2009－OS－110 (5)
2009／1／28

共通指標を用いた仮想マシンの性能見積もり方式の提案とその評価

柳沢 満† 竹村 俊徳†
†日本電気株式会社 サービスプラットフォーム研究所

要旨

近年、仮想化技術を利用したサーバ統合が普及し始めている。サーバ統合後、高負荷・障害等のメンテナンスで性能の異なるサーバ間でVM(仮想マシン)のマイグレーションを行う場合、仮想化方式の違いや、サーバ間のハードウェア性能差などにより、移動後のVMの負荷を正確に見積もることは困難である。我々は、サーバの処理性能を同一基準で計測し単純比較を可能にする性能指標(例:数値演算回数)を利用したモデルを考案し、(1)VMの移動先での負荷をシミュレートすることによって事前に性能を把握する方法と、(2)VMが稼働しているサーバの空きキャパシティを計測する方法を提案する。本稿では、これらの性能見積もり方式を提案し、その評価結果と今後の方針について報告する。

A method for estimated performance of virtual machines by common index and its evaluation

Mitsuru YANAGISAWA† Toshinori TAKEMURA†
†Service Platforms Research Laboratories, NEC Corporation

Abstract

Recently, the use of server integration with virtualization technology has become popular. After servers are integrated, when virtual machine is migrated between servers for maintenance of a high load or fault, it is difficult to estimate the load of virtual machine because of difference of types of virtualization and the hardware performance between servers. We design to model using common performance index that enables a simple comparison. We propose (1) measurement method by performance simulation on the destination server that the virtual machine is migrated to, and (2) measurement method of free resource in the server. In this paper, we report evaluation result and the future works.

1 はじめに

過去、企業情報システムは大規模ホストコンピュータを中心とした中央集権的な役割を持つものが多かった。その後コンピュータ自体の性能が向上して安価で提供されたことや、Unix,Windows の登場でシステム構成が小さくなることにより、それらが拠点ごとに分散配置されて相互に接続されるようになった。システムの急速な普及と運用後の機能追加に伴う増強により、企業で負担する TCO(運用管理コスト)が増大している。そこでこれらのコストを抑えるために、サーバ仮想化技術を活用したサーバ統合が普及し始めている。

サーバ仮想化技術は 1 台の物理サーバ上に複数の仮想マシンを稼動させる技術であり、特に物理サーバの性能向上に伴って急速に利用が広まっている。XenSource,VMware,Microsoft から仮想化ミドルウェアが提供されており、広く普及している。

仮想化技術を利用するメリットの一つは物理サーバ間で仮想マシンの動的なマイグレーションを行うことで、実行中の仮想マシンを稼動中のサービスを停止することなく、別の物理サーバに移動させられることである。
近年、サーバの性能を高めるために、CPU を例とすればマルチコア/マルチソケット/HyperThreading 等の技術が提供され、これらのそれぞれが混在する構成もあり、それらがシステムのサーバで利用される。製品の中では、Intel Core i7 のような複雑な構成もある。
システム構築時に行うキャパシティプランニング技術は、ユーザ企業側のクライアント数やアクセス頻度等の情報を基にして、利用される企業情報システムのアプリケーションの処理量を見積もり、サーバ等のハードウェアで性能ボトルネックを発生させないためにスペックを決定する作業である。その目標は負荷ピーク時でもクライアントに対して処理パフォーマンスを維持することである。この落とし込みには様々な手法が存在する。
我々は仮想マシンを性能が異なる別のサーバで動作させた場合に性能ボトルネックによるアプリケーションの処理量の低下を防ぐため、その仮想マシンを移動させたと仮定した際に発生する負荷を正確に見積もる研究を進めている。
本稿ではサーバ上の仮想マシンの処理性能を同一基準で計測し、単純比較を可能にする性能指標を利用したモデルより、性能を見積もる方式をまとめ、移動先で負荷をシミュレートすることによって事前に性能を把握する方法と、別の仮想マシンが稼働しているサーバの空きキャパシティを計測する方法を提案する。
以下では 2 節で関連研究について述べ、3 節で課題について述べる。4 節では課題解決のための提案方式について述べ、5 節でその評価結果を述べる。最後に今後の課題とまとめを述べる。

2 関連研究

仮想化ソフトウェアベンダの VMware では Capacity Planner[1]が提供されており、顧客企業で運用中のサーバの稼動状況を調べ、VMware 社が持つ性能データと比較しながら、サーバ統合後の最適なシステム構成を提示することができる。
同社の VMmark[2]では、VM を稼動させるサーバ上で 6 種類のサービス(メールサーバと Java アプリケーションサーバ、スタンバイサーバ、Web サーバ、データベースサーバ、そしてファイルサーバ)を想定した計測を実施し、どのサービスがスループットを維持しやすいかを判定することが可能である。
キャパシティプランニングの例[3]として、企業情報システムのアプリケーションの TPS(Transaction Per Second)が見積もられる。大きく 3 種類の手段が存在し、クライアント数やアクセス頻度等から業務アプリケーションで処理を待ち行列としてモデル化して TPS を算出するものと、発生する処理の種類と量，評価項目などを定めたモデル・プログラムを作って、コンピュータ上で性能を計算・予測するものと、実際の本番環境で動作させる前に検証環境を準備しアプリケーションの処理で発生する負荷を再現するものがある。このようにサーバの性能を見積もるキャパシティプランニング技術を利用して、リソース使用量のボトルネックを起こさないための手法が報告されている。
参考文献[4]では、性能予測を行う際に待ち行列モデルを用いて、改善前と実際に見積もった結果と照らし合わせて評価を行っている。

3 課題

近年、データセンタや企業情報システムでは、数多くのサーバが運用されている。その中でもシステムの増強時に以前までのモデルから新しいモデルへ置換するとハードウェアとソフトウェアが異なっている可能性がある。ハードウェアに関しては購入した時期の違いなどにより、CPU/メモリ/ハードディスク/NIC 等の性能が異なる。ソフトウェアに関しては OS 自体やバージョンの違いで性能差が発生する。
導入時以外でもハードウェア障害時のパーツの交換及び、システム稼動時に購入した OS/ミドルウェア/アプリケーションのライセンスの保守契約期間などにより、新しいバージョンへ置換せざるを得ないケースがあり、その結果処理性能が変化してしまうこともある。これらの理由から、サーバ上のアプリケーションの処理能力が同一とならないため、どのサーバ上で処理させればボトルネックにならないかを的確に判断することが困難になる。
さらに、仮想化技術を利用したサーバ統合後には、高負荷・障害等のメンテナンスの理由で、ある仮想マシンを複数のサーバへ移動して稼動させることも在り得る。この場合、クライアントからの要求に対するアプリケーションの処理量が一定と仮定した場合、性能が低いサーバへ仮想マシンをマイグレーションするとその処理に掛かるサーバのリソース使用量が高くなるケースもあり、その結果アプリケーションの処理量が低下してしまう場合も想定される。
キャパシティプランニングを行う場合は、ユーザ企業に対して処理の種類や量をヒアリングし、性能を見積もる方法として待ち行列モデルを利用してサーバのリソース消費を予測するサイジング手法が存在する。これでアプリケーションの処理量を机上で見積もることは出来るが、仮想マシンはサーバへマイグレーション

することが前提なので、複数の性能が異なるサーバ上で稼動させた時にアプリケーションの処理量を見積もるには工数や費用が掛かる。この理由から複数サーバ上を移動する仮想マシン上のアプリケーションの処理量を正確かつ簡単に見積もることは難しい。このような課題に対して、性能計測を行う仮想マシンをマイグレーションすることによって、サーバの処理性能を同一基準で計測し単純比較を可能にするモデルを考案した。

4　提案方式

4.1 モデル化

アプリケーション処理を行う仮想マシンではその処理により、CPU,メモリ,NW,HDD リソースの負荷が発生している。そこで性能計測用の仮想マシン(計測 VM)は**図 4-1** のようにアプリケーション処理/サーバの空きキャパシティを想定した負荷を発生させて性能指標を求める際、以下の 1～4までのアクションを繰り返す。1)共通の性能指標(例:秒間あたりの数値演算回数)分の処理を行い、負荷を発生させる。2)サーバと仮想マシン自体のリソース使用量をサーバから計測 VM に通知する。これは負荷監視ツールを利用することで把握できる。3)仮想マシン上で発生した各リソース使用量と性能指標と対応を取る。この共通の性能指標分の処理から、どの程度リソースを消費するのかが判る。4)アプリケーションが発生したソース使用量/サーバの空きキャパシティ分のリソース使用量を目標とし、その値になるように性能指標を調整する。

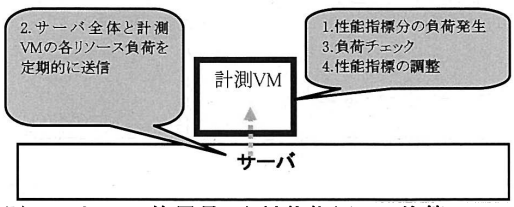

図 4-1:リソース使用量から性能指標への換算

性能の異なるサーバへマイグレーションした後の仮想マシンのリソース使用量を見積もるために、移動先のサーバ上で性能指標分の負荷を発生させることによって、そのサーバ上でのリソースの消費量を知ることができる。

また、サーバ上で利用可能なリソース使用量を見積もるためには、計測 VM をマイグレーションさせ、少量の性能指標分の負荷を出力しながら、その期間のリソース使用量を収集することにより、両者を対応づける調整を行う。その後空きキャパシティ分のリソース使用量となるように性能指標に換算する。これを複数のサーバ上で実施することで処理性能を単純比較可能な性能指標に換算できる。これによって、空リソースがどの程度存在するのか判明する。

マイグレーション後の仮想マシンの負荷を見積もるためには、各リソースを同時に負荷発生させる必要がある。

あるサーバ上でベンチマークを行った結果から利用可能なリソース使用量を見積もることを考えると、仮想マシンが NW I /O,HDD I/O の負荷を掛けることにより、I/O 処理でも CPU 時間を消費するため、計測したベンチマーク結果から複数のリソースで発生した負荷によるリソースの使用量を見積もるのは難しい。**図 4-2** は仮想化ソフトウェア(Xen)を使って DomainU から NW 出力した DomainU,Domain0 の CPU 負荷を計測した結果である。この違いはサーバの CPU 性能によって変化する。この消費を含めて**図 4-3** のように CPU のリソース使用量の目標負荷の中で他のリソースが影響していない①に相当する負荷を掛けて調整するようにする。

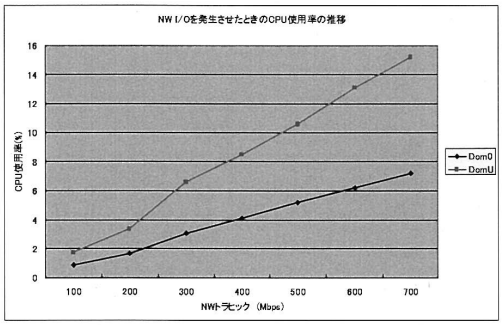

図 4-2:NW 出力負荷発生時の CPU 使用率

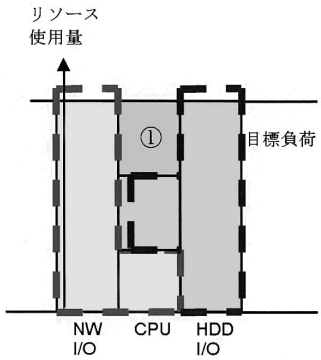

図 4-3:複数リソースの目標負荷発生

4.2 負荷発生方法

図 4-4 のように CPU の負荷の発生方法に関しては監視間隔である 1000ms 以内に演算期間と sleep 期間を入れるように調整して必ず収まるようにし、秒間

あたりの演算回数を性能指標と定義した。この演算回数を増やすと利用時間が多くなるので CPU 使用率が高くなる。演算回数を増減することより、CPU 使用率を調整することが可能になる。NW I/O 及び HDD I/O に関しても同じ原理でそれらの負荷(bit/sec)を調整することができる。
なお、メモリに関しては仮想マシンに利用可能な容量があらかじめ割り当てられており、その範囲内で利用される。また、通常運用時には頻繁に拡大・縮小されないため、本提案方式の評価では、固定サイズとする。

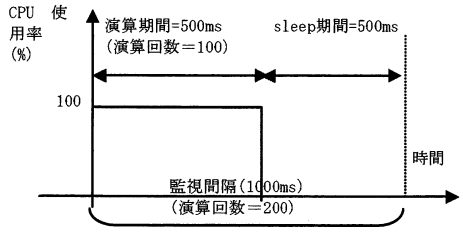

図 4-4:CPU 負荷の調整方式

4.3 負荷調整方法
計測 VM が CPU 使用率と演算回数を対応付ける調整方法として、計測フェーズと検証フェーズの 2 つが存在する。
計測フェーズ:
サーバの空リソースを利用して少量の演算回数分の負荷を発生させて、5%程度の CPU 使用率(初回負荷)を検出し、少量の演算回数から各計測方式で目標と定めた CPU 使用率に対応した演算回数を算出する。
検証フェーズ:
計測フェーズで算出した目標になる演算回数分の負荷を発生させ、CPU 使用率が許容範囲内となることを検証する。
ここで、計測 VM は CPU 使用率と演算回数の対応付けを行う際、初回に監視した負荷に誤差が混在しているのでその負荷を採用せず、安定した部分で採用する。安定しない場合は目標の CPU 使用率に合わせるため、再度調整を行い、その回数を調整回数とする。これにより、初回の負荷の読み飛ばしの処理を行い、許容範囲に入った計測が何回維持できたかの確認を行う。

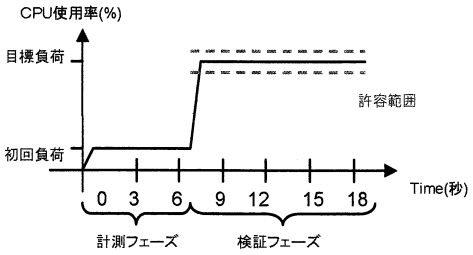

図 4-5:計測フェーズと検証フェーズの関係

4.4 計測方式の種類
4.4.1 最大ベンチマーク計測
サーバ上の仮想マシンの各リソースの最大性能を計測するために実施するもので、サーバ起動時に他の仮想マシンが存在しない状態で計測用の仮想マシン上で負荷を掛け、最大の演算回数を見積もる。

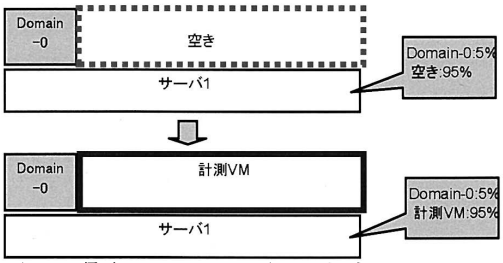

図 4-6:最大ベンチマーク計測の概念図

4.4.2 空きキャパシティ計測
サーバ上に他の仮想マシン(業務 VM)が共存する状況で、残りのリソースでどれだけ処理ができるかを調べるためのものである。少量の CPU 使用率に対する演算回数から、空きキャパシティ分の演算回数を見積もる。この計測を複数のサーバ上で実行することにより、それらのサーバの処理の余力を比較することができる。

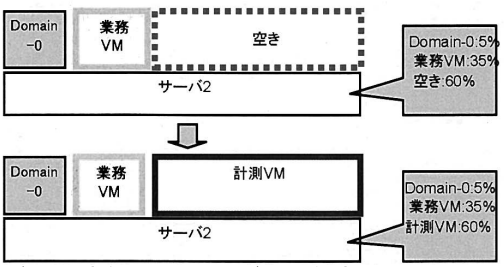

図 4-7:空きキャパシティ計測の概念図

4.4.3 負荷シミュレート
ターゲットとなる仮想マシン(業務 VM)が発生するリソ

ース使用量を共通の演算回数に換算し、移動先のサーバ上で再現することにより、実際の業務で利用する仮想マシンをマイグレーションする前にどの程度リソースを消費するかを把握できる。

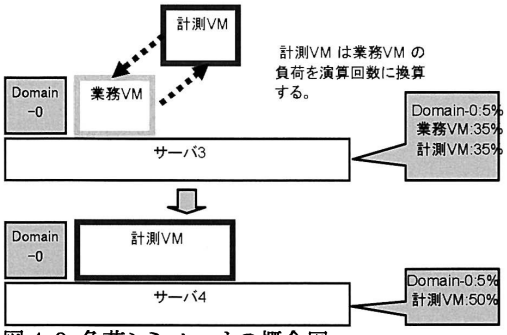

図 4-8:負荷シミュレートの概念図

5　性能評価

本稿では、モデルの検証のため、もっとも簡単な構成としてシングルコアの CPU リソースに限定して実施する。

5.1 評価環境と前提条件

評価は表の環境で実施した。

表 5-1:評価環境

SV#1	CPU	Xeon3.0Ghz シングルコア HT(ON/OFF)
	メモリ	2GB
	ネットワーク	1000BASE-T
	ハードディスク	77GB
SV#2	CPU	Xeon3.6Ghz シングルコア HT(ON/OFF)
	メモリ	2GB
	ネットワーク	1000BASE-T
	ハードディスク	77GB
SV#3	CPU	Core2Duo 2.6Ghz マルチコア
	メモリ	8GB
	ネットワーク	1000BASE-T
	ハードディスク	77GB
共有ディスク	ハードディスク	730GB
仮想化ソフト	Xen	3.1.0
	OS(Domain 0)	FedoraCore6
	メモリ(Domain 0)	512MB
性能計測 VM	OS(Domain U)	ttylinux
	メモリ(Domain U)	128MB
	ディスクイメージ	100MB

表 5-2:前提条件

SV パラメータ	
SV 負荷通知間隔	3(秒)
計測 VM のパラメータ	
初回演算回数	160(回)
許容範囲維持回数	2(回)
許容範囲	1(%)
負荷読み飛ばし回数	2(回)

5.2 評価項目

3 つの計測方式に関して以下の視点で評価を行った。

・見積もり精度

事前に計測した CPU 使用率/演算回数の関係と計測 VM が計測した値と比較。

・見積もり時間

計測 VM が計測した調整回数/時間の精度の検証。

・静的指標(ハードウェアスペック)との対応

異なるサーバ間で CPU スペック上の性能差(比率)と計測 VM が計測した性能差(比率)とを比較して同程度の精度となることを検証。

5.3 結果と考察

5.3.1 事前計測

SV#1,SV#2,SV#3 上で計測 VM を起動して性能計測プログラムを実行し、0 から 100%まで CPU 使用率を変化させた時の演算回数をグラフ化したものである。演算回数と CPU 使用率は比例関係となっている。

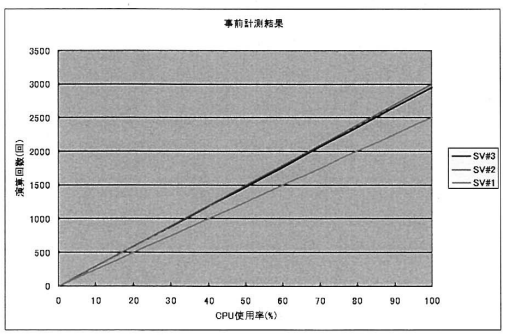

図 5-1：演算回数と CPU 使用率の関係

5.3.2 最大ベンチマーク計測

サーバ上に計測 VM のみ存在し、サーバ上で処理できる最大演算回数を見積もる。ここで演算時間は

計測 VM を対象のサーバに移動した時間とそのサーバ上で調整が完了した時間を含んだ総計である。

表 5-3:事前計測と計測結果

	SV#1		SV#2		SV#3	
	CPU 使用率(%)	演算回数	CPU 使用率(%)	演算回数	CPU 使用率(%)	演算回数
事前計測	100	2500	100	3000	100	2955
計測結果	99.4	2519	99.5	3076	99.5	2935

表 5-4:調整回数と演算時間

	SV#1		SV#2		SV#3	
	調整回数	演算時間(s)	調整回数	演算時間(s)	調整回数	演算時間(s)
計測結果	1	26	1	28	1	24

事前に計測した演算回数と調整した演算回数がほぼ合致していることを示している。SV#1,SV#2,SV#3 上では 1 回で調整できている。

5.3.3 空きキャパシティ計測

サーバ上に計測 VM と、CPU 使用率 50%分の負荷を発生させた業務 VM が存在する場合のサーバ上の空きキャパシティ分の演算回数を見積もる。ここで演算時間は計測 VM を対象のサーバに移動した時間とそのサーバ上で調整が完了した時間を含んだ総計である。

表 5-5:事前計測と計測結果

	SV#1		SV#2		SV#3	
	CPU 使用率(%)	演算回数	CPU 使用率(%)	演算回数	CPU 使用率(%)	演算回数
事前計測	50	1250	50	1500	50	1477
計測結果	49.2	1247	49.5	1527	49.9	1432

表 5-6:調整回数と演算時間

	SV#1		SV#2		SV#3	
	調整回数	演算時間(s)	調整回数	演算時間(s)	調整回数	演算時間(s)
計測結果	1	29	1	30	1	34

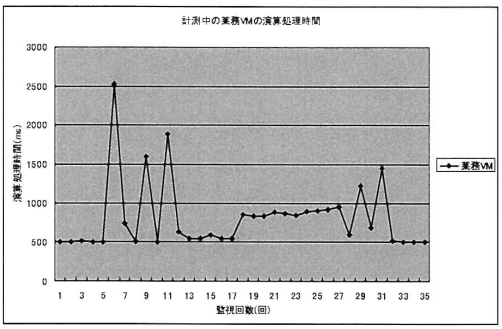

図 5-2 : 目標負荷発生時の業務 VM の演算時間

事前に計測した演算回数と調整した演算回数がほぼ合致していることを示している。**図 5-2** より計測 VM の 2 回の移動で Domain0 が CPU を消費するために、業務 VM の処理は 500ms から 2000ms まで延長してしまう。実際の空きキャパシティ分の負荷を出力させると、CPU 時間の競合で 500ms の処理が 1000ms 近辺まで延長した。この結果からシステムで決められた処理時間があるならばそれに到達する手前までに空きキャパシティ分の演算回数を見積もる必要がある。

5.3.4 負荷シミュレート

移動元サーバで演算回数を見積もり、移動先のサーバでその演算回数分の処理を行う。移動元サーバ上で計測 VM が調整する CPU 使用率は 20%とする。なお、全体時間は以下の1～8 までの処理に掛かった総計である。移動元サーバ上での調整処理と移動先サーバ上での負荷発生を含めたものは以下のフローとなる。

	初期 VM 配置条件 移動元サーバ:業務 VM 移動先サーバ:なし。 退避先サーバ:なし。 プールサーバ:計測 VM
1	移動元サーバ上の業務 VM が退避先のサーバに移動する。
2	プールサーバ上の計測 VM が移動元サーバに移動する。
3	計測 VM は移動元サーバ上で負荷を調整する。
4	移動元サーバ上の計測 VM がプールサーバに移動する。
5	退避先のサーバ上の業務 VM が移動元サーバに移動する。(計測前の初期状態と同じ。)
6	プールサーバ上の計測 VM が移動先サーバに移動する。
7	計測 VM は調整済みの演算回数分の負荷を出力する。
8	移動先サーバ上の計測 VM はプールサーバに移動する。(計測前の初期状態と同じ。)

1 で移動させずに計測 VM が調整処理を行うと、このアプリケーション処理を行う仮想マシンにリソース競合により、影響が発生する可能性がある。

表 5-7:移動元での調整結果

	SV#1		SV#2	
	CPU 使用率(%)	演算回数	CPU 使用率(%)	演算回数
事前計測	20	500	20	600
計測結果	19.9	503	20.1	621

表 5-8:移動先でのシミュレート結果

	SV#2		SV#3	
	CPU 使用率(%)	演算回数	CPU 使用率(%)	演算回数
計測結果	16.4	-	21	-

表 5-9:調整回数と全体時間

	SV#1→SV#2		SV#2→SV#3	
	調整回数	全体時間(s)	調整回数	全体時間(s)
計測結果	1	83	1	77

SV#1(移動元)→SV#2(移動先)
事前計測の結果から SV#2 の方が CPU の性能が高いため、同じ処理を実行させると、CPU 使用率は 20%→16.4%に少なく観測された。
SV#2(移動元)→SV#3(移動先)
事前計測の結果から SV#3 の方が CPU の性能が若干低いため、同じ処理を実行させると CPU 使用率は 20%→21%となり若干多く観測された。
調整による計測 VM 及び、業務 VM の移動に時間がかかるため、全体時間が掛かっている。この移動時間を短縮する必要がある。

5.3.5 全体

SV#1 と SV#2 で演算回数を比較すると CPU のクロックのみ異なるので、周波数の比率になる(SV#1:SV#2=1:1.2)。SV#3 自体はマシンのアーキテクチャが異なるため、SV#2 と比較すると同程度になり、周波数の大小の特性だけでは正確に見積もれないことが判明した。

6 おわりに

最終的な課題に挙げた仮想マシン上のアプリケーションの処理量の再現として CPU、メモリ、HDD I/O、NW I/O リソースを複合的に計測して同一基準で比較可能なモデルを提案し、その一部を評価した。
共通の性能指標を使用して同一基準で評価することにより、30 秒程度で目標になる CPU 使用率に対する演算回数を見積もることができており、見積もった演算回数は事前計測した値と近似しており、CPU 使用率に関しては目標負荷の 1%以内で調整できることを確認した。
HDD I/O、NWI/O を発生させると CPU 負荷が発生する。負荷の大きさによってはサーバ全体で CPU リソースのボトルネックが発生する可能性がある。これを検知・回避する仕組みを入れる必要がある。
マルチ CPU を想定した場合には実 CPU 分の演算スレッドを使って負荷を出力させることにより、そのサーバ上の実 CPU ごとに負荷を掛けるようにする。
空きキャパシティ計測では CPU 使用率を再現させると、他の仮想マシンへの影響が発生してしまうため、見積もり精度を向上し、他の仮想マシンへの影響を抑える施策を検討する必要がある。
負荷シミュレートでは移動時間の短縮の施策として計測 VM の軽量化を行う。
同じサーバの性能であっても利用する仮想化ミドルウェアの方式によって、結果も異なると思われるので、この評価と検証も実施する予定である。

参考文献

[1] Capacity Planner
http://www.vmware.com/jp/products/capacity_planner
[2] VMmark
http://www.vmware.com/jp/products/vmmark/
[3]キャパシティプランニングの進め方
http://itpro.nikkeibp.co.jp/article/COLUMN/20060807/245309/
[4] 河野 知行,森本 舞,簡単なシミュレーションによるボトルネック箇所の特定と改善効果の判定(CPU とディスクの性能評価) 2002(55),p.43-48,20020607(ISSN 09196072)情報処理学会研究報告. EVA, [システム評価]
[5]川西 俊之,紀 一誠,即時・待時混在システムの階層的性能評価法Vol.2007, No.63(20070622) pp. 15-22,情報処理学会研究報告. EVA, [システム評価]
[6] 木下 俊之, 高橋 幸雄, 梅原 元, 資源要求のある待ち行列網のモデル化と評価, Vol. 第 53 回平成 8 年後期, No. 1(19960904) pp. 153-154 社団法人情報処理学会
[7] 木下 俊之, 高橋 幸雄, 逐次アクセス資源のある計算機システムの待ち行列網による近似評価の一手法, Vol. 42, No. SIG_14(TOM_5)(20011215) pp. 1-13 社団法人情報処理学会 ISSN:03875806
[8]性能評価の基礎と応用, 共立出版 亀田 寿夫 (著), 李 頡 (著), 紀 一誠 (著)