

サーバ処理性能の評価方法に関する一検討

岡田 昭宏

E-mail: okada.akihiro@lab.ntt.co.jp

ITシステムは機能と性能の2つの側面で評価できる。ITシステム開発においては機能の側面が優先され性能の側面が軽視されがちであるが、企業活動の効率低下や顧客獲得機会の損失等の影響を考慮すれば、性能に関しても十分な検討が必要である。しかし、性能に関連する要因はITシステムを構成する様々な要素にわたっており、高精度な性能設計には様々な知識、経験を要する。筆者らはこれらの問題を解決するためにITシステムの性能評価・診断フレームワークを検討している。同フレームワークではシミュレーション技術を1つの中核としており、ITシステム構築の早い工程で精度の高い処理性能の評価が可能である。本稿はこのフレームワークのなかのサーバ処理性能について、これまでの検討状況と今後の検討課題をまとめたものである。

キーワード ITシステム, サーバ処理性能, シミュレーション

A Study of Server Performance Evaluation

Akihiro OKADA

E-mail: okada.akihiro@lab.ntt.co.jp

IT systems should be evaluated with the following two aspects; functions and performance. Performance issues cause enterprise degradation and loss of business chances. Therefore, solutions to improve IT system performance as well as functions must become more important. Because there are many factors that influence performance of an IT system and the factors influence mutually, we adopt simulation tools which can simulate IT system behavior to evaluate performance of IT systems. This paper describes how to evaluate server performance by using simulation tools and summarizes further studies about this subject.

Keywords IT system, server performance, simulation

1. はじめに

企業活動の効率化、迅速化への要求はますます高まっており、その実現に向けてITシステムの役割がますます重要になっている。ITシステムの重要度が増すにつれて、トランザクション処理の応答時間増大といったITシステムの処理性能低下が直接企業活動のパフォーマンス低下に直結するようになる。ITシステムの処理性能はシステム的设计または更改時にあらかじめ作りこまれるべきものであるが、実際にはシステムの機能実現が優先され目標性能の実現が後回しになることが少なくない。また、性能を左右する要因が多岐にわたるといった性能設計の技術的な困難さから、十分な性能設計の検討を行っても目標とする処理性能が得られないことも少なくない。

目標とする処理性能を達成するための方法としては、プロアクティブなアプローチとリアクティブなアプローチとがある。プロアクティブなアプローチは、ITシステム構築と並行して処理性能を作りこんでいく進め方であり、代表的な手法としてITシステムの動作を模擬するシミュレータを用いて処理性能の事前評価をおこなう方法がある。ITシステム

のシミュレータの代表的な例としてはOPNET社のOPNET[1]、HyPerformix社のHyPerformix[2]、COMPUWARE社のVantage[3]がある。リアクティブなアプローチは、ITシステム構築後に実際のITシステムの処理性能を測定し問題点を突き止めて対処する進め方である。例えば、負荷エミュレータで擬似的な負荷を発生させて、実際のサーバやNWの負荷を測定・分析していくなかで処理性能のボトルネックを明確にし、処理性能向上に向けた対処をしていく。システムの総合評価環境の代表的な例としてはIBM社のRational[4]、負荷エミュレータの例としてはMERCURY社のLoadRunner[5]、負荷の測定・分析ツールの例としてはアイ・アイ・エム社のES/1 NEO[6]がある。

処理性能の問題に限らないが、システム構築の過程で生じる問題は、できるだけ早い工程で発見し対処する必要がある。一般的に後工程で問題が発覚するほどその対処に時間とコストを要するためである。したがって、ITシステム構築の際には、プロアクティブなアプローチを取り入れ、早い段階で処理性能を作りこんでいかなければならない。そのためには、ITシステムを実際に構築する前に処

理性能を評価することが可能なシミュレーション技術が適用しており、シミュレーション技術を効果的に用いることで、低コストで高い精度の処理性能評価が期待できる。

このような背景のもとで、IT システムの性能評価を簡易にかつ精度よくおこなうためのフレームワーク[7-12]の検討がおこなわれている。このフレームワークはシミュレーション技術を1つの中核としており、プロアクティブな性能評価が可能である。このフレームワークではリアクティブなアプローチについても検討を行っているが、本稿ではシミュレーション技術を用いたプロアクティブな性能評価に焦点を絞ることとする。

2. サーバ処理性能評価の周囲状況

2.1. サーバ処理性能評価の必要性

IT システムは、大きく分ければデータを処理するサーバとデータを流通させるネットワーク(以降 NW)の2つから構成される。最近のIT システムの普及状況や、NWとサーバの高速化のアンバランス(経験則ではあるが、NWの高速化はギルダールの法則と呼ばれ6~9ヶ月で2倍、サーバの高速化はムーアの法則と呼ばれ12~18ヶ月で2倍と言われており、NWはサーバの2倍のペースで高速化していることになる)から、今後サーバがボトルネックになるケースが次第に増えてくると予想できる。

このような考察のもとで、本稿では今後重要性が高まっていくと思われるサーバの処理性能評価方法について検討していく。

2.2. サーバ処理性能に関する検討テーマ

サーバの処理性能に関する検討には、サーバ処理性能を向上させるための検討と、サーバ処理性能の評価方法の検討とがある。前者の性能向上の検討はハードウェア、OS、ミドルウェア、等の改良による高速化を目指した検討である。後者の評価方法の検討は与えられたハードウェア上でアプリケーションを実行させる時にどの程度の処理性能が得られるかを推定するための検討であり、IT システム構築時にどの程度のスペックのハードウェアが必要かを見積もる際に必要な技術である。

本稿は、与えられたシステムにおいてどの程度の処理性能が期待できるかを評価することを主題としており、後者、つまり性能評価手法に関する検討をおこなう。

2.3. サーバ処理性能の検討範囲

サーバ処理性能を考える際には、アプリケーションを含まないサーバ単体の単純な処理能力を考える場合と、アプリケーションを含めたIT システム全体の実行能力(例えば、トランザクション処理性能)を考える場合とがある。両者を厳密に区別することはしないが、前者の単体の能力ではCPUの処理能力を例に取るとMFLOPS(1秒間に実行できる浮動小数点演算の回数)や動作クロックなどの低いレイヤ(物理レイヤ寄り)での指標を用い、ハードウェアの

評価に向いている。後者のIT システム全体の能力では実際のアプリケーションの処理時間やスループットといった高いレイヤ(アプリケーション寄り)における総合的な指標を用い、IT システム全体の処理性能の評価に向いている。

本検討はIT システム全体の処理性能の評価を最終的な目的としているため、サーバのハードウェア単体の性能評価ではなく、サーバ上で実行するアプリケーションの実行能力の性能評価を検討課題とする。ただし、性能評価を行う過程で前者の指標を用いることもあるため、低レイヤの性能評価とは無関係ではない。

本稿では、単に「サーバの処理性能」といった場合、アプリケーションのレイヤも含めた総合的な処理性能を意味することとする。

2.4. サーバ処理性能を決める要因

サーバの処理性能に影響する要因は大きく分けてハードウェアの処理能力とソフトウェア実行時の処理負荷に分けられる。

ハードウェアの主な構成要素として中央処理装置(以降CPU)、ハードディスクドライブ(以降HDD)、メモリがある。処理性能向上のためには、処理性能の高いCPUを用いたり、複数のCPUを並列に動作させるなどの方法や、HDDの情報転送速度を上げたり、複数のHDDを用いてRAID(Redundant Arrays of Independent(or Inexpensive) Disks)を構成する方法などがある。

ソフトウェアの処理負荷には、CPUへの負荷、HDDへの転送量、メモリの使用量等がある。ソフトウェアの主な構成要素としては、オペレーティングシステム(以降OS)、ミドルウェア、アプリケーションがある。処理性能向上のための検討として、OSやミドルウェアの設定を最適化するためのチューニングや、アプリケーションのアルゴリズム改善といった検討課題があるが、前述の通り本稿では検討の対象とはしない。

2.5. サーバ処理性能の評価手法の現状

サーバの処理性能を評価する手法には、(手法a) アプリケーションの動作を詳細に把握(ハードウェアもしくはソフトウェアで実現するプロンプトにより測定[13])し分析する手法、(手法b) サーバの内部構造を待ち行列理論等に基づいてモデル化し解析またはシミュレーションで分析する手法、(手法c) ベンチマーク値等のマクロな指標等を用いておおよその値を推定する手法などがある。図1は各手法が、性能評価をおこなうにあたってハードウェア、ソフトウェアに関する詳細な情報をどの程度必要とするかを大まかに示したものである。

詳細な測定に基づく手法(a)は、実際の環境を詳細に分析することで性能評価を行う手法である。そのため、評価するアプリケーションの実行ファイルが存在し、それを実行するサーバも整っており、加えてアプリケーションの動作を

詳細にトレースできる環境が必要である。この制約により、すでに構築したITシステムの性能改善に有効ではあるが、構築中のITシステムの性能評価への適用が難しいため本検討の目的には沿わない。

論理的解析による手法(b)は、アプリケーションの負荷(例えば、CPU 使用時間、HD アクセス量)を測定または予測し、サーバの構造をモデル化することで、解析またはシミュレーションにより処理性能を推定する手法である。推定の精度はモデルの正確さに依存するが、精密なモデルの構築にはコストがかかるためトレードオフとなる。

マクロな指標による方法(c)は、サーバの内部構造を考慮せずブラックボックスとして扱い、外部に見えてくるサーバの性質を捕らえ、その性質を利用して推定する手法である。例えば、サーバのベンチマーク値と処理能力が比例すると仮定して、あるサーバでのアプリケーションの処理時間から、任意のサーバでの処理時間を推定する。高い推定精度は期待できないが、最も簡易な手法である。

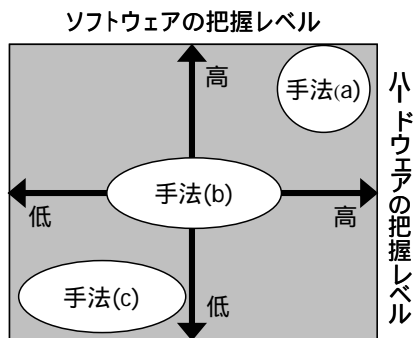


図1 各評価手法の位置付け

2.6. プロアクティブなサーバ処理性能評価手法の例

ここでは、プロアクティブな性能設計に適しており、詳細な評価が可能な手法(b)を用いたサーバの処理性能評価手順の一例を示す。

本手順ではITシステムの動作を模擬するシミュレータとしてOPNETとそのサーバ性能評価用追加モジュールであるSCE(Server Characterization Editor)とSSM(Advanced Server Specialized Model)を用いている。

おおまかな手順は以下のとおりである(図2)。

1. アプリケーションの処理負荷を決定
2. アプリケーションの実行パターンを決定
3. サーバの構成を決定
4. シミュレーションの条件設定
5. シミュレーションの実行と結果の考察以降、それぞれの手順について簡単に説明する。

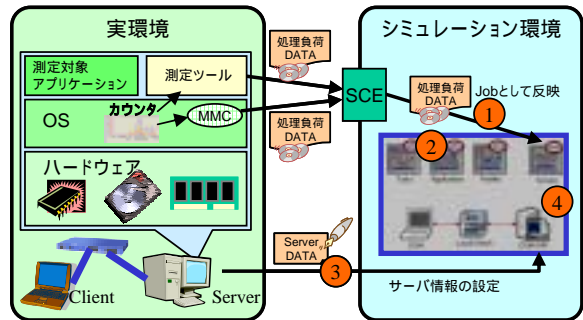


図2 サーバ処理性能評価環境の例

2.6.1. アプリケーションの処理負荷を決定

アプリケーションを実行させ、その処理負荷を測定する。OPNETでサポートしている処理負荷の測定ツールとしてはNetIQ社のAppManager[14]、CONCORD社のSystemEDGE[15]、HP社のOpen View Performance[16]、BMC社のPATROL[17]がある。

また、サポート外であるがMicrosoft Windowsに標準で実装されているMMC(Microsoft Management Console)を用いる方法もある。この方法は、以下の特徴がある。

- ・ OSに標準で実装されており、測定に向けた特別な準備が不要であるためアプリケーションのインストールおよび設定にかかる手間とアプリケーションのコストが抑えられる。
- ・ 処理負荷が軽いため測定に影響を与えにくい。
- ・ 短い周期での測定が可能である。
- ・ MMCの出力形式をOPNET向けの形式に変換する手順が別途必要となる(ツールを自作して対応した)

表1にMMCを用いて収集する項目の一例を示す。

表1 MMC収集項目の一例

項目名	説明
Process(*)\ID Process	プロセスのID
Process(*)\% Processor Time	プロセスの名称
Process(*)\%IO Read Operations/sec	プロセスのCPU使用率
Process(*)\%IO Write Operations/sec	HDD読み回数
Process(*)\%IO ReadBytes/sec	HDD書き込み回数
Process(*)\%IO WriteBytes/sec	HDD読み量 (byte/sec)
Process(*)\Virtual Bytes	HDD書き込み量 (byte/sec)
Process(*)\Working Set	使用仮想メモリ容量

2.6.2. アプリケーションの実行パターンを決定

アプリケーションが実行されるタイミングを決める。実際にアプリケーションが利用されるときの呼び出しのパターンを考慮する必要がある。例えば、実行間隔がランダムとみなせれば指数分布を指定すればよい。アプリケーションの実行頻度を変化させることで、数年先の想定負荷のもとでの処理性能をあらかじめ評価することもできる。

2.6.3. サーバの構成を決定

サーバの構成を指定する。指定する値は代表的なベンチマークである SPEC CPU2000[18]の値, HD のインターフェースの種類, RAID の構成, HDD の規格, 等がある。代表的なサーバの構成はシミュレータに登録されているため, リストから選択するだけで設定できる。サーバの構成を任意に変化させれば, 任意のサーバでの処理性能を評価できる。

2.6.4. シミュレーションの条件設定

OPNET では CPU 処理時間を求めるために CPU 処理性能の代表的なベンチマークである SPEC CPU2000 を用いている。SPEC CPU2000 自身は8つのベンチマーク値を持っているため, この8つのなかのどのベンチマーク値を用いるかをシミュレーションの条件として指定する必要がある。8つの内訳は「整数演算」のベンチマークと「浮動小数点演算」のベンチマーク, コンパイラオプションの制限の「あり」と「なし」, 「シングルタスク処理」におけるベンチマークと「マルチタスク処理」におけるベンチマークの3つの項目の組み合わせであり, $2^3=8$ 個となる。通常は「整数演算」, 「コンパイラオプション制限あり」, 「マルチタスク処理」の場合のベンチマーク値を用いる。アプリケーションが浮動小数点演算を主におこなっていることがわかっていれば, 浮動小数点演算のベンチマーク値を用いればよい。

2.6.5. シミュレーションの実行と結果の考察

図3は, 現状(CPUが1つ)と対策実施後(サーバのCPUを2つに増設)のCPU使用率の比較をおこなった場合の例を示したものである。現状のCPU使用率がたびたび100%に達している状況であるが, 対策によってCPU使用率に余裕を持って運用できるようになることがわかる。

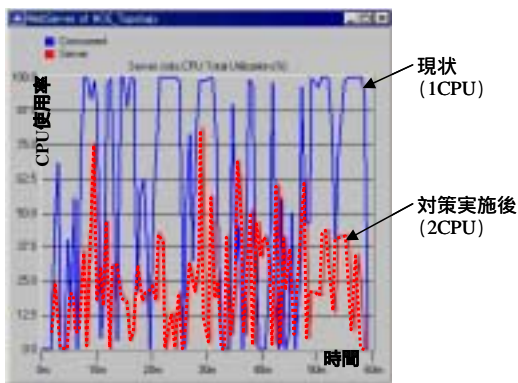


図3 シミュレーション結果の一例

2.7. 現状の問題点

サーバの内部構造は非常に複雑であり, 簡単な待ち行列モデルで表現することは困難である。実際, あるサーバで測定したアプリケーションの処理負荷の情報から別のサーバにおける処理時間をベンチマーク値から推定すると大きな誤差が生じることがある。一例として, SPEC CPU2000 の結果を紹介する。SPEC CPU2000 は SPECint

が12個, SPECfp が14個の異なるベンチマークの結果から得られるベンチマーク値の幾何平均で定義されている。図4はそれぞれのベンチマークの結果が, 異なる2台のサーバ間でどのような結果になっているかを示したものである。図中の値はサーバA (Pentium4(3.06GHz)) のベンチマーク値に対するサーバB (Athlon XP 3200+) のベンチマーク値の比である。サーバAとサーバBはSPECintがほぼ等しいものを選んだが, ベンチマーク"189.lucas"ではベンチマーク値の比が0.67倍となっており, これはサーバBの処理性能がサーバAの処理性能の0.67倍であることを意味している。一方, 別のベンチマーク"300.twolf"では比が1.38倍となっており, サーバBの処理性能がサーバAの処理性能の1.38倍となる。この結果から, SPECintが同じ値のサーバであっても, 評価するアプリケーションが異なることで処理時間に約2倍 ($1.38 / 0.67 = 2.06$) の開きが生じてしまうことがわかる。このような例から, サーバを変更したときの処理性能の予測が簡単でないことがわかる。

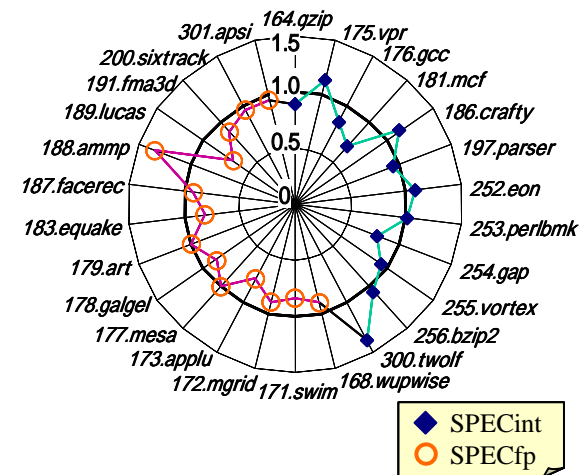


図4 ベンチマーク値の比の比較

2.8. 技術的課題

2.7で述べた様な課題を解決するためには, シミュレーションによる評価に加えて, サーバとアプリケーションに関する処理性能に関連の深い特性を把握したうえで, その特性を加味した評価をおこなう必要がある。

シミュレーションを行う際のパラメータとしては, CPU や HDD 等のスペックや, アプリケーションの負荷に関する情報を与えているが, それだけでは十分ではない。例えば, サーバの特性として CPU 特有の性質 (例えば, L1, L2 キャッシュの量, パイプラインの深さ, 分岐予測の精度, 等) や, HDD 特有の性質 (例えば, 内周と外周の速度差, プラッタサイズ, キャッシュの利用方法) といった項目を考慮して, より正確に評価するための方法が必要となってくる。

今後の課題は, 性能評価にどのようなパラメータを用いるべきかを精査し, そのパラメータを用いて性能評価をより正確におこなう方法を確立していくことである。

3. サーバ処理性能の評価精度向上に向けた検討

2章ではサーバ処理性能の評価手法の現状を述べた。本章では今後の評価手法について検討する。

3.1. 検討が対象とする場面

検討の対象は、アプリケーションの処理負荷が測定または予測により明確になっており、与えられたアプリケーションの実行頻度のもとで、与えられた処理性能(応答時間, スループット, 等)を満たすサーバの構成を求めることとする。アプリケーションの改善や OS, ミドルウェア等のチューニングによる性能向上は検討対象としない。

本検討の実際の利用場面としては、IT システムを新規開発する際のサーバ構成の決定、またはサーバ更改による IT システムの性能改善の際のサーバ構成の決定が想定できる。

3.2. 検討の目的

本検討の目的は、「簡易な方法」を用いて「十分な精度」でサーバの処理性能を評価するための技術を確立することである。

「簡易な方法」は、サーバとアプリケーションに関するいくつかの特徴を把握し、それらの値をパラメータとした簡単な計算で求められる方法と定義する。取得したパラメータを変数とした関数を作成し計算することで求めても良いし、IT システムのシミュレータを用いて求めても良い。ただし、本検討の最終的な目的はサーバを含めた IT システム全体の性能評価環境の構築であるため、この最終的な目的を見据えた方法である必要がある。

「十分な精度」については、感覚的ではあるが、2~3割程度の誤差は許容し、2~3倍の誤差は許容しないと云った精度を目標とする。サーバの必要スペックの見積もりは、ネットワークの必要帯域などの見積もりと比較すると大きな誤差を伴う場合が多い。これは、前述の通りサーバが多くのパーツ(CPU, メモリ, HDD 等)から構成され、それぞれが処理性能に複雑に絡み合っており、正確な処理性能の予測を困難にしているからである。このようなことから、実際のシステム開発ではサーバの見積りに大きな安全率を見込むことが多いため、このように考えた。

3.3. 検討手法

サーバのハードウェアおよびソフトウェアに関するパラメータ p (ベクトル) と関数 f から処理時間 T が求められるものと仮定する(式1)。ただし、処理時間 T を正確に求めるためにはパラメータ p の値の決定や関数 f の計算が非常に難しいものになると思われる。

$$T = f(p) \quad (\text{式1})$$

今回検討する手法は、パラメータ p のなかから測定が容易で処理性能への影響が大きいパラメータ p' を抽出し、

計算が容易な関数 f' を用いることで処理時間 T を推定する手法である(式2)。

$$T \approx f'(p') \quad (\text{式2})$$

f' は、2.5 で述べた通り、サーバの内部構造をモデル化し解析またはシミュレーションをおこなうマイクロな手法(手法(a),(b))とベンチマーク等の結果から概算するマクロな手法(手法(c))とがある。マイクロな手法は、精密なモデルを用いることで高い精度を得ることが期待できるが、場合によっては実機を用意して実測する以上のコストがかかることも考えられるため、ある程度の精度に止める必要がある。一方、マクロな手法は、内部構造等を考慮しない簡易な方法であるためコストは低く抑えられるが、高い精度を得るとは難しい。

今回検討する手法はマクロな手法とマイクロな手法の両方を組み合わせ両者の長所を生かした手法である。性能評価のベースとなる値をマイクロな手法であるシミュレーションを用いて求めることで精度を確保し、マクロな手法でシミュレーションでは反映しきれない補正をおこなう。

補正を行うタイミングは、補正方法(a) シミュレーションのパラメータを補正する方法と、補正方法(b) シミュレーション結果を補正する方法とがある(図4)。シミュレーションのパラメータを補正する方法(補正方法(a))では、シミュレーション中のサーバの振る舞いが補正でき、サーバ間の通信状況なども補正されるため評価結果の精度向上が期待できる。シミュレーション結果を補正する方法(補正方法(b))では、補正するパラメータを変更するだけであればシミュレーションを再度実行する必要がないため短時間で多くのパターンの評価が可能となる。

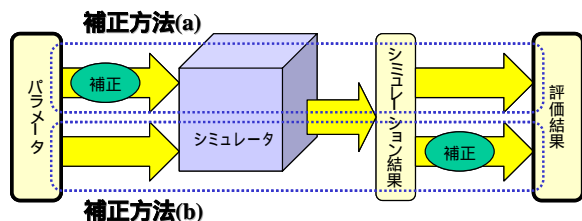


図4 補正方法のイメージ

本検討は IT システム全体の性能評価を最終的な目標としているため、サーバの振る舞いを補正可能な、シミュレーションのパラメータを補正する方法(補正方法(a))を選択する。

3.4. 今後の検討課題

3.4.1. 使用するパラメータの決定

補正に用いるパラメータを決定する必要がある。パラメータは、処理性能の推定に有効で、値の決定が容易に可能である必要がある。

考えられるパラメータの一例としては、アプリケーションの演算処理が整数演算を主体とするか浮動小数点演算を主体とするかという尺度が考えられる。CPU の処理能力

のベンチマークである SPEC CPU2000 では整数演算と浮動小数点演算の処理能力を分けて評価している。これは、最近の CPU が主に浮動小数点演算の高速化を狙った拡張機能 (MMX(Multi Media eXtension), SSE(Streaming SIMD Extension)等) を実装し、浮動小数点演算の処理能力を優先して高速化し続けていることから、CPU の演算処理能力を一くりに表せないためである。実際、SPEC CPU2000 の整数演算のベンチマーク SPECint と浮動小数点演算のベンチマーク SPECfp との値をいくつかのサーバについて調べてみると、両者の値の比が CPU の種類によって大きく異なることがわかる (図 5)。

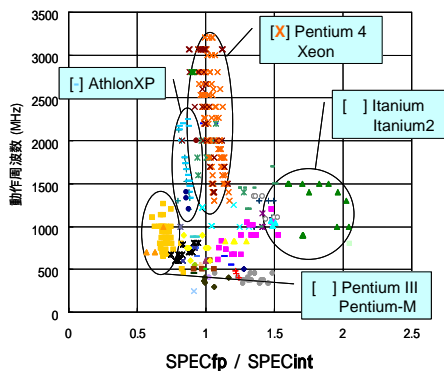


図 5 整数演算と浮動小数点演算のベンチマーク値

この差を考慮するためには、アプリケーションの処理内容を分析するなどして整数演算処理と浮動小数点演算処理の内訳を調べ、シミュレーションの補正を行う必要がある[x]。

3.4.2. 簡易化

IT システムを作成する前に性能評価をおこなうためには、アプリケーションに関するパラメータを何らかの形で見積もらなければいけないが、精度良い見積もりには IT システムに関する深い知識と経験が必要である。特に、アプリケーションの仕様が固まる以前の大まかな見積もりであればなおさらである。そこで、本検討の今後の課題の1つとして、多少精度は落ちて簡易に処理性能の見積もりができるフレームワークの作成を検討している。

例えば、アプリケーションの種類等の容易に判別できる情報とアプリケーションに関するパラメータとの関係をあらかじめ大まかに把握しておくことで、アプリケーションの種類を特定するだけである程度の精度で推定が可能となる。この様なフレームワークを作ることによって、性能設計の経験がなくてもある程度の精度で推定ができる。図 6 は単なるイメージであるが、アプリケーションの種類とパラメータ値との関係をマッピングしたものである (各アプリケーションが実際にどこに位置するかは今後検証していく)。

4. おわりに

現状のサーバ処理性能の評価手法についてまとめ、その一例を示すとともに、今後のサーバ処理性能の評価方法の方向性を検討した。

評価精度向上と簡易化のバランスをとりつつ検討を進めていきたい。

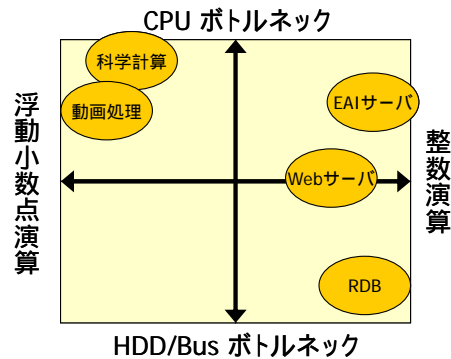


図 6 アプリケーションの種類とパラメータ

文 献

- [1] OPNET, <http://www.opnet.com/>
- [2] HyPerformix, <http://www.hyperformix.co.jp/>
- [3] Vantage, <http://www.compuware.co.jp/products/vantage/vantage.html>
- [4] Rational, <http://www-306.ibm.com/software/rational/>
- [5] LoadRunner, <http://www.mercury.co.jp/products/loadrunner/>
- [6] ES/1 NEO, <http://www.iim.co.jp/es1/neo.html>
- [7] H. Yamada, T. Yada, "Information Technology System Architecture Planning Platform (ITAP)," NTT REVIEW, Vol.13, No.5, pp.78-84, 2001.
- [8] 井上, 山田, 矢田, "IP 系ネットワークサービス / IT システムの設計・性能評価技術フレームワーク," NTT R&D, Vol.52, No.2, pp.134-140, 2003.
- [9] 矢田, 川口, 山田, "IT システムの性能設計におけるビジネスプロセスモデリングとビジネスプロセス駆動型シミュレーション手法," NTT R&D, Vol.52, No.2, pp.141-147, 2003.
- [10] 川原, 矢田, 川口, 山田, "アプリケーショントラヒックプロファイリング手法," NTT R&D, Vol.52, No.2, pp.148-153, 2003.
- [11] 山田, "コンテンツスイッチ型負荷分散トラヒック制御法に関するシミュレーションモデリング," NTT R&D, Vol.52, No.2, pp.154-160, 2003.
- [12] 山田, "分散コンピューティング環境におけるアプリケーションレベルプロトコルのシミュレーション性能評価," NTT R&D, Vol.52, No.2, pp.161-167, 2003.
- [13] 片山, 他, "ハードウェアトレーサを用いた性能解析ツールの概要," 情報処理学会第 45 回全国大会, 7P-1, 1992.
- [14] AppManager, http://www.netiq.co.jp/products/appmanager/app_index.htm
- [15] SystemEDGE, <http://www.empire.com/products/systemedge/>
- [16] Open View Performance, <http://www.openview.hp.com/index.html>
- [17] PATROL, http://www.bmc.com/ja_JP/
- [18] SPEC CPU2000, <http://www.spec.org/cpu2000/>
- [19] 岡田, "IT システム性能評価におけるサーバ性能推定手法の検討," 信学全大, B7-27, March 2004.