

Java アプリケーションサーバーの 仮想化オーバーヘッドの測定

高橋 一晃^{†1} 大井 仁^{†1}

本報告では J2EE のベンチマークである SPECjAppServer2004 のサーバー群を仮想化によって異なる 2 つの構成で統合し、その際のオーバーヘッドを測定した。また、トランザクション毎に計測した CPU 時間から、SPECjAppServer2004 のシステム全体への負荷を予測し、3 つの異なる割合のトランザクションにおける予測結果と実測結果の比較をする。

CPU 使用率がスケーリングファクターと似たカーブを描くと予測していたが、特に 2 つを同じ仮想マシンに統合した構成では、予測値と実測値に相違が見られた。

Measurement of Virtualization Overhead in a Java Application Server

KAZUAKI TAKAHASHI ^{†1} and HITOSHI OI ^{†1}

In this technical report, we present our measurement results of virtualization overhead in SPECjAppServer2004, which is a benchmark suite of a 3-tier J2EE server system. By breaking down the CPU time for each transaction, we predict the total CPU utilization in two different consolidated configurations under three transaction mixes.

While predicted CPU utilization draws similar curves to the throughput over the system scaling factor, utilization of each virtual machine exhibits differences between predictions and measurements, especially while two of servers are placed in the same virtual machine.

1. はじめに

仮想化技術は複数のシステムを一つの物理的なシステムに統合することを可能にする技術である。複数のシステムで一台の物理システムを共有することにより、システムの構築に本来必要なコンピュータの台数や管理および維持コストの削減が可能となり、有効なシステムの運用が実現される。しかし、仮想化にはこれらメリットばかりではなく、デメリットも存在する。

デメリットとしてまず挙げられるのは、仮想化にはオーバーヘッドが存在することである。仮想化を行わないで構築したシステムと仮想化を行い構築したシステムでは、一般的に後者のスループットが悪化する。それ以外のデメリットとして管理が複雑になることが挙げられる。仮想化を用いる場合、新たに Xen¹⁾ や VMWare²⁾ などのハイパーバイザなどの知識が管理者に要求される。

本報告では、仮想化のオーバーヘッドに焦点を当て、測定した結果について述べる。なお、仮想化対象のシステムとしては、SPECjAppServer2004³⁾ を採用した。採用理由としては、以下の理由が挙げられる。

多層構成のサーバーを統合するとオーバーヘッドが発生する。J2EE は 3 層以上の多階層アーキテクチャを想定しており、J2EE テクノロジーで実装されている SPECjAppServer2004 は本報告で適切な仮想化対象のシステムとなると判断した。

本報告では特に仮想マシン間通信のオーバーヘッドを探るため、3 層サーバーのうちの 2 つを同一仮想マシンに配置した構成と全てのサーバーを独立した仮想マシンに配置した構成とを比較する。

また、SPECjAppServer2004 の各トランザクションに必要な CPU 時間を求め、これらを足し合わせる事で、システム全体の負荷を予測した。この予測値と実測値の比較も合わせて報告する。

2. SPECjAppServer2004

2.1 サーバーの観点での SPECjAppServer2004 構成

SPECjAppServer2004 は、Web サーバー (Web)、Application サーバー (App)、DB サーバー (DB) の 3 つのサーバー群で構成されている (図 1)。矢印は各ドメインの呼び出し関係を表している。図 1 の Driver は SPECjAppServer2004 に含まれるトランザクション生成プログラムである。Driver の生成するトランザクションは主に顧客に車を販売する HTTP

^{†1} 会津大学
Aizu University

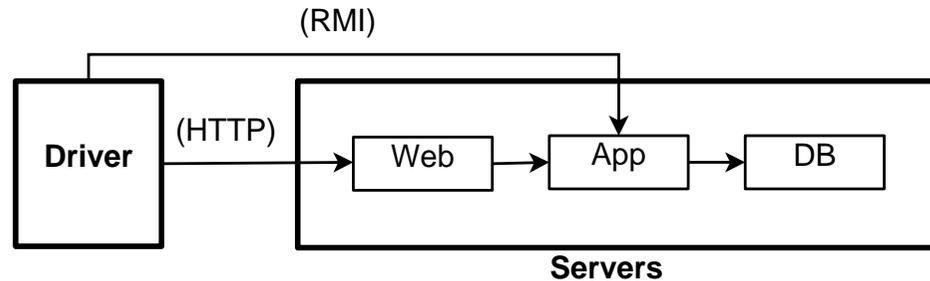


図 1 SPECjAppServer2004 システム構成
Fig.1 SPECjAppServer2004 System Configuration

によるものと自動車の製造を発注する Remote Method Invocation (RMI) によるものに分けられ、それぞれ Web, App サーバーに対して発行される。

2.2 アプリケーションの観点での SPECjAppServer2004 構成

SPECjAppServer2004 の想定するシステムは以下の 5 つのアプリケーションドメインに分けられて実装されている (図 2)。

- **Dealer** : 在庫追跡や自動車販売などを行う Web ベースのインターフェースを提供する。
- **Customer** : 注文の発注・確認・キャンセルなどを行う。
- **Mfg** : 自動車を製造する。WorkOrder, LargeOrder という 2 種類の製造ラインを持つ。
- **Supplier** : Mfg に対し、部品を提供する。
- **Corporate** : 顧客情報や在庫情報を管理する。

前述のように、システムに対するアクセスは Dealer か Mfg を通して行われる。SPECjAppServer2004 はこれら 2 つのドメインにおけるスループットの総和を性能の指標として定義している*1。

2.2.1 Dealer・Mfg ドメインのトランザクション

Dealer ドメインには以下の 3 つのトランザクションがある。

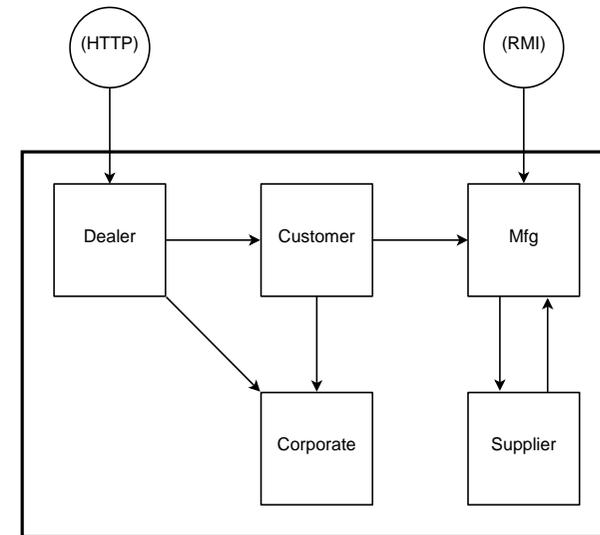


図 2 アプリケーションドメイン アーキテクチャー
Fig.2 Application Domain Architecture

- **Browse** : 購入可能な自動車の一覧を閲覧する。
- **Purchase** : 自動車を購入する。
- **Manage** : 在庫を管理する。

Mfg ドメインには以下の 2 つのトランザクションがある。

- **WorkOrder** : 少量の製品を計画的に製造する。
- **LargeOrder** : 大量注文時に大量の製品を製造する。

上記トランザクションはシステムの規模を表すスケーリングファクターに対して一定の割合で Driver から生成される。また Dealer トランザクション間の比率は Purchase (25%)・Manage (25%)・Browse (50%) となっている (表 1)。

3. 実験

3.1 実験環境

3.1.1 システム構成

Xen Hypervisor を用いて、表 2 の物理マシンに複数の仮想マシンを作成し、Web サー

*1 本研究で報告する結果は 4) の Section 3.7 で定義される "Research and Academic Usage" に該当し、SPEC による公式な記録と比較してはならない

表 1 スケーリングファクター (SF) に対するトランザクションレート

Table 1 Transaction rate per SF

Dealer	Browse	$0.500 * SF/second$
	Purchase	$0.250 * SF/second$
	Manage	$0.250 * SF/second$
Mfg	WorkOrder	$0.225 * SF/second$
	LargeOrder	$0.025 * SF/second$

表 2 サーバー環境

Table 2 Experimental System Specifications

CPU	Xeon(dual core) 1.86GHz
Memory	4GB
HDD	SATA 80GB × 3
OS	CentOS 5
Hypervisor	Xen 3.1.2
Web	Apache 2.2.14
Application	JBoss 4.0.0
DB	PostgreSQL 8.3.5
Java	1.4.2.19

表 3 仮想マシン構成 (3-VM)

Table 3 Virtual Machine Configurations(3-VM)

	VCPU	Mem
Web	1	512 MB
App	2	2048 MB
DB	1	1024 MB
Dom0	1	512 MB

バー・App サーバー・DB サーバーを構築した(表 3, 表 4).

なお, 上記の環境にはカーネルパラメータのチューニングや各種ミドルウェアのチューニングを行ってある⁵⁾⁶⁾⁷⁾.

3.1.2 実験方法

本研究では前述の 3 つのサーバーを仮想化により 2 つの構成で実験を行った. 2-VM 構成においては, Web と App が同一の仮想マシンに配置され, DB は独自の仮想マシンに配置される. 3-VM 構成では全て個別の仮想マシンに配置される.

最初にシステムのサイズ(スケーリングファクター, SF)を増加させ, スループットの変化および各サーバーにおける CPU 使用率の変化を観測する.

表 4 仮想マシン構成 (2-VM)

Table 4 Virtual Machine Configurations(2-VM)

	VCPU	Mem
Web+App	2	2048 MB
DB	1	1024 MB
Dom0	1	512 MB

表 5 Dealer ドメインのトランザクションの割合 (%)

Table 5 Transaction Ratios in Dealer Domain (%)

mix	Browse	Purchase	Manage
mix0	50	25	25
mix1	60	20	20
mix2	40	30	30

次に各トランザクションを個別に実行しその際のサーバー毎の CPU 使用率を測定する. その値に基づきシステム全体の負荷を予測したものと実測値の比較を行う. なお個別のトランザクションの実行には 8) のカスタマイズされたドライバーを用いた.

上記測定においては, Dealer ドメインのトランザクションの割合を表 5 のように変化させ, 各測定値を比較した. mix0 は SPECjAppServer2004 のオリジナルであり mix1, mix2 は各々 Browse トランザクションの割合を増減させたものである.

表 6 と表 7 は, 各トランザクションあたりの CPU 時間を各仮想マシンごとにまとめたものである. これは, Xen の Domain0 において xentop で CPU 使用率を測定したものを, 各仮想マシン毎に集計し直して算出した.

Dom0 は, Domain0 の負荷を指している. なお, 表 6 の Web 列の WorkOrder と LargeOrder にデータが存在しないのは, それら Order は RMI アクセスのトランザクションであり, Web サーバーが介在しない為である.

3.2 実験結果

3.2.1 スループットの比較

3-VM (2-VM) における各 mix のスループットを図 3, 図 5, 図 7(図 4, 図 6, 図 8) に示す. また, Total とは Dealer と Mfg ドメインのスループットの合計値である.

いずれのグラフにおいても Dealer ドメインのスループットは SF=11 をピークとし, SF=14 では下っている事がわかる.

同じ mix における Dealer トランザクションのスループットを比較すると 2-VM では

表 6 各トランザクションの CPU 時間 3-VM (ms)
Table 6 CPU Time per Transaction in 3-VM (ms)

	Web	App	DB	Dom0	Total
Browse	5.11	71.28	4.36	6.62	87.38
Purchase	1.50	38.51	5.93	3.48	49.42
Manage	1.43	42.03	7.61	4.12	55.20
WorkOrder	-	22.79	12.85	3.91	39.55
LargeOrder	-	46.09	8.57	4.45	59.11

表 7 各トランザクションの CPU 時間 2-VM (ms)
Table 7 CPU Time per Transaction in 2-VM (ms)

	Web+App	DB	Dom0	Total
Browse	70.95	4.24	3.71	78.89
Purchase	37.71	5.69	2.78	46.19
Manage	41.25	7.21	3.34	51.80
WorkOrder	22.56	12.50	3.80	38.86
LargeOrder	44.91	8.19	3.80	56.90

SF=11 までほぼ線形に増加しているが、3-VM においては増加の割合が落ちている。これは、仮想化の層が多い 3-VM のオーバーヘッドの差と考えられる。一方 Mfg トランザクションのスループットは 2-VM と 3-VM 間に有為な差が見られず、いずれの構成においても SF=14 までほぼ線形に増加している。原因としては Mfg トランザクションは RMI により Web サーバーを経ず直接 App サーバーにアクセスする事、Dealer トランザクションより比較的执行レートが低い事、WorkOrder はレスポンス時間に影響されず一定時間毎に発行される事などが考えられる (表 1)。

3.2.2 負荷の比較

3-VM (2-VM) における CPU 使用率のグラフを図 9, 図 11, 図 13 (図 10, 図 12, 図 14) に示す。各 SF において左側のグラフが予測値であり、右側が実測値である。

予測値の計算には、表 1 から各 SF において 1 秒間に生成されるトランザクション数を求め、表 6, 表 7 の各トランザクションあたりの CPU 時間を乗じ算出した。実際のベンチマークにおいてはサーバーの負荷が上がればスループットが下がるため CPU 使用率は 100%を越えないが、予測値の計算では単に SF に基づき積算している。

SF=14 においてはシステムは飽和し CPU 使用率の実測値は 100%に近付く。この影響は主に App サーバーに現れ、その CPU 使用率は SF=11 のそれよりも下っている。一方 DB サーバーの使用率は SF = 11 → 14 においても若干の増加が見られる。

予測値は前述のようトランザクション毎の使用率を積算したもののなので SF=14 においては 100%を越えている。しかし、実測値において見られた App サーバーと DB サーバー間の比率の変化は見られない。システムが飽和する以前の $SF \leq 11$ においては全般的に予測値は実測値より小さく、VM 間では特に DB サーバーの使用率の差が顕著である。

仮想マシン構成間の比較では 2-VM においてより実測値と予測値の差が大きい。原因としては予測値においては VM 毎に CPU 使用率を求めたが、2-VM 構成においては Web と App の 2 つのサーバーを一括りで扱っているため精度が劣った、という可能性が考えられる。

4. おわりに

J2EE サーバーのベンチマークである SPECjAppServer2004 をモデルとして仮想化によるサーバーの集約のオーバーヘッドの測定を行った。

各トランザクションがサーバー毎に消費する CPU 時間を測定し、スケーリングファクター (SF) に対するシステム全体への負荷 (CPU 使用率) を予測した。実測値との比較では、スループットと (Dom0 を除いた) 負荷の変化傾向には近似性が見られたが、負荷の絶対値としては SF=11 において 3-VM 約 10%, 2-VM 約 30%の違いが見られた。考えられる原因としては、

- (1) 複数トランザクションが実行される場合相互干渉があり、単一トランザクション毎の測定に基づく予測との誤差
- (2) 負荷予測の場合各トランザクションの負荷を加算しているが、実際の測定の場合はクローズドループであり、大きな SF ではスループットが下がり結果 CPU 負荷も下がる

といったものがあげられる。

現在、8) のキューイングペトリネットモデルに基づいたシミュレーターを開発中であり、前述の原因 (2) の解決が期待できる。また、計測前の予測では VM 構成の差は主に Dom0 に現れると考えていたが、実際は各サーバーの負荷に差が見られた。これについてまだ考察の余地があると考えられる。それ以外にも、対象システムを昨年リリースされた SPECjAppServer2004 の後継ベンチマークである SPECjEnterprise2010⁹⁾ に移行することも計画している。

謝辞 トランザクション毎の CPU 時間測定のためのカスタムドライバーを提供して頂いた Dr. Samuel Kounev に感謝します。

参 考 文 献

- 1) Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T.L., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A.: Xen and the art of virtualization, *SOSP*, pp.164-177 (2003).
- 2) VMware Inc.: VMware.com, <http://www.vmware.com/>.
- 3) The Standard Performance Evaluation Corporation: SPECjAppServer2004, <http://www.spec.org/jAppServer2004/>.
- 4) The Standard Performance Evaluation Corporation: SPECjAppServer2004 Run and Reporting Rules, <http://www.spec.org/jAppServer2004/docs/RunRules.html>.
- 5) Red Hat, Inc.: パフォーマンスチューニングの落としどころ (2008).
- 6) 独立行政法人 情報処理推進機構: SPECjAppServer2004 による JBoss の性能・信頼性評価-評価手順- -カーネル 2.4 と 2.6 の比較- (2005).
- 7) 独立行政法人 情報処理推進機構: 2004 年度オープンソースソフトウェア活用基盤整備事業 「OSS 性能・信頼性評価 / 障害解析ツール開発」 Java アプリケーション層の評価 (2004).
- 8) Kounev, S.: J2EE Performance and Scalability - From Measuring to Predicting, *SPEC BENCHMARK WORKSHOP 2006, AUSTIN, TEXAS* (2006).
- 9) The Standard Performance Evaluation Corporation: SPECjEnterprise2010, <http://www.spec.org/jEnterprise2010/>.

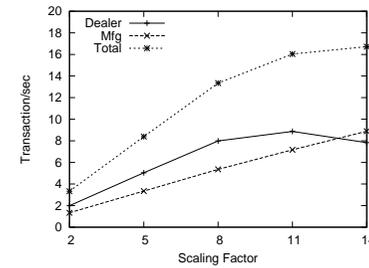


図 3 スループット (mix0, 3-VM)
Fig. 3 Throughput (mix0, 3-VM)

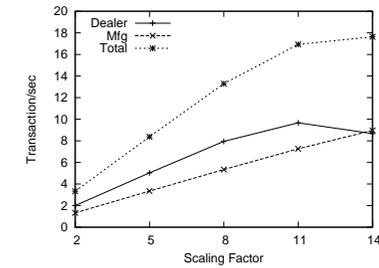


図 4 スループット (mix0, 2-VM)
Fig. 4 Throughput (mix0, 2-VM)

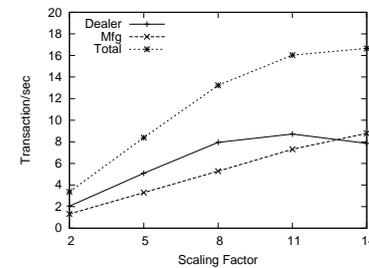


図 5 スループット (mix1, 3-VM)
Fig. 5 Throughput (mix1, 3-VM)

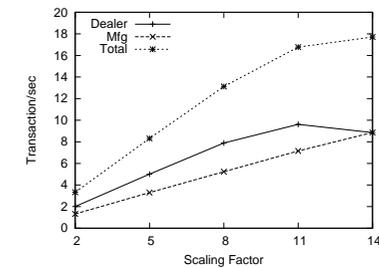


図 6 スループット (mix1, 2-VM)
Fig. 6 Throughput (mix1, 2-VM)

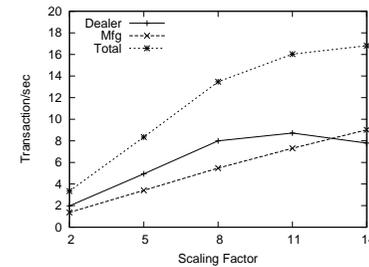


図 7 スループット (mix2, 3-VM)
Fig. 7 Throughput (mix2, 3-VM)

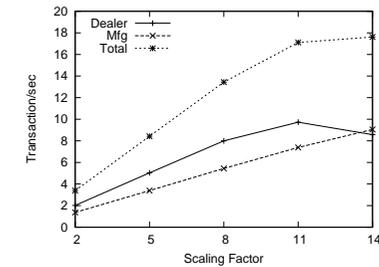


図 8 スループット (mix2, 2-VM)
Fig. 8 Throughput (mix2, 2-VM)

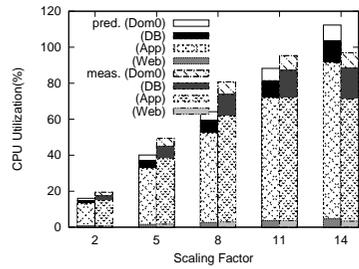


図 9 CPU 使用率 (mix0, 3-VM)
 Fig. 9 CPU-Utilization (mix0, 3-VM)

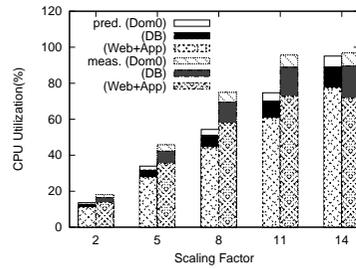


図 10 CPU 使用率 (mix0, 2-VM)
 Fig. 10 CPU-Utilization (mix0, 2-VM)

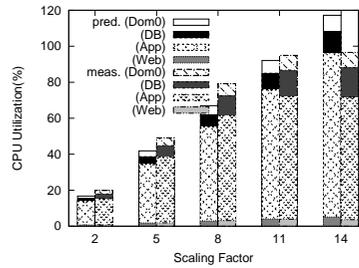


図 11 CPU 使用率 (mix1, 3-VM)
 Fig. 11 CPU-Utilization (mix1, 3-VM)

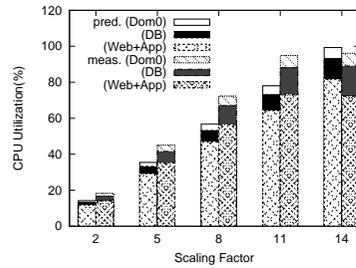


図 12 CPU 使用率 (mix1, 2-VM)
 Fig. 12 CPU-Utilization (mix1, 2-VM)

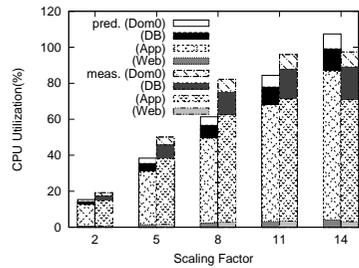


図 13 CPU 使用率 (mix2, 3-VM)
 Fig. 13 CPU-Utilization (mix2, 3-VM)

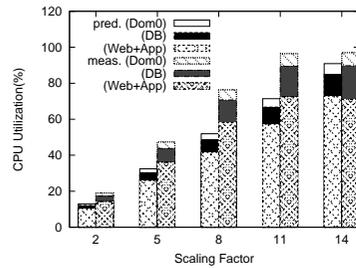


図 14 CPU 使用率 (mix2, 2-VM)
 Fig. 14 CPU-Utilization (mix2, 2-VM)