サーバ仮想化技術導入に当たっての評価ポイント

野村総合研究所
システム技術一部　上級テクニカルエンジニア

松本　健（まつもとけん）

情報技術本部にてシステム基盤を中心とした新技術の調査、評価を行うITスペシャリスト。
最近はESB/BPM/ユーティリティコンピューティングの調査、評価を行っている。

1．サーバ仮想化が求められる背景 36
2．サーバ仮想化とその関連技術 37
3．活用例 40
4．サーバ仮想化技術の評価ポイント 42
5．動的ワークロード管理技術の評価ポイント 46
6．今後の動向 48

要旨

ITシステムの高度化、複雑化に伴い、システムで使用されるサーバ数が増加し、維持管理コストの負荷が増大している。また、サーバにはピーク時の負荷に対応するために余剰リソースが搭載されており、リソースが有効活用されているとはいえない。そのため、システムリソースを有効活用することによるコスト削減や、運用負荷の低減へのニーズが高まっている。このニーズに対応する技術として、ユーティリティコンピューティングを実現する主要技術であるサーバ仮想化技術が実用レベルに達しつつあり、すでに一部では導入が始まっている。しかし、必ずしも目的を満たしているとは言い切れず、最適な製品を評価することが重要であるが、評価手法はこれまで確立されていない。NRIでは、サーバ仮想化とその関連技術にフォーカスを当てて実機評価を行い、今後導入に当たって活用し得る技術面、実用面を考慮した評価手法を確立した。

キーワード：　ユーティリティコンピューティング、仮想マシン、パーティショニング、動的ワークロード管理、ポリシーベース運用、ブレードサーバ、サーバコンソリデーション

Advanced and complicated IT system infrastructures force to increase the number of servers and their maintenance cost. While to cope with the high needs at peak time excessive resources are implemented, resources are not utilized effectively. In this regard, the measurements to reduce the cost and extra workload for effectively utilizing those resources are increasingly needed. To meet these needs, server virtualization technology has almost reached a practical level to enable the implementation of utility computing in actual environment. This article is focused on the server virtualization and related technologies. Furthermore, describing tips for implementation in actual environment and technology perspective in future.

Keywords : Utility computing, Virtual machine, Partitioning, Dynamic work load management, Policy based management, Blade server, Server consolidation

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

1．サーバ仮想化が求められる背景

(1) リソース利用率の低下

これまでのシステムは、定常的に性能を確保するために、ピーク時のリクエストをさばけるように設計を行ってきた。このような設計をすると、ピーク時以外（つまり平常時）のリクエストは非常に少なく、リソースがほとんど使われていないことが多い、つまり余剰リソースが生まれる原因となっている。システムの処理のピークは、業務、業態によって異なるが、月に1回締め時のみ、年に2～3回棚卸時のみなどさまざまであり、いずれもピーク時以外のリソース利用率は低い（図1参照）。

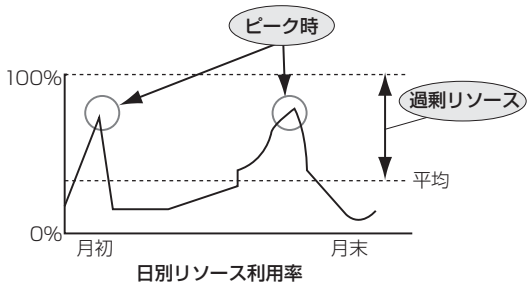


図1　リソース利用率の低下

(2) 維持管理コストの負荷増大

IT化が進む中、さまざまな業務がシステム化され、維持管理対象のサーバが増えてきている。既存のシステムの処理量の増加に対応するためにサーバ台数を増やしたり、レスポンスタイムを迅速にするためにサーバをスケールアウトし、台数の増加も行われる。そのため管理対象となるサーバが数十台、数百台となり、管理負荷が増大する。

(3) ユーティリティコンピューティング

上記2つの理由から、余剰リソースを減らし、サーバ台数を減らすことが課題となっている。この課題を解決するための技術としてユーティリティコンピューティングがある（図2参照）。ユーティリティコンピューティングとは、ITリソースを電気や水道のように、使いたいときに使いたいだけ使えるサービスとして提供することである。

ニーズに応じて必要なだけのリソースをサービスに割り当てたり、突発的な負荷に備えて保持する余剰リソースをサービス間でシェアしたり、異なるピーク時間を持つサービス間でリソースを融通することが可能になる。ユーティリティコンピューティングを実現するための主要な技術としては、サーバなどのシステムリソースの仮想化と、動的なワークロードの管理、ポリシーベースの運用がある。現在は特にサーバ仮想化と動的ワークロード管理の技術が熟してきており、本稿ではこれらに焦点をあてて述べる。

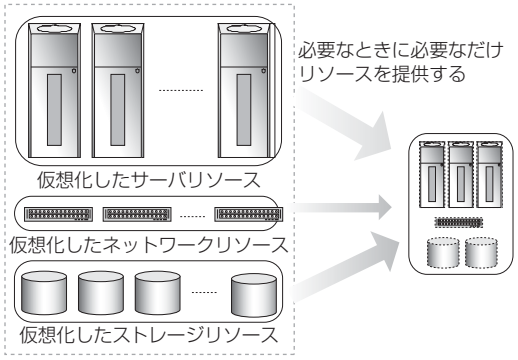


図2　ユーティリティコンピューティング

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

2．サーバ仮想化とその関連技術

(1) サーバ仮想化の種類

サーバ仮想化は、1台の物理的なサーバリソースを分割、統合し、仮想的に何台ものサーバが動作しているかのように見せたり、複数の物理サーバを1台のサーバに見せる技術である。この仮想化されたサーバごとにOSやアプリケーションを割り当てることができる。このようなサーバ仮想化を実現する方法は、大きく分けて2種類ある。

① ブレードサーバ

複数の物理サーバを束ね、1つまたは複数の仮想サーバを構成するタイプ（図3参照）。

ブレードサーバとは、プロセッサ、メモリ、ネットワークインターフェイスを備えたマザーボード部分である「ブレード」と、スイッチや電源を備え、ブレードを複数収容する「エンクロージャ」で構成される。

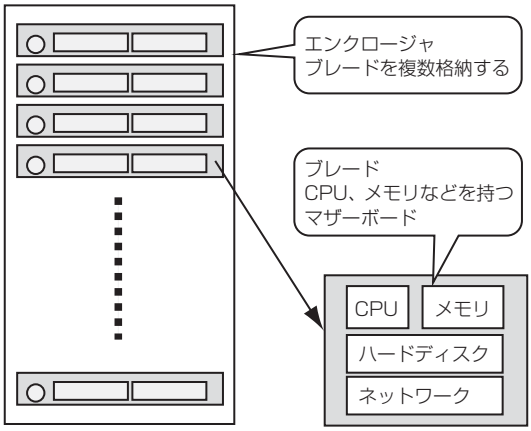


図3　ブレードサーバ

② パーティショニング・仮想マシン

1つの物理サーバを、複数の仮想サーバに分割するタイプ（図4参照）。

パーティショニングは、単一筐体内に複数のOSイメージを起動させることで、仮想的に複数のマシンがあるかのように利用できるようにする技術である。仮想マシンはOS上で主にPC/AT互換ハードウェアをエミュレートする技術である。

本稿では現在実用レベルに達しつつあるパーティショニング・仮想マシンについて詳しく説明していく。

a. パーティショニング

パーティショニングには大きく分けて物理パーティショニングと論理パーティショニングの2種類がある。1台の物理サーバを物理的なブロック単位（セル単位）に分割するものを物理パーティショニングという。セルとは、1つ以上のプロセッサ、メモリ、I/Oカードを1つの塊にしたものである。物理パーティショニングでは、ハードウェア障害や負荷の影響がパーティション同士で影響を与えないのが特徴である。またCPU単位、メモリブロック単位で分割し、どのパーティションにも自由にリソースを配分できるようにしたものを論理パーティショニングという。

パーティショニングは主にUNIX系ベンダが採用しており、ヒューレットパッカード社では物理パーティショニングをnPartitions（nPars）、論理パーティショニングをVirtual

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

Partitions（vPars）、IBM社ではPhysical Partitions（PPAR）、Logical Partitions（LPAR）と命名している。

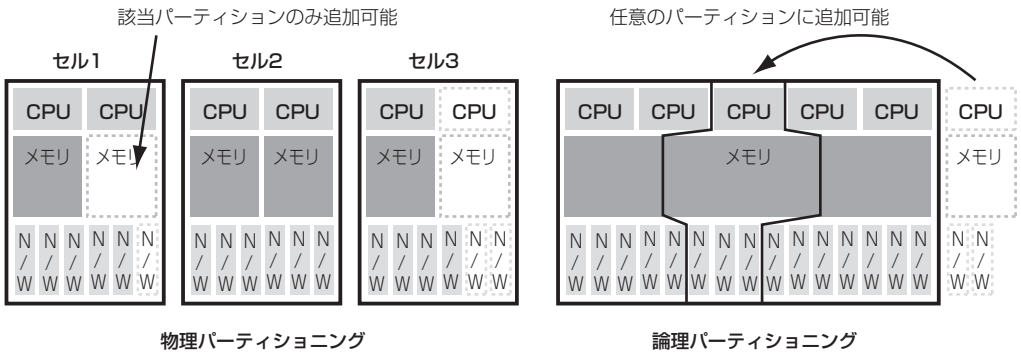


図4　パーティショニング

b. 仮想マシン

仮想マシンでは、WindowsなどのOS上に（ホストOS）PC/ATのハードウェアを仮想的に構築する（仮想マシン）。この仮想マシンにPC/AT上で動作するOSをインストールすることができる（ゲストOS）（図5参照）。

仮想マシンを採用するベンダにはVMWare社、マイクロソフトなどがあり、VMWare GSX（VMWare社）、Virtual Server（マイクロソフト）などの製品が有名である。この方式は動作の安定性に加え、VMWare ESXのようにホストOSを必要とせず非常に高速なものも出始めている。

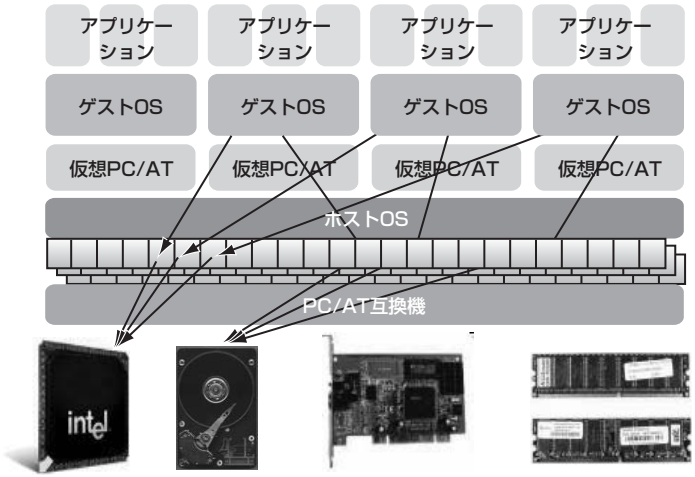


図5　仮想マシン

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

(2)動的ワークロード管理

動的ワークロード管理とは、仮想化されたサーバ(仮想サーバ)にかかるアクセス負荷(ワークロード)をリアルタイムに捉え、その状況に応じて仮想サーバに割り当てる物理サーバリソースを動的に変更し、サーバアクセス負荷に対応できるようにすることである。

例として、図6のような流通系システムを考えてみる。このシステムは発注、会計、商品マスタエントリなどのサブシステムで構成されており、各サーバは、各処理のピーク時に対応できるサイジングを行った結果としてのサーバリソースで構成されている。ピーク時間帯は、発注、商品マスタエントリ、会計、情報発信の場合はオンライン処理が中心であるため昼間であり、バッチサーバの場合は夜

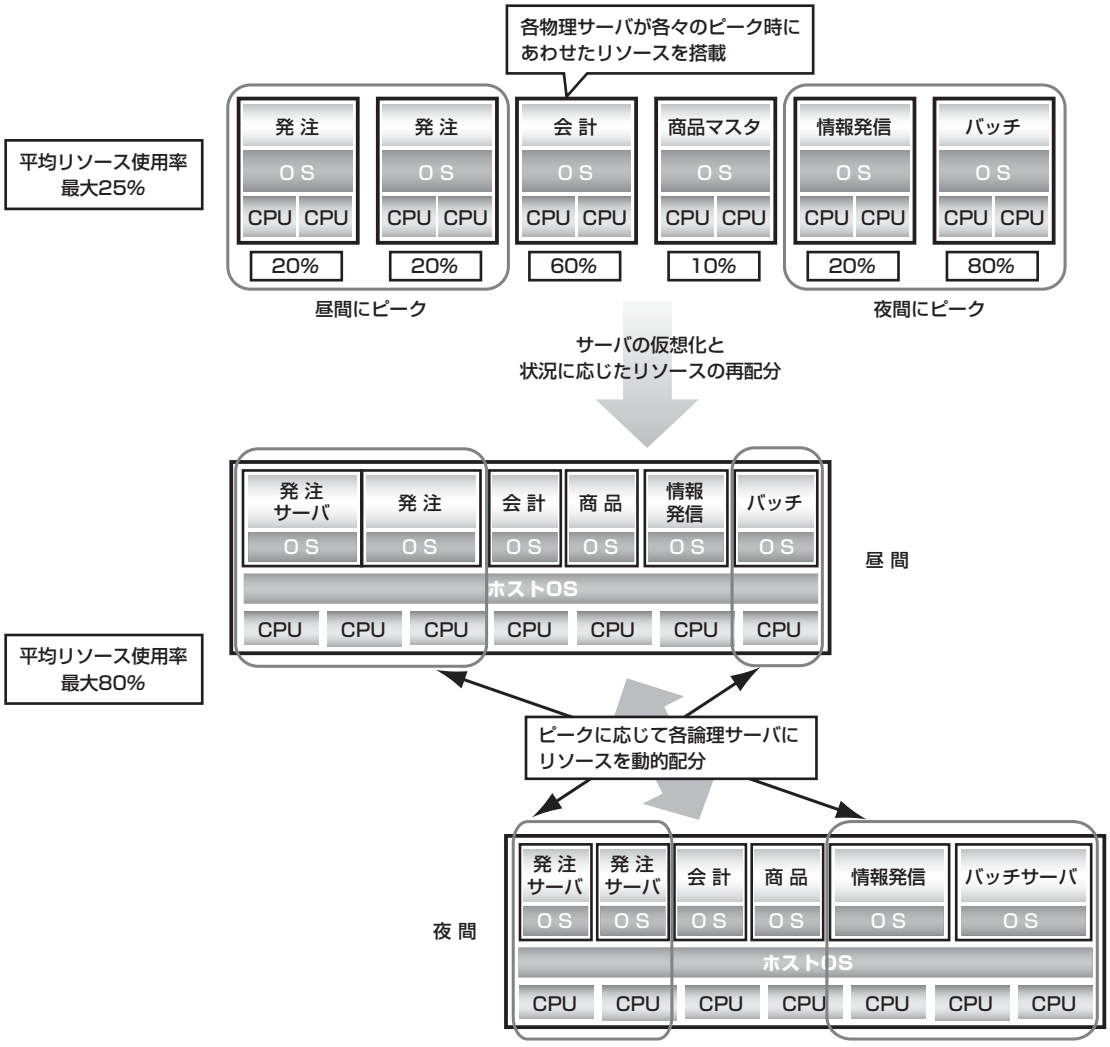


図6　動的ワークロード管理の例

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

間稼動するため夜間である。このように、一日の単位で考えると、ピークの時間帯は昼と夜の2回あることになる。さらにピーク時と平常時のアクセス量には差があり、各サーバの使用率は平均して25%程度とする。

このシステムを仮想サーバを用いて構築すると、各仮想サーバに割り当てることのできる物理リソースは動的に変更できるようになる。するとピーク時間帯がずれているため、昼間はオンライン系のシステムに物理サーバリソースを集中的に割り当て、夜間はバッチ系システムにリソースを集中的に割り当てるように、動的にリソース量を変更することにより、物理サーバリソースを有効に利用することができ、全体としてより少ない物理サーバでシステムを実現できる。場合にもよるが、最も効果的な場合、物理サーバリソースの使用率は80%程度まで上げることが可能といわれている。

3．活用例

もともとユーティリティコンピューティングを実現する技術として、サーバリソースの柔軟な利用を行うためにサーバ仮想化技術が発展してきた。ここではサーバ仮想化技術のさまざまな活用方法について考えてみる。

(1) リソースの最適配分

前述のサーバ仮想化、動的ワークロード管理の説明でも述べたが、負荷に応じた動的な

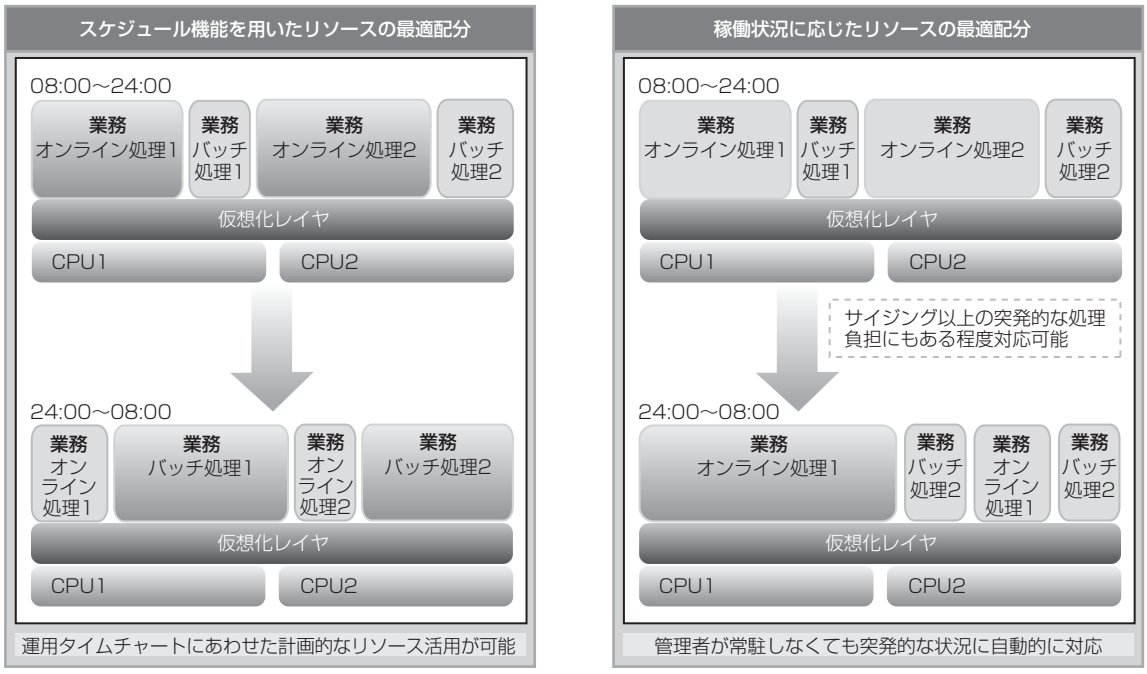


図7　リソースの最適配分

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

リソース配分がある。リソース配分方法としては、オンライン処理やバッチ処理などピーク時間帯が前もってわかっている場合には時間帯によるリソースの制御、突発的なWebサイトへのアクセスのような場合には、稼動状況に応じた制御が必要になってくる（図7参照）。

（2）開発機への応用

仮想化されたサーバは、すべて同一の仮想デバイスで構成されている。つまり、同一のデバイス、ハードウェア構成をもつサーバである（もちろん割り当てられている物理リソースの量によって、CPU処理能力、ネットワーク帯域、ディスク容量などは異なる）。OSから見れば、すべての仮想サーバは同一のデバイスドライバを用いて制御される。

現在、システム開発時に各サブシステムの構築を行うアプリケーションチームでは、それぞれに開発機を購入し、アプリケーション開発を行っている。開発機導入当初はすべてのアプリケーションチームで環境が同じかもしれないが、さまざまなプロジェクトの事情によりスペックが各チームで変更される場合がある。そのためチームごとに設定項目が異なるため、設定項目のミスが発生することがある。また、開発機の調達、廃棄にも、発注から納入、セッティング、OSインストールまで数日～数か月かかり、必要がなくなった場合に償却の問題から簡単に捨てることができないなどの手間がかかるという問題がある。

仮想サーバを利用することで、同一の仮想デバイスを利用することからまったく同じ構成のサーバをチームごとに作成することができるため、設定項目や値がすべて同じになり項目の設定ミスの発生件数を削減することができる。また、仮想サーバ自体の作成、削除については、必要なときにGUIなどを利用して即時に実行することができるため、手間がかかっていた調達、廃棄の問題も軽減することが可能である（図8参照）。

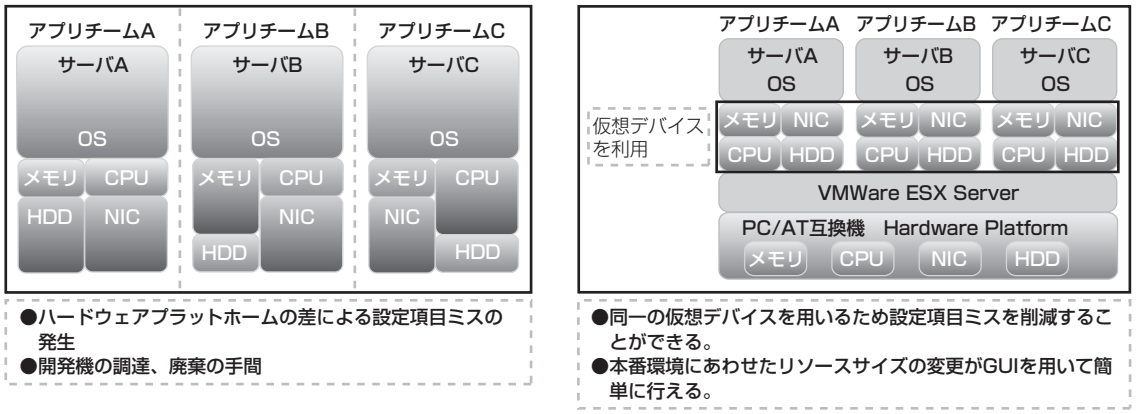


図8　開発機への応用

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

（3）待機系サーバへの応用

H/A（High Availability：高可用性）構成において、待機系サーバを集約することによって物理サーバ台数を減らすことも考えられる。

複数のH/A構成のシステムの待機系サーバを1台の物理サーバ上に仮想サーバとして構築する。待機系の仮想サーバは、初期構成時にはハートビート情報などを処理できる最小の物理リソースを与えておく。フェイルオーバ時には正常なサーバと同等のリソースを動的配分する。これにより、待機系サーバの統合を行い、休眠リソースの削減を図ることが可能である（図9参照）。

4．サーバ仮想化技術の評価ポイント

サーバ仮想化技術を利用するにあたり、仮想的なサーバを作ることに関しては、当然ではあるがどの製品も可能である。しかし、ワークロード管理を行う場合に、単に仮想的なサーバが構築できるだけではなく、いかに滞りなく物理サーバリソースを自由に割り当てたり、解放できるかにも留意する必要がある（表1参照）。ここではこれらの留意すべきポイントを説明する。

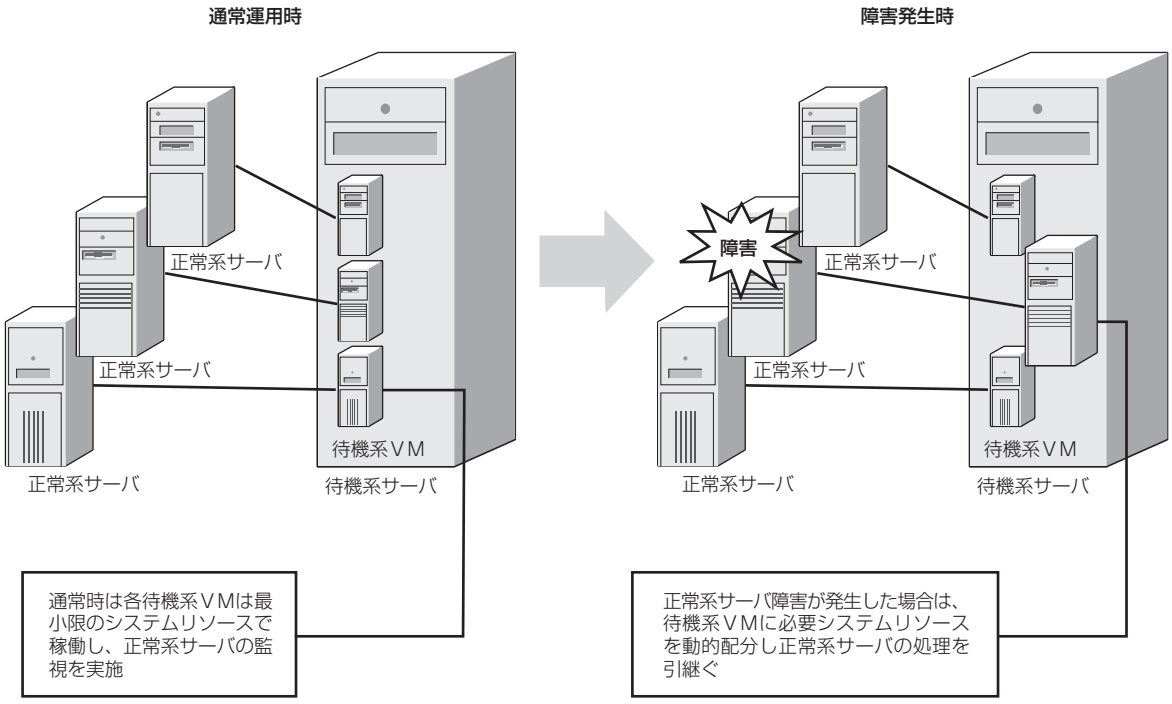


図9　待機系サーバへの応用

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

	製品A	製品B
動的リソース配分単位	1物理CPU単位	%単位
物理構成変更時の仮想化への影響	メモリ、N/Wなどはリブートが必要	リブート不要
ホストOSの有無	必要	不要
ゲストOSの種類	ホストOSと同種のUNIX	PC/AT上で動作するOS
仮想サーバ間の独立性	パーティション間はN/W経由のアクセスのみ可	パーティション間はN/W経由のアクセスのみ可
仮想サーバの制御方法	コマンドライン	C/C++のAPI
性能	ホストOSのオーバヘッドあり	ホストOSがないため高速
信頼性対策	別途クラスタソフトでの対応	別途クラスタソフトでの対応
課金単位	物理マシンCPU数	物理マシンCPU・作成可能なVM数

※ N/W：ネットワーク

表1　サーバ仮想化技術の評価ポイント

(1) 動的なリソースの配分単位

単一物理サーバ内に存在する仮想サーバは、それぞれの仮想サーバごとに必要とする物理サーバリソースを分割して利用する。リソースの配分も製品によって異なる。

物理サーバリソースの配分の単位が、物理デバイス単位になっているタイプのものがある。たとえば物理サーバがCPUを4つ持っている場合、最高でも4つの仮想サーバしか構成できず、かつそれぞれの仮想サーバが物理CPUを1つしか持つことができないという製品も存在する。この場合、CPUの割り当て単位が1物理CPUとなるため、4CPUの物理サーバ上に3つの仮想サーバを構築した場合、そのうちの1つの仮想サーバが2物理CPUを使えることになる。つまり、サーバ負荷に応じて割り当てと解放のできるCPUリソースが1物理CPU単位となるため、きめ細かなリソース配分は難しい。

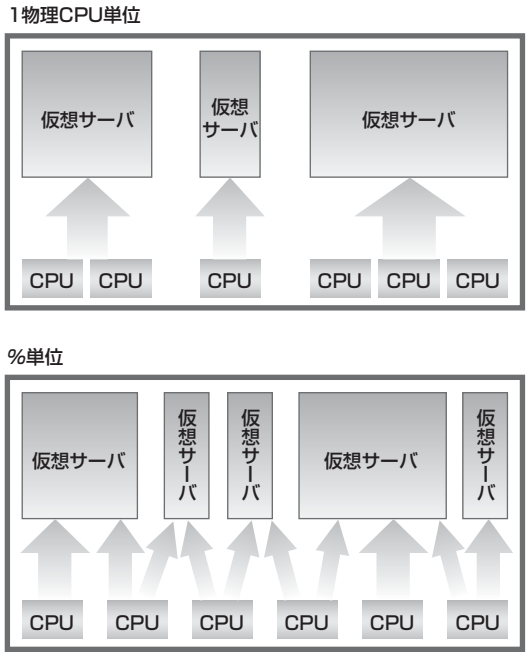


図10　リソース配分単位

もう1つの方法は、全物理リソースを全体に占める割合で各仮想サーバに割り当てることができるタイプがある。このタイプは物理サーバが持っているすべてのCPUの何%を仮想サーバに割り当てるかという設定ができるため、物理CPUの数以上に仮想サーバを構築できるというメリットがある。加えて割り当てや解放も%単位で行えるため、よりきめ細かなワークロードに対応するリソース配分が可能となる（図10参照）。

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

(2) 物理構成変更時の仮想化への影響

各仮想サーバが稼動している時に、ピーク時間帯にある仮想サーバへのアクセスが集中し、サーバの処理負荷が増大し、現在のリソースでは対処し切れない場合、動的に物理リソースの割り当てを増やして正常に処理を続行しようとする。仮想サーバに割り与えられている物理リソースの配分を仮想サーバが実行状態にある場合に動的に変更するときの留意点として、リソース配分変更の反映タイミングがある。

理想的には物理リソース配分の設定直後に変更されることが望ましい。CPUの配分方法がタイムシェアリング方式をとっているものが多いため、この反映は設定直後に実施されるものが多い。ただし、製品によっては仮想サーバに最低限のリソースとして半固定的にCPUリソースや、メモリ、ネットワークインターフェイスなどを割り当てるため、配分変更後はリブートが必要なものもあるため注意が必要である。

(3) ホストOSの有無

仮想サーバを構築する際に、仮想サーバそのものを制御するためのプログラムがOSを必要とするものとそうでないものがある。このときのOSをホストOSと呼び、物理サーバ上にインストールしておく必要がある。

ホストOSを利用するタイプは、通常WindowsやUNIXなどを使用する。したがって、物理サーバのハードウェア構成や種類が異なってもホストOSが稼動すれば仮想サーバ製品がその上で動作することが保証される。ただし留意すべき点は、実行時のホストOSのオーバヘッドを考慮する必要があるということと、ホストOSのライセンス料が発生するということである。

ホストOSが不要なタイプでは、仮想サーバ制御用にカスタマイズされた非常に軽量な独自のOSを利用することで、オーバヘッドを削減し高速に動作させることが可能である。ただし、留意すべき点は、ハードウェアの相性がある点である。前述のホストOSの場合は広く利用されている一般的なOSを使用するため対応するハードウェアは数多いが、専用のOSを利用するホストOS不要タイプではベンダ側で稼動確認をして動作保障ができるハードウェアの数は限られている場合が多い。

(4) 対応するゲストOSの種類

仮想サーバ上で稼動するOSをゲストOSと呼ぶ。対応するゲストOSの種類は、製品によって異なる。概してUNIX系とインテル系の製品で種類が分けられる。

UNIX系ベンダは、前述のとおりパーティション技術を採用しており、基本的に自社OSのみの対応が中心のものが多い（HP-UX、AIXなど）。これらベンダは自社で販売している物理サーバ上でパーティショニングを行うものが多いのがその理由である。

インテル系では、仮想サーバがx86系マシ

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

ンをエミュレートするVirtual Machineであるため、ゲストOSはx86で稼動するOS（たとえば、Windows、Linux、FreeBSD、NetWare、DOSなど）に対応しているものがほとんどである。

したがって、仮想サーバとして実現したいシステムが同一OSでかつUNIXである場合はUNIX系の製品を利用すべきだが、WindowsやLinuxなどの複数のOSをもつ仮想サーバを同一の物理サーバ内に実現したい場合には、インテル系の製品の利用が適している。

（5）仮想サーバ間の独立性

同一の物理サーバ内に存在する仮想サーバ同士は独立したサーバとして動作することを前提としている。データセンタなど複数の異なったシステムが同一の物理サーバ内に仮想サーバとして存在することも当然考えられる。その場合、あるシステムの仮想サーバが、故意にあるいは偶然にでも他の仮想サーバのファイルを容易に改ざんできることは好ましくない。したがって、仮想サーバ同士のリソースは、物理サーバ同士と同様、ネットワークを介したアクセス以外には不可能であることが望ましい。

（6）仮想サーバの制御方法

仮想サーバの設定状況の監視やリソース稼動状況、および設定変更やリブートなど運用時を考慮した、外部から操作可能なAPI（Application Program Interface）が用意されているかどうかもポイントとして挙げられる。動的ワークロード管理を行うときに専用の管理ツールを使用して行うだけでなく、他のシステムで使用している既存の運用管理ツールと連携し、統一した管理を行うことで管理の手間を軽減することも重要である。

製品によってはシェルから起動するコマンドを持っているものから、C/C++のAPIを公開しているものまでさまざまである。

（7）性能

まず、仮想サーバを制御するソフトウェアのオーバヘッドがある。ホストOSの有無で述べたが、このソフトウェアがホストOSを必要とするものとそうでないものがあり、ホストOSを必要としないもののほうがオーバヘッドは少ない。

NRIで実施したサーバ仮想化製品の実機評価結果、サーバ仮想化におけるオーバヘッドはホストOSを必要としないもので10%以下に抑えられていることがわかっている。

次に、仮想化されているリソースを動的に再配分したときのオーバヘッドがある。CPU、ネットワーク、ディスクのように時分割でアクセスを行っているデバイスなどではオーバヘッドは発生しないが、メモリのように時分割アクセスを行っていないものはオーバヘッドが大きい場合がある（図11参照）。

また、COMポート、LPTポート、CDやテープドライブなど仮想化されていない複数のデバイスに、仮想サーバが同時にアクセスす

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

ると大幅な性能劣化を引き起こすことや、デバイスがフリーズする可能性もあるため、同時アクセスを抑制する運用を考える必要がある。

(8) 信頼性対策

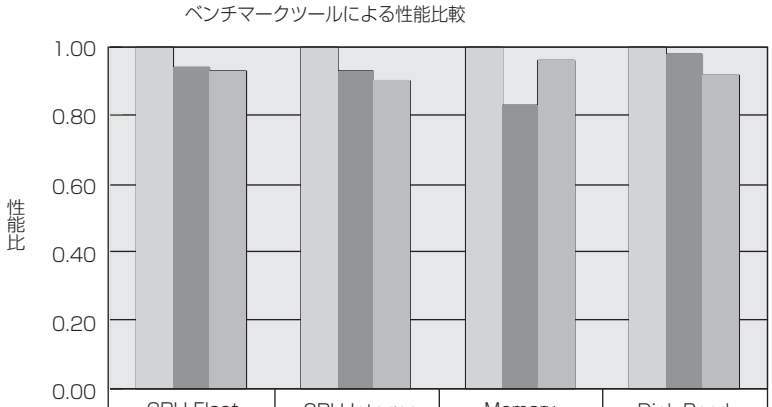
パーティショニング・仮想マシンでは、仮想サーバを制御するソフトやホストOSがフリーズした場合、その上で動作している仮想サーバもフリーズ、もしくは制御不能になってしまう。万が一の場合を考慮し、仮想サーバに対応するクラスタリングソフトの有無なども重要である。

(9) 課金単位

サーバ仮想化製品のライセンス体系として、稼動する物理マシンのCPU数に依存するもの、同時に構築可能な仮想サーバ数に依存するもの、上記を混合したものなど製品によって差異がある。最初に構築する仮想サーバ数や、使用していくうちに仮想サーバ数を増減する場合も考慮に入れる必要があるだろう。

5. 動的ワークロード管理技術の評価ポイント

ここではワークロード管理を行う上で考慮すべきポイントを示す。動的ワークロード管理は、通常の管理ツールを用いて人間がその



	CPU Float	CPU Integer	Memory	Disk Read
物理マシン上のLinux(2Way)	1.00	1.00	1.00	1.00
仮想マシン上のLinux(2Way)	0.94	0.93	0.83	0.98
仮想マシン上のLinux(1Way)	0.93	0.90	0.96	0.92

図11　仮想サーバのオーバヘッド

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

つどリソース再配分を判断するのではなく、管理負荷を削減するためにいかに自動化できるかという観点が必要である（表2参照）。

	製品C	製品D
監視可能項目	CPUのみ	CPU、メモリ
設定可能なトリガ	CPU利用率 アプリケーション独自のカスタマイズ項目	CPU利用率、メモリ利用量
再配分可能リソース	CPU	CPU、メモリ、ネットワーク、ディスク
設定可能アクション	なし	SNMP、SMTP、スクリプトの起動

表2　動的ワークロード管理技術の評価ポイント

（1）監視可能項目

動的ワークロード管理を行うため、各仮想サーバにどのくらいの負荷がかかっているかを把握できなければならない。そのためにはまずどのような仮想サーバ内のリソースが、現在どれくらい利用されているのかを把握することが必要である。具体的には、仮想サーバのCPUの利用率、仮想サーバに割り与えられているメモリの利用率、仮想サーバに割り当てられているネットワーク帯域などである。

基本的にはCPU、メモリ、ネットワーク帯域の利用率がリアルタイムに監視できれば十分であるが、仮想サーバ上で稼動しているシステムによっては、単純なCPUネック、メモリネックということだけではないこともある。そのため作りこみが発生するが独自の監視項目を作成できる製品もある。たとえばSyslogの中身や、アプリケーション固有のログなどを監視項目として設定することが可能なものである。

（2）設定可能なトリガ

上記で述べた監視可能な項目（CPU、メモリ、ネットワーク帯域）の値を収集し、リソースの再配分を行うためのロジックを動作させるための閾値を設定し、その閾値を超えたときにリソースの再配分を行う（図12参照）。

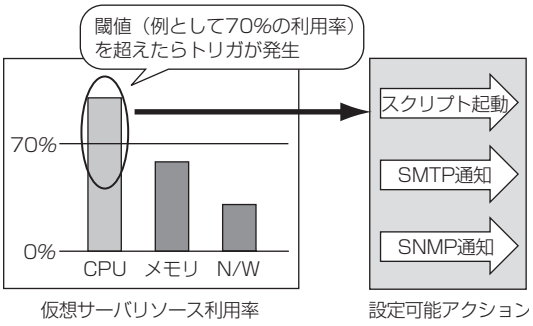


図12　トリガ、設定可能アクション

閾値の設定は、上記のCPU利用率、メモリ利用量の閾値を%で指定するような単純なものから、さまざまな監視項目を組み合わせることでより上位の閾値を設定できるものもある。より上位の閾値の例としては、仮想サーバで処理できるスループットやレスポンスタイムを閾値とするなどである。

（3）再配分可能リソース

監視可能項目で監視していた値が閾値を超えたときに、リソース配分を変更する。このときに変更できるリソースの種類によって、より多くの状況に対し、きめ細かなワークロ

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

ード管理ができる。

最低限仮想サーバに割り当てられている物理CPUの配分の変更はできる必要があるが、メモリの配分、ディスクI/O帯域の配分、ネットワーク帯域の設定を変更できるものもある。

(4) 設定可能アクション

値を監視し、値がある閾値を超えたときに、仮想サーバのリソースの変更や他のシステムへの通知などを行う。このときの仮想サーバから外部への通知方法により、他のシステムとの運用監視の統合や、連携、拡張のしやすさが異なる。

単純なものでは再配分可能リソースで述べたCPU利用率、メモリ使用量などの閾値を超えたら、それぞれ不足したリソースを確保するように設定を変更する機能のみをもつ。製品によっては、CPU、メモリの増減以外の動作を行うためスクリプトを起動することのできるものや、外部のプログラムを起動できるものもある。また、システム管理者に動作状況を知らせる方法としてSMTPを用いてメール送信するものや、他の監視ツールと連携するためにSNMPに対応しているツールもある。

6．今後の動向

ここまでは特にパーティショニング・仮想マシンのような一台の筐体を論理的に分割するサーバ仮想化技術を中心に現状の技術とその評価ポイントについて述べてきた。ここでは今後のサーバ仮想化関連技術の方向性について述べる。

今後発展する技術の1つに、ポリシーベースの運用技術がある。動的ワークロード管理では、人間の管理負荷を削減するためにシステムが自分自身にかかっている負荷に応じて適切なリソース配分を自律的に行えるようになることを目指している。これを実現するために、事前に設定したポリシーに従ってリソースの自動配分を行うような、サービスレベルを自動的に維持するポリシーベースの運用をする製品が出てきている（図13参照）。

また、パーティショニング仮想マシンの動的ワークロード管理から、複数の物理マシンへの適用も進んでいる。パーティショニング・仮想マシンのサーバ仮想化、動的ワークロード管理は、自社製品の仮想サーバ間で利用する。しかし、サーバ仮想化技術の適用先であるデータセンタには、実際には既存の複数の異なるベンダのハードウェアが数多く存在す

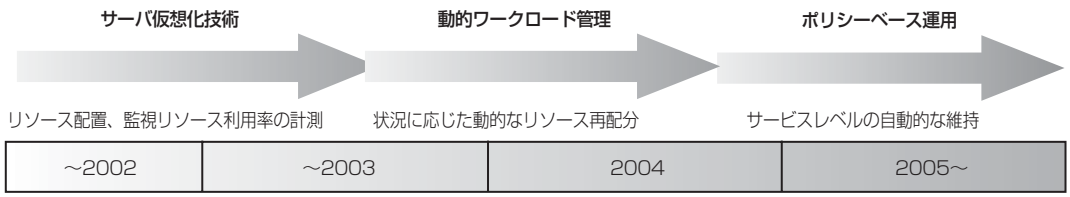


図13　今後の動向

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.

る。このようなマルチベンダのリソースを有効活用するために統一的に管理できる製品も出始めている。

また、研究レベルでは、ポリシー自体の改善を効率的に行えるように運用ノウハウデータベースと接続し、実際の設定とその結果のケースを蓄積することで、未知のパターンの処理負荷がかかってきた場合でも類似したパターンを検索してリソースの設定変更を行うように、そのつど対応しなければならない人間の判断回数を低減し、ある程度システム自身を自律的に制御させるような技術の開発が進められている。

●参考文献●

［1］VMWare 社
http://www.vmware.com/
［2］IBM
http://www.ibm.com/
［3］ヒューレット・パッカード
http://www.hp.com/
［4］サン・マイクロシステムズ
http://www.sun.com/
［5］eGenera
http://www.egenera.com/
［6］Opsware
http://www.opsware.com/
［7］Forrester Research
http://www.forrester.com/

レポートに掲載されているあらゆる内容の無断転載・複製を禁じます。すべての内容は日本の著作権法及び国際条約により保護されています。
Copyright © 2005 Nomura Research Institute, Ltd. All rights reserved. No reproduction or republication without written permission.