ESXi ホスト上で実行される仮想マシンは、仮想 CPU を最大 32 個まで搭載するように構成できます。

重要 仮想マシンでマルチコア仮想 CPU 設定を構成する場合は、構成がゲスト OS EULA の要件に準拠するようにしてください。

手順

- 1 vSphere Client で、インベントリにある仮想マシンを右クリックして、[設定の編集] を選択します。
- 2 [ハードウェア] タブをクリックして、[CPU] を選択します。
- 3 [仮想ソケット数] ドロップダウン メニューから値を選択します。
- 4 [ソケットあたりのコアの数] ドロップダウン メニューから値を選択します。
算出されたコアの総数は、ホスト上の論理 CPU の数以下になります。
- 5 [OK] をクリックします。

仮想 NUMA 制御

非常にメモリ消費量が多い仮想マシンに対し、詳細オプションを使用して手動でデフォルトの仮想 CPU 設定をオーバーライドできます。

これらの詳細オプションは、仮想マシン構成ファイルに追加できます。

表 13-1. 仮想 NUMA 制御のための詳細オプション

オプション	説明	デフォルト値
<code>cpuid.coresPerSocket</code>	仮想 CPU ソケットあたりの仮想コア数の決定。この値が 2 以上の場合、仮想マシンに仮想 NUMA トポロジがあれば、仮想 NUMA ノードのサイズも決定します。各物理ホストの仮想 NUMA トポロジを正確に知っている場合、このオプションを設定できません。	1
<code>numa.vcpu.maxPerVirtualNode</code>	<code>cpuid.coresPerSocket</code> は 2 のべき乗で制約が強すぎる場合、直接、 <code>numa.vcpu.maxPerVirtualNode</code> を設定できます。このケースでは、 <code>cpuid.coresPerSocket</code> は設定しません。	8
<code>numa.autosize</code>	このオプションを設定するとき、仮想 NUMA トポロジには、仮想ノードあたり、各物理ノードのコア数と同数の仮想 CPU があります。	FALSE

表 13-1. 仮想 NUMA 制御のための詳細オプション (続き)

オプション	説明	デフォルト値
<code>numa.autosize.once</code>	これらの設定で仮想マシンのテンプレートを作成した場合、その後仮想マシンをパワーオンするたびに、設定は同じであることが保証されます。仮想 NUMA トポロジは、仮想マシンの仮想 CPU の構成数が増えたり減ったりすると再評価されます。	TRUE
<code>numa.vcpu.min</code>	仮想 NUMA トポロジを生成するために必要な最小限の、仮想マシン内の仮想 CPU 数。	9

注意 `numa.autosize` を TRUE に設定し、構成が手動またはスクリプトで設定されている場合、ゲストの一部に、動的な変更を処理できないものがある可能性があります。

たとえば、`numactl` システムユーティリティで構成された Linux アプリケーションが設定されており、ノードあたり 4 個のコアがある 1 台の物理ホスト上でテストされたとします。このホストは、8 個の仮想 CPU がある 1 台の仮想マシンのために、2 つの NUMA ノードが必要です。同じ仮想マシンが、ノードあたり 8 個のコアがあるシステム上で実行される場合、これは vMotion 操作中に起きる可能性があり、`numa.autosize` が TRUE に設定されていると、1 つの仮想 NUMA ノードのみが作成されます (2 つの仮想 NUMA ノードではない)。`numactl` が 2 番目の仮想 NUMA ノードを参照している場合、その操作は失敗します。

これを回避するには、スクリプトはまずクエリ `numactl --hardware` を実行する能力が必要です。または、NUMA トポロジを明示的に設定するか、デフォルトの `numa.autosize.once` 設定を有効にする必要があります。

NUMA 制御の指定

メモリを大量に使用するアプリケーションがある場合、または仮想マシンの台数が少ない場合、仮想マシンの CPU とメモリの配置を明示的に指定することによってパフォーマンスを最適化できます。

コントロールを指定することは、仮想マシンがメモリ集約型のワークロード (メモリ内データベース、または大きいデータセットを持つ科学計算アプリケーションなど) を実行する場合に役に立ちます。システムのワークロードが単純で変化しないことがわかっている場合も、NUMA の配置を手動で最適化できます。たとえば、同様のワークロードを持つ 8 台の仮想マシンを実行する 8 プロセッサシステムは、明示的に簡単に最適化できます。

注意 ほとんどの状況において、ESXi ホストの自動 NUMA 最適化で良好なパフォーマンスが得られます。

ESXi には NUMA 配置用のコントロールが 3 セット備わっているため、管理者が仮想マシンのメモリとプロセッサの配置を操作できます。

vSphere Client では、次のオプションを指定できます。

NUMA ノードのアフィニティ このオプションを設定した場合、アフィニティに指定したノード上でのみ、NUMA は仮想マシンをスケジュール設定できます。

CPU アフィニティ このオプションを設定した場合、仮想マシンは、アフィニティに指定したプロセッサのみを使用します。

メモリ アフィニティ このオプションを設定した場合、サーバは指定したノード上でのみメモリを割り当てます。

NUMA ノードアフィニティを指定した場合にも、仮想マシンは NUMA で管理されますが、そのマシンの仮想 CPU は、NUMA ノードアフィニティに指定したノード上でのみスケジュール設定できます。同様に、NUMA ノードアフィニティに指定したノードからのみ、メモリを取得できます。CPU またはメモリアフィニティを指定した場合、仮想マシンは、NUMA で管理されなくなります。これらの仮想マシンの NUMA 管理は、CPU およびメモリアフィニティの制約を削除した場合は有効です。

手動での NUMA 配置は、プロセッサリソースをシステム全体で均等に配布する ESXi のリソース管理アルゴリズムに干渉する可能性があります。たとえば、プロセッサ集約型のワークロードを持つ 10 台の仮想マシンを 1 つのノードに手動で配置し、2 台の仮想マシンだけを手動で別のノードに配置した場合、システムは、12 台の仮想マシンすべてに均等にシステムリソースのシェアを与えることはできません。

仮想マシンと特定のプロセッサとの関連付け

仮想マシンの仮想 CPU を固定のプロセッサに関連付けることによって、仮想マシン上のアプリケーションのパフォーマンスを向上できることがあります。これにより、仮想 CPU が NUMA ノード間で移行されないようにできます。

手順

- 1 vSphere Client で、インベントリにある仮想マシンを右クリックして、[設定の編集] を選択します。
- 2 [リソース] タブを選択して、[CPU の詳細] を選択します。
- 3 スケジュール設定のアフィニティ パネルで、CPU アフィニティを優先するプロセッサに設定します。

注意 NUMA ノード内のすべてのプロセッサのボックスを手動で選択する必要があります。CPU アフィニティは、ノード単位ではなくプロセッサ単位で指定します。

メモリ アフィニティを使用した、メモリ割り当てと特定の NUMA ノードとの関連付け

仮想マシン上の将来のすべてのメモリ割り当てが、特定の NUMA ノードに関連付けられたページを使用することを指定できます（手動メモリ アフィニティとも呼ばれます）。

注意 将来のメモリ割り当てに使用するノードを指定するのは、CPU アフィニティも指定した場合だけにしてください。メモリ アフィニティの設定だけを手動で変更すると、自動 NUMA 再バランシングが適切に動作しなくなります。

手順

- 1 vSphere Client で、インベントリにある仮想マシンを右クリックして、[設定の編集] を選択します。
- 2 [リソース] タブを選択し、[メモリ] を選択します。
- 3 NUMA メモリのアフィニティ パネルでメモリ アフィニティを設定します。

例: 単一の NUMA ノードへの仮想マシンのバインド

次の例は、8 方向サーバ上の 2 方向の仮想マシン用に最後の 4 つの物理 CPU を単一の NUMA ノードに手動バインドする場合を示しています。

CPU（たとえば、4、5、6、7）は、物理 CPU 番号です。

- 1 vSphere Client のインベントリ パネルで仮想マシンを選択し、[設定の編集] をクリックします。
- 2 [オプション] を選択し、[詳細] をクリックします。
- 3 [構成パラメータ] ボタンをクリックします。
- 4 vSphere Client で、プロセッサ 4、5、6、7 の CPU アフィニティをオンにします。

次に、この仮想マシンをノード 1 上でのみ実行するとします。

- 1 vSphere Client のインベントリ パネルで仮想マシンを選択し、[設定の編集] をクリックします。
- 2 [オプション] を選択し、[詳細] をクリックします。
- 3 [構成パラメータ] ボタンをクリックします。
- 4 vSphere Client で、NUMA ノードのメモリ アフィニティを 1 に設定します。

これらの 2 つのタスクを完了すると、仮想マシンが NUMA ノード 1 でのみ実行されるようになり、可能な場合、同じノードからメモリを割り当てようになります。

仮想マシンと指定 NUMA ノードの関連づけ

NUMA ノード アフィニティを指定するために NUMA ノードを仮想マシンに関連付けた場合、NUMA が仮想マシンの仮想 CPU およびメモリのスケジュールを設定できる NUMA ノードを特定のノード セットに制限することができます。

注意 NUMA ノードのアフィニティを制限すると、NUMA ノード全体の仮想マシンのバランスが公平になるように再調整するための ESXi NUMA スケジューラの機能に影響を与えることがあります。NUMA ノードのアフィニティを指定する前には必ず、再バランスの問題を検討してください。

手順

- 1 vSphere Client で、インベントリにある仮想マシンを右クリックして、[設定の編集] を選択します。
- 2 [オプション] タブをクリックします。
- 3 [詳細] - [全般] を選択します。
- 4 [構成パラメータ] をクリックします。
- 5 [行の追加] をクリックし、新しいオプションを追加します。
- 6 名前の列に、「**numa.nodeAffinity**」 と入力します。
- 7 値の列に、仮想マシンをスケジュール設定できる NUMA ノードを入力します。
複数のノードを指定するには、コンマで区切ったリスト形式を使用します。たとえば、仮想マシンのリソース スケジュール設定を NUMA ノード 0 および 1 に制限するには、「**0,1**」 と入力します。
- 8 [OK] をクリックします。
- 9 [OK] をクリックして、仮想マシン プロパティ ダイアログ ボックスを閉じます。

詳細属性

ホストまたは個々の仮想マシンの詳細属性を設定すると、リソース管理をカスタマイズできます。

ほとんどの場合、基本的なリソース割り当て設定（[予約]、[制限]、[シェア]）を変更するか、デフォルト設定をそのまま使用することで、適切なリソース割り当てが行われます。ただし、詳細属性を使用すると、ホストまたは特定の仮想マシンのリソース管理をカスタマイズできます。

この章では次のトピックについて説明します。

- [ホストの詳細属性の設定 \(P. 99\)](#)
- [仮想マシンの詳細属性の設定 \(P. 102\)](#)

ホストの詳細属性の設定

ホストの詳細属性を設定できます。



注意 VMware テクニカル サポートまたはナレッジ ベースの記事で特に指示がない限り、詳細オプションの変更はサポートされていないと見なされます。その他の場合はすべて、これらのオプションの変更はサポートされていないと見なされます。ほとんどの場合、デフォルトの設定で最適な結果が得られます。

手順

- 1 vSphere Client で、インベントリからホストを選択します。
- 2 [構成] タブをクリックします。
- 3 [ソフトウェア] メニューで、[詳細設定] をクリックします。
- 4 詳細設定ダイアログ ボックスで、適切な項目（[CPU] または [メモリ] など）を選択します。
- 5 右パネルで属性を探し、値を編集します。
- 6 [OK] をクリックします。

メモリの詳細属性

メモリの詳細属性を使用して、メモリのリソース使用率をカスタマイズできます。

表 14-1. メモリの詳細属性

属性	説明	デフォルト
Mem.SamplePeriod	作業セット サイズを見積もるためにメモリ動作が監視される、定期的な時間間隔を指定します。これは仮想マシン実行時間の秒数で測定されます。	60
Mem.BalancePeriod	メモリの自動再割り当ての定期的な時間間隔を秒単位で指定します。空きメモリ量の大幅な変更によっても、再割り当てが開始されます。	15

表 14-1. メモリの詳細属性 (続き)

属性	説明	デフォルト
Mem.IdleTax	アイドルメモリ税率をパーセンテージで指定します。この税は、事実上、仮想マシンが有効に使用しているメモリよりも多くのアイドルメモリを仮想マシンに請求します。税率が0パーセントの場合は、作業セットを無視する割り当てポリシーを定義し、シェアに基づいて厳密にメモリを割り当てます。税率を高くすると、非生産的にアイドルメモリを蓄えている仮想マシンからアイドルメモリを再割り当てできる割り当てポリシーになります。	75
Mem.ShareScanGHZ	使用可能なホストのCPUリソース1GHzあたりについて、ページを共有するタイミングを調べるためにスキャンするメモリページの最大量(1秒あたり)を指定します。たとえば、デフォルトを1GHzあたり4MB/秒にします。	4
Mem.ShareScanTime	ページを共有するタイミングを調べるために、仮想マシン全体をスキャンする時間を分単位で指定します。デフォルトは60分です。	60
Mem.CtlMaxPercent	メモリバルーンドライバ(vmmemctl)を使用して、任意の仮想マシンから解放されるメモリの最大量を、構成されたメモリサイズのパーセンテージに基づいて制限します。0を指定すると、すべての仮想マシンに対して解放を無効にします。	65
Mem.AllocGuestLargePage	ホストの大きなページでのゲストの大きなページのバックアップを可能にします。TLBミスを減らし、ゲストの大きなページを使用するサーバワークロードのパフォーマンスを向上させます。0=無効。	1
Mem.AllocUsePSharePool および Mem.AllocUseGuestPool	ゲストの大きなページをホストの大きなページでバックアップできる可能性を高めることで、メモリの断片化を低減します。ホストメモリが断片化されていると、ホストの大きなページの利用可能率が低下します。0=無効。	15
Mem.MemZipEnable	ホストに対するメモリ圧縮を有効にします。0=無効。	1
Mem.MemZipMaxPct	圧縮メモリとして格納できる各仮想マシンのメモリの最大のパーセンテージとして、圧縮キャッシュの最大サイズを指定します。	10
LPage.LPageDefragEnable	大容量ページのデフラグを有効にします。0=無効。	1
LPage.LPageDefragRateVM	1台の仮想マシンが1秒間に試みる大容量ページのデフラグの最大回数。可能な値の範囲は1~1024です。	32
LPage.LPageDefragRateTotal	1秒間に試みる大容量ページのデフラグの最大回数。可能な値の範囲は1~10240です。	256
LPage.LPageAlwaysTryForNPT	ネストしたページテーブル(AMD社では「RVI」、Intel社では「EPT」という名称)に大きなページの割り当てを試みます。このオプションを有効にすると、すべてのゲストメモリは、ネストしたページテーブルを使用するマシン(たとえば、AMD Barcelonaなど)で大きなページによって補助されます。NPTが使用可能でない場合は、ゲストメモリの一部のみが大きなページで補助されます。0=無効。	1

NUMAの詳細属性

NUMAの詳細属性を使用して、NUMAの使用率をカスタマイズできます。

表 14-2. NUMAの詳細属性

属性	説明	デフォルト
Numa.RebalancePeriod	ミリ秒単位で指定される、再バランス期間の頻度を制御します。再バランスの頻度を上げると、実行中の仮想マシンを多数持つマシンにおいて特に、CPUオーバーヘッドが増大することがあります。再バランスの頻度を上げると、より適正になる場合もあります。	2000
Numa.MigImbalanceThreshold	NUMAリバランスは、各仮想マシンのCPU時間資格と実際の消費との間の差を考慮して、ノード間のCPU不均衡を計算します。このオプションは、仮想マシンの移行を起動するのに必要なノード間の最小負荷不均衡をパーセントで制御します。	10

表 14-2. NUMA の詳細属性 (続き)

属性	説明	デフォルト
Numa.RebalanceEnable	NUMA 再バランシングおよびスケジュール設定を有効にします。すべての NUMA 再バランシングと仮想マシンの初期配置を無効にして、NUMA スケジュール設定システムを実質的に無効にするには、このオプションを 0 に設定します。	1
Numa.RebalanceCoresTotal	NUMA リバランサを有効にするために必要な、ホスト上の合計プロセッサ コアの最小数を指定します。	4
Numa.RebalanceCoresNode	NUMA リバランサを有効にするために必要な、ノードあたりのプロセッサ コアの最小数を指定します。 このオプションと Numa.RebalanceCoresTotal は、NUMA 再バランシングが有効な場合に合計のまたはノードあたりの少数のプロセッサがスケジュール設定の公平性を解決できる、小型の NUMA 構成 (たとえば、2 方向 Opteron ホスト) 上で NUMA 再バランシングを無効にする場合に役に立ちます。	2
Numa.AutoMemAffinity	CPU アフィニティ セットを持つ仮想マシンのメモリ アフィニティを自動的に設定します。	1
Numa.PageMigEnable	メモリの局所性を改善するための NUMA ノード間での自動的なページ移行を行います。その場合も、手動で設定されたページ移行速度は有効です。	1

仮想 NUMA の詳細属性

仮想 NUMA の詳細属性を使用して、仮想 NUMA の使用率をカスタマイズできます。

表 14-3. NUMA の詳細属性

属性	説明	デフォルト
cpuid.coresPerSocket	仮想 CPU ソケットあたりの仮想コア数の決定。この値が 2 以上の場合、仮想マシンに仮想 NUMA トポロジがあれば、仮想 NUMA ノードのサイズも決定します。各物理ホストの仮想 NUMA トポロジを正確に知っている場合、このオプションを設定できます。	1
numa.autosize	このオプションを設定するとき、仮想 NUMA トポロジには、仮想ノードあたり、各物理ノードのコア数と同数の仮想 CPU があります。	FALSE
numa.autosize.once	これらの設定で仮想マシンのテンプレートを作成した場合、その後仮想マシンをパワーオンするたびに、設定は同じであることが保証されます。仮想 NUMA トポロジは、仮想マシンの仮想 CPU の構成数が変更されると再評価されます。	TRUE
numa.vcpu.maxPerVirtualNode	cpuid.coresPerSocket は 2 のべき乗で制約が強すぎる場合、直接、 numa.vcpu.maxPerVirtualNode を設定できます。このケースでは、 cpuid.coresPerSocket は設定しません。	8
numa.vcpu.min	仮想 NUMA トポロジを生成するために必要な最小限の、仮想マシン内の仮想 CPU 数。	9
numa.vcpu.maxPerMachineNode	NUMA ノードで同時にスケジュール可能な、同一の仮想マシンに属する最大仮想 CPU 数。この属性を使用することにより、別の NUMA ノードの別の NUMA クライアントに強制して、確実に最大の帯域幅を得ます。	仮想マシンが実行中の物理ホストの、ノードあたりのコア数。
numa.vcpu.maxPerClient	NUMA クライアント中の最大仮想 CPU 数。1 つのクライアントは、単一エンティティとして NUMA に管理された仮想 CPU のグループです。デフォルトでは、各仮想 NUMA ノードは NUMA クライアントです。ただし、ある仮想 NUMA ノードが物理的な NUMA ノードより大きい場合、単一の仮想 NUMA ノードを複数の NUMA クライアントでバックアップできます。	numa.vcpu.maxPerMachineNode に等しい

表 14-3. NUMA の詳細属性 (続き)

属性	説明	デフォルト
numa.nodeAffinity	仮想マシンの仮想 CPU とメモリがスケジュールできる NUMA ノードのセットに制約を加えます。 注意 NUMA ノードのアフィニティを制限すると、NUMA ノード全体の仮想マシンを再バランスする NUMA スケジューラの機能を妨害している場合があります。NUMA ノードのアフィニティを指定する前には必ず、再バランスの問題を検討してください。	
numa.mem.interleave	1 台の仮想マシンに割り当てられたメモリが、それが構成する NUMA クライアントが実行されているすべての NUMA ノードで静的にインタリーブされるかどうかと、公開されている仮想 NUMA トポロジがないかどうかを指定します。	True

仮想マシンの詳細属性の設定

仮想マシンの詳細属性を設定できます。

手順

- 1 vSphere Client で、インベントリにある仮想マシンを右クリックして、[設定の編集] を選択します。
- 2 [オプション] をクリックして、[詳細] - [全般] をクリックします。
- 3 [構成パラメータ] をクリックします。
- 4 表示されるダイアログボックスで [行の追加] をクリックして、新しいパラメータとその値を入力します。
- 5 [OK] をクリックします。

仮想マシンの詳細属性

仮想マシンの詳細属性を使用して、仮想マシンの構成をカスタマイズできます。

表 14-4. 仮想マシンの詳細属性

属性	説明	デフォルト
sched.mem.maxmemctl	選択した仮想マシンからバレーニングによって解放される、メガバイト (MB) 単位での最大メモリ量です。ESXi ホストが追加メモリを解放しなければならない場合、ESX ホストはスワップを強制されます。スワップはバレーニングよりも好ましくありません。	-1 (制限なし)
sched.mem.pshare.enabled	選択した仮想マシンのメモリ共有を有効にします。 このブール値のデフォルトは [True] です。仮想マシンに対して [False] に設定すると、メモリ共有がオフになります。	True
sched.swap.persist	仮想マシンがパワーオフになった場合に仮想マシンのスワップファイルを残すか、削除するかを指定します。デフォルトでは、システムは、仮想マシンがパワーオンになった時点でその仮想マシンのスワップファイルを作成し、仮想マシンがパワーオフになった時点でスワップファイルを削除します。	False
sched.swap.dir	仮想マシンのスワップファイルの VMFS ディレクトリの場所です。デフォルトでは、仮想マシンの作業ディレクトリになります。つまり、仮想マシンの構成ファイルを格納する VMFS ディレクトリです。このディレクトリは、仮想マシンにアクセス可能なホスト上にある必要があります。仮想マシン (または仮想マシンから作成されたクローン) を移動する場合は、この属性をリセットする必要があることがあります。	workingDir に等しい

インデックス

A

AMD Opteron ベースのシステム 91, 100

B

Baseboard Management Controller (BMC) 67

C

CPU

アドミッションコントロール 22

オーバーコミット 15

仮想 95

割り当ての管理 15, 17

CPU アフィニティ

NUMA ノード 97

潜在的な問題 22

ハイパースレッド 19

CPU 仮想化

ソフトウェア ベース 15

ハードウェア アシスト 16

CPU 電力効率 22

CPU の構成 17

CPU バインド アプリケーション 16

D

Distributed Power Management、[DPM] を参照

DPM

オーバーライド 70

アドミッションコントロール 14

監視 70

しきい値 69

自動化レベル 69

前回のスタンバイ終了 70

有効化 69

DRS

vMotion ネットワーク 55

移行 51

移行のしきい値 54

移行の推奨 54

一部自動化 56

仮想マシンの移行 53

完全自動化 56

グループ パワーオン 52

手動 56

初期配置 51, 52

単一仮想マシンのパワーオン 52

無効化 58

ルールの作成 72

ロード バランシング 51

DRS クラスタ

管理されていないホストの追加 60

管理対象ホストの追加 59

共有 VMFS ボリューム 55

共有ストレージ 55

作成 56

前提条件 55

妥当性 62

プロセッサの互換性 55

リソースの管理 59

リソースの提供者として 9

DRS グループ

仮想マシン 72

ホスト 71

Dynamic Voltage and Frequency Scaling (DVFS) 22

E

EVC (Enhanced vMotion Compatibility) 16, 55, 56

I

iLO、構成 67

IPMI (Intelligent Platform Management Interface)、構成 67

L

LPage.LPageAlwaysTryForNPT 99

LPage.LPageDefragEnable 99

LPage.LPageDefragRateTotal 99

LPage.LPageDefragRateVM 99

M

Mem.AllocGuestLargePage 99

Mem.AllocUseGuestPool 99

Mem.AllocUsePSharePool 99

Mem.BalancePeriod 99

Mem.CtlMaxPercent 99

Mem.IdleTax 31, 99

Mem.MemZipEnable 99

Mem.MemZipMaxPct 99

Mem.SamplePeriod 30, 99

Mem.ShareScanGHz 35, 99

Mem.ShareScanTime 35, 99

N

NUMA

- AMD Opteron ベースのシステム 94
- CPU アフィニティ 97, 98
- IBM Enterprise X-Architecture 94
- オペレーティング システムについての課題 91
- 仮想 94, 95
- 最適化アルゴリズム 93
- サポートされるアーキテクチャ 94
- 手動制御 96
- 詳細属性 100
- スケジュール設定 92
- 説明 91
- 透過的なページ共有 94
- 動的ロード バランシング 93
- ページの移行 93
- ホーム ノード 93
- ホーム ノードと初期配置 93
- メモリのアフィニティ 98
- メモリ ページの共有 94
- Numa.AutoMemAffinity 100
- Numa.MigImbalanceThreshold 100
- Numa.PageMigEnable 100
- Numa.RebalanceCoresNode 100
- Numa.RebalanceCoresTotal 100
- Numa.RebalanceEnable 100
- Numa.RebalancePeriod 100

S

- sched.mem.maxmemctl 32, 102
- sched.mem.pshare.enable 102
- sched.swap.dir 102
- sched.swap.persist 102
- SMP 仮想マシン 16
- SSD へのスワップ 35
- Storage vMotion
 - 推奨 78
 - データストア クラスタの互換性 90

V

- vCenter Server イベント 70
- VMFS (仮想マシン ファイル システム) 55, 102
- VMFS ポリュームの要件 55
- VMM 25, 26
- vmmemctl, Mem.CtlMaxPercent 99
- vMotion 要件 56
- VMX スワップ ファイル 31
- VMX へのスワップ 31
- vSphere DRS、ルール の作成 73
- vSphere HA 56, 59, 61

W

- Wake プロトコル 67
- WOL (Wake-On-LAN)、テスト 68

あ

- アイドル メモリ税 31
- 赤の DRS クラスタ 66
- アドミッション コントロール
 - CPU 22
 - 拡張可能なリソース プール 48
 - リソース プール 47
- アフィニティ ルール
 - 仮想マシン内 88
 - 作成 72, 73
 - ストレージ DRS 87
- アプリケーション
 - CPU バインド 16
 - 単スレッド 16
- アラーム 70

い

- 移行のしきい値、DRS 54
- 移行の推奨 54
- 一部自動化された DRS 56
- 移動、リソース プール 59, 60

え

- エラー隔離の修正 38

お

- オーバーコミットされた DRS クラスタ 65
- オーバーヘッド メモリ 25
- 親リソース プール 43

か

- 拡張可能な予約、例 48
- 隔離、ハイパースレッド 21
- カスタム自動化モード、DRS 57
- 仮想 CPU、数の変更 95
- 仮想 NUMA
 - 詳細オプション 95
 - 詳細属性 101
 - 制御 95
- 仮想マシン
 - DRS クラスタからの削除 61
 - DRS クラスタへの追加 60
 - DRS 自動化レベル 57
 - 移行 53
 - オーバーヘッド メモリ 30
 - 仮想プロセッサ数 16
 - 監視 26
 - 構成ファイル 55
 - 詳細属性 102
 - 特定のプロセッサへの割り当て 21

- メモリ 25, 31
- メモリの共有 35
- リソースの消費者として 10
- リソース プールからの削除 47
- リソース プールへの追加 46
- 仮想マシン間のアフィニティ ルール、競合 73
- 仮想マシン間非アフィニティ ルール、作成 88
- 仮想マシンとホスト間のアフィニティ ルール
 - 競合 74
 - 使用 74
- 仮想マシン内非アフィニティ ルール 88
- 仮想マシンの DRS グループ 72
- 仮想マシンのアフィニティ 71-74
- 仮想マシンの非アフィニティ 71-74
- 仮想マシン ファイル システム (VMFS) 55, 102
- 仮想マシン モニタ (VMM) 25
- 監視、ストレージ I/O コントロール 40
- 監視ソフトウェア 70
- 完全に自動化された DRS 56

き

- 黄色の DRS クラスタ 65
- 兄弟 43

く

- クラスタ設定、アフィニティ ルール 72, 73
- クラスタリング、データストア 77, 82
- グループ
 - DRS の仮想マシン 72
 - DRS ホスト 71
- グループ パワーオン 52

さ

- 作業セット サイズ 30

し

- シェア、ストレージ I/O コントロール 39
- シェアおよび制限、ストレージ I/O コントロール 40
- しきい値、ストレージ I/O コントロール 42
- システム リソース割り当てテーブル (SRAT) 92
- 自動化レベル
 - 仮想マシン 57
 - ストレージ DRS 85
 - データストア クラスタ 79
- 手動の DRS 56
- 詳細属性
 - NUMA 100
 - 仮想 NUMA 101
 - 仮想マシン 102
 - ストレージ I/O コントロール 42
 - ホスト 99
 - メモリ 99

- 初期配置
 - NUMA 93
 - ストレージ DRS 78
- 序文 7

す

- 推奨、ストレージ DRS 84
- スタンバイの終了エラー 70
- スタンバイ モード、前回のスタンバイ終了 70
- ストレージ DRS
 - I/O ロード バランシング 80
 - アフィニティ ルール 87
 - 概要 79
 - しきい値 80
 - 自動化レベル 85
 - 初期配置 78
 - 推奨 78, 84, 85
 - スケジュール設定タスク 86
 - 積極性 80
 - 統計情報 89
 - 非アフィニティ ルール 88
 - 無効化 79
 - メンテナンス モード 83
 - 有効化 79
 - 領域ロード バランシング 80
- ストレージ I/O コントロール
 - 監視 40
 - シェアおよび制限 39-41
 - しきい値 42
 - 制限事項 39
 - 有効化 41
 - 要件 39
- ストレージの移行の推奨 78
- ストレージ要件 55
- スワップ スペース 33
- スワップ ファイル
 - VMX 31
 - 削除 34
 - 使用 32
 - 場所 33

せ

- 制限 12
- 前回のスタンバイ終了 70

た

- 単一仮想マシンのパワーオン 52
- 単一スレッド アプリケーション 16
- 単一プロセッサ仮想マシン 15

て

- データストア
 - 互換性 81
 - メンテナンス モード 83
- データストア クラスタ
 - 概要 77
 - 作成 78–80
 - 自動化レベル 79
 - データストアの削除 82
 - データストアの追加 82
 - メンテナンス モード 83
 - リソースの提供者として 9
- デュアル プロセッサ仮想マシン 15
- 電力管理ポリシー、CPU 22

と

- 動的ロード バランシング、NUMA 93

は

- ハイパースレッド
 - CPU アフィニティ 19
 - 隔離 21
 - コア共有モード 20
 - サーバ構成 20
 - とホスト 19
 - パフォーマンスの影響 18
 - 無効化 17
 - 有効化 19
- ハイパースレッド モード 20
- ハイパースレッド用のサーバ構成 20
- パフォーマンス、CPU バインド アプリケーション 16
- バルーニング、メモリ 32
- パワーオン、単一仮想マシン 52

ふ

- 物理プロセッサ 17
- プロセッサ、互換性要件 55
- プロセッサ固有の動作 16

へ

- ページの移行、NUMA 93

ほ

- ホーム ノード、NUMA 93
- ホスト
 - DRS クラスタからの削除 62
 - DRS クラスタへの追加 59, 60
 - 詳細属性 99
 - メモリ割り当て 30
 - メンテナンス モードへの切り替え 61
 - リソースの提供者として 9
- ホスト、DRS クラスタからの削除 61
- ホスト DRS グループ 71
- ホスト キャッシュ、スワップ 34

- ホスト キャッシュ構成 35
- ホスト キャッシュへのスワップ 34
- ホストの電力管理、カスタム ポリシー 23
- ホストの電力管理ポリシー、設定 23
- ホストのローカルスワップ
 - DRS クラスタ 33
 - スタンドアロン ホスト 33
- ポリシー
 - CPU 電力管理 22
 - ホストの電力管理 23

ま

- マルチコア プロセッサ 18

む

- 無効な DRS クラスタ 66

め

- メモリ
 - オーバーコミット 26, 33
 - オーバーヘッド 25
 - オーバーヘッド、理解 29
 - 仮想マシン 31
 - 仮想マシン間での共有 35
 - 共有 26
 - 詳細属性 99
 - バルーン ドライバ 32
 - 未使用の解放 31
 - 割り当ての管理 29
- メモリ圧縮 36
- メモリ圧縮キャッシュ
 - サイズの設定 36
 - 無効化 36
 - 有効化 36
- メモリ仮想化
 - ソフトウェア ベース 26
 - ハードウェア アシスト 27
- メモリ使用率 37
- メモリのアイドル税 31
- メモリのアフィニティ、NUMA ノード 97
- メモリの解放 31
- メモリの共有 26
- メモリの信頼性 38
- メンテナンス モード
 - アフィニティ ルールの無視 84
 - データストア 83, 84
 - ホスト 61
 - ホストを切り替える 61
- メンテナンス モードへの切り替え、ホスト 61

ゆ

- 有効な DRS クラスタ 63

よ

- 要件、データストア クラスタ 81
- 予約 12

り

- リソース管理
 - カスタマイズ 99
 - 目的 10
- リソースの消費者 10
- リソースのタイプ 9
- リソースの提供者 9
- リソース プール
 - アドミッション コントロール 47
 - 移動 59, 60
 - 親 43
 - 仮想マシンの削除 47
 - 仮想マシンの追加 46
 - 兄弟 43
 - 削除 47
 - 作成 45
 - 属性の編集 46
 - メリット 44
 - ルート リソース プール 43
- リソース割り当て設定
 - シェア 11
 - 推奨事項 13
 - 制限 12
 - 変更 13
 - 予約 12

る

- ルート リソース プール 43

ろ

- ロード バランシング
 - 仮想マシン 53
 - ストレージ DRS 80
 - データストア 77
- 論理プロセッサ 17, 18

