

# 仮想環境を利用したマルチテナントシステムの性能推定の一考察

市原 利浩<sup>†</sup> 魚住 光成<sup>†</sup> 北山 泰英<sup>†</sup> 小笠原 大治<sup>†</sup>

三菱電機 (株) <sup>†</sup>, 三菱電機インフォメーションシステムズ (株) <sup>‡</sup>

## 1. はじめに

近年注目を浴びているクラウドコンピューティングを活用する技術の一形態として、マルチテナント[1]がある。マルチテナントとは、一つのハードウェア資源を使って、複数の利用者にあたかもシステムを占有しているかのようにサービスを提供する形態のことである。このマルチテナントのテナント数を増やすと、例えばハードウェアコストを上昇させずに契約数を上昇させることが可能となるが、時間性能の劣化も発生する。そのため、テナント数の増加に伴う時間性能の変化を予め把握しておくことが、マルチテナントでサービスを提供する上で必要である。

本稿では、仮想環境を利用したマルチテナントシステムにおいて各テナントの同等のアプリケーションが繰り返し動作する場合の性能推定方法について、我々のアプローチを報告する。

## 2. 我々のアプローチ

### 2.1. マルチテナント処理のモデル

対象とするマルチテナントのアプリケーションは、CPU 処理とディスクアクセス処理を繰り返し行うものとし、1テナントの1回の処理モデルを図1のように考える。図1が示すCPU処理時間  $Tq\_cpu$  とディスクアクセス時間  $Tq\_io$  には、CPU コアの割り付け待ちや他のテナントのディスクアクセス処理優先のための待ち時間が含まれると考え、それぞれの実処理時間を  $Ts\_cpu$ 、 $Ts\_io$  としたモデルである。

この時、CPU 処理とディスクアクセス処理の待ちを含む処理時間  $Tq$  と実処理時間  $Ts$ 、1秒あたりの処理件数  $\lambda$  の関係は、待ち行列モデルを利用して  $Ts(\lambda)$  の関数で表すことができることから、マルチテナントの時間性能は、アプリケーション単体の性能  $Tq(=Ts)$  と、使用するサーバの  $Ts$  の特性  $Ts(\lambda)$  から導出できると考えられる。

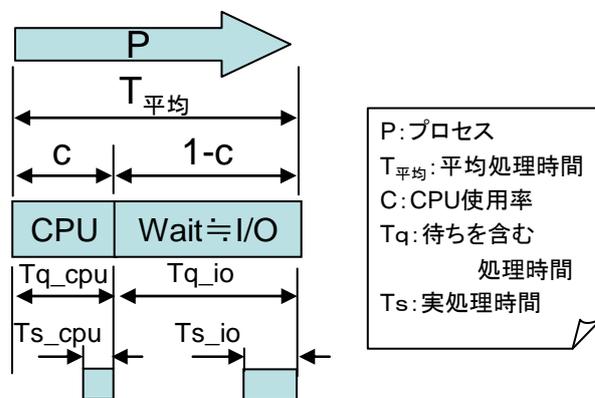


図 1: 1 処理のモデル

### 2.2. マルチテナント性能推定モデル

我々が提案するマルチテナント性能推定モデルは、2.1節の処理モデルに基づき、あるアプリケーション実行時の実処理時間を算出する関数を  $Ts(\lambda)$ 、同等ハードウェア環境における別アプリケーション実行時の実処理時間の関数を  $Ts'(\lambda)$  とした場合、これらは、下記 (2.1) の関係になると考え、計測により導出した  $\alpha$  と  $Ts'(\lambda)$  の近似式によって  $Ts(\lambda)$  を定義するものである。

$$Ts(\lambda) = \alpha \times Ts'(\lambda) \quad (2.1)$$

図2に示すマルチテナント性能推定モデルは、汎用的なプログラムによって、多重度が1~nまでの計測を行い、 $Ts'(\lambda)$  を導出する。また、同等ハードウェア環境で多重度が1の場合において、使用するアプリケーションの  $Ts$  を導出し、多重度が1の場合の  $Ts(\lambda)$  と  $Ts'(\lambda)$  の値、及び (2.1) の関係式から係数  $\alpha$  を算出する。

このモデルに従い、CPU 処理とディスクアクセス処理における係数  $\alpha\_cpu$  と  $\alpha\_io$ 、実処理時間  $Ts(\lambda)\_cpu$  と  $Ts(\lambda)\_io$  をそれぞれ導出し、2.3節の待ち行列モデルの式に適用して、 $Tq\_cpu(\lambda)$  と  $Tq\_io(\lambda)$  を算出する。処理時間  $T$  は、図1の処理モデルより、下記 (2.2) の式となることから、n 多重時に処理時間が算出できる。

$$T = Tq\_cpu(\lambda) + Tq\_io(\lambda) \quad (2.2)$$

### 2.3. 待ち行列モデル

アプリケーションのターンアラウンドが、CPU (および CPU の割り付け待ち) と、ディスクアクセス (および I/O Wait) からなるとき、それぞれの処理時間は、実処理を行なうサービス時間  $Ts$  と、待ち行列の待ち時間  $Tw$  から構成

Performance estimation of a MultiTenant System in Virtual Environment

<sup>†</sup>Toshihiro Ichihara, Uozumi Mitsunari,

Mitsubishi Electric Corporation

<sup>‡</sup>Yasuhide Kitayama, Daiji Ogasawara

Mitsubishi Electric Information Systems Corporation.

されると考えられる。

CPU 処理は、複数のコアが存在する場合、コアの割り付け要求に対して複数の窓口が存在する待ち行列の M/M/m のモデルで表すことができると仮定する。ディスクアクセス処理は、アプリケーションが書き込みもしくは読み込みの要求を発行する段階で、対象はユニークに特定されるため、待ち行列の M/M/1 のモデルで表すことができると仮定する。これより、窓口 m において、Ts と Tq、λ の関係式を、窓口数ごとに表すことが出来る。なお、本稿における窓口とは、待ち行列モデルにおける逐次処理をする場所のことである。

### 3. 性能推定の手順

#### 3.1. サーバ特性の計測

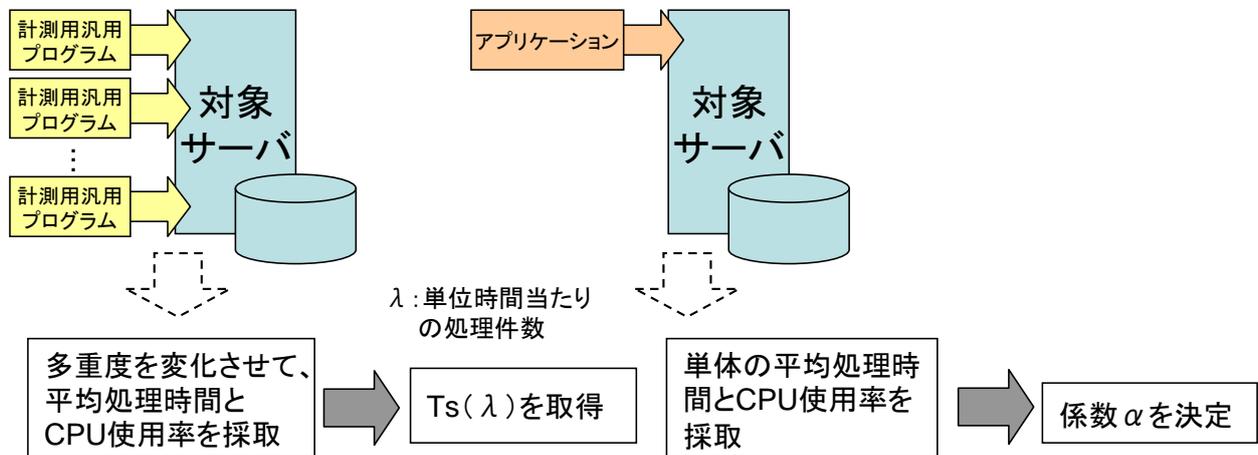
対象となるサーバ上で、計測用の汎用プログラムを用意して性能特性の計測を行なう。このプログラムは、CPU 処理とディスクへの書き込みを交互に繰り返し行うもので、テナント数 n を 1 から目標とする数まで変え、各汎用プログラムを並行して動作させる。汎用プログラムは、隙間無く連続して実行し、1 回の平均処理時間（ターンアラウンド時間）T、CPU 使用率 C を計測する。これにより、到着率 λ、待ちを含む CPU とディスクアクセスの処理時間 Tq\_cpu、Tq\_io は、それぞれ次の式によって求められる。

$$\lambda = n/T \quad (2.3)$$

$$Tq\_cpu = T \times C \quad (2.4)$$

$$Tq\_io = T \times (1 - C) \quad (2.5)$$

上記 (2.3) ~ (2.5) の式と待ち行列モデルによって、テナント数毎の CPU とディスクアクセスの実処理時間 Ts\_cpu、Ts\_io を求める。CPU についてはコア数が窓口数となり、ディスクアクセスの窓口数は 1 となる。次に、Ts\_cpu と λ、Ts\_io と λ の関係から多次元の関数 Ts\_cpu(λ)、Ts\_io(λ) をそれぞれ導出する。



#### 3.2. アプリケーション性能の計測

使用するアプリケーションを 1 テナント（単体）で隙間なく連続動作させ、3.1 節同様に、1 回の平均処理時間（ターンアラウンド時間）T と CPU 使用率 C の計測値から、λ、Ts\_cpu、Ts\_io を求める。次に、この λ、Ts\_cpu、Ts\_io と 3.1 節で導出した Ts\_cpu(λ)、Ts\_io(λ)、及び 2.2 節の (2.1) に基づいた下記の関係式から、α\_cpu、α\_io をそれぞれ算出する。

$$Ts\_cpu = \alpha\_cpu \times Ts\_cpu(\lambda)$$

$$Ts\_io = \alpha\_io \times Ts\_io(\lambda)$$

#### 3.3. マルチテナント時間性能の算出

2.2 節の (2.2) の式より、アプリケーションの処理時間 T は、Tq\_cpu(λ) と Tq\_io(λ) の和である。Tq\_cpu(λ) と Tq\_io(λ) は、3.1 節で導出した Ts\_cpu(λ) と Ts\_io(λ)、3.2 節で求めた λ、係数 α\_cpu、α\_io から算出する。

### 4. まとめ

CPU 処理とディスクアクセスの実処理時間 Ts\_cpu(λ) と Ts\_io(λ) は、ハードウェア構成で決まり、アプリケーションには依存しないため、同一ハードウェアを使用する場合、Ts\_cpu(λ) と Ts\_io(λ) を再利用することができると考えられる。そのため、使用するサーバの多重時の CPU 処理とディスクアクセスの実処理時間を汎用プログラムによって予め計測しておき、使用するアプリケーションの 1 テナント時の平均処理時間を計測することで、複数テナント時の平均処理時間が推定できると考える。今後は、本手法の評価に取り組む予定である。

### 参考文献

[1] 米持幸寿, クラウドを実現する技術, インプレスジャパン, 2009年9月