

# メモリの影響と CPU 処理性能を考慮した リソースサイジングに関する検討

若井 祐樹<sup>†</sup>      松浦 陽平<sup>†</sup>  
三菱電機 情報技術総合研究所<sup>†</sup>

## 1. 背景・目的

システム開発の上流工程において、計算機のリソースサイジングを行う際には待ち行列モデルを用いることが一般的である。従来のサイジング手法は、1 処理当りのリソース消費量を本番相当の環境から計測する。その計測値を性能モデルに反映することで本番環境の高負荷条件下の応答時間を見積もることが出来る。しかし近年では、サイジングが行われる際に、計測環境からスペックを変える等、リソースを計測した環境と見積もる対象の性能が異なるケースや、大量のデータを処理する等、メモリの容量を大量に使用することでメモリボトルネックとなり性能劣化が起きるケースも多く存在する。そのため、計測した環境が異なることによる処理性能の差と、メモリボトルネックによる性能劣化の有無を反映するサイジング手法が求められている。本稿では、そのサイジング手法を実現するための性能モデルを検討した結果を示す。

## 2. 従来手法

従来のサイジング手法の1つとして、小杉らの性能モデル [1]がある。この性能モデルでは、評価対象のサーバの CPU、I/O、アプリケーションを計測し、その値をそれぞれ M/M/n 型待ち行列モデルとして反映し、それらを直列に接続することでサーバ全体の性能モデルを構築している。

## 3. 課題

文献 [1]の性能モデルには、2つの課題が存在する。1つは、計測環境と見積もる環境の CPU や I/O などは等しいという前提であるため、見積もる環境のスペックを変更する等、計測環境と異なった場合に性能差によるズレが生じ、正確なサイジングが出来ない点である。もう1つは、サーバのメモリリソースを考慮していないため、大量のデータ処理を行った際にメモリ容量の不足によるメモリボトルネックから処理性能の劣化が起きる場合に正確なサイジングが出来ない点である。そのため、より正確なシステムのサイジングを行うには、環境を変更した場合の反映手段とメ

モリリソースを含めた性能モデルが必要である。

## 4. 解決策

本稿では、3 章の課題を解決するため、メモリリソースを考慮する木下らの性能モデル [2]と、CPU の性能が異なる場合に一方の性能を相対値で計算する紀の手法 [3]を参考に、従来モデルを改良した性能モデルを提案する。その際、メモリリソースを考慮する 4.1 節のモデルが理論による性能の計算が困難であるため、シミュレーションにて応答時間を計測することでサイジングを行うことにした。4 章では、文献を基にそれぞれの課題に対応した性能モデルについて述べる。

### 4.1. メモリリソースを考慮するモデル

文献 [2]の性能モデルはメモリを表すメモリ部と CPU、I/O による内部処理部で構成している。メモリ部は窓口数の分だけジョブを内部処理部に送ることでメモリによるボトルネックを表している。このモデルを参考に、本稿による 1 台当りのサーバを見積もる性能モデルを図 1 に示す。

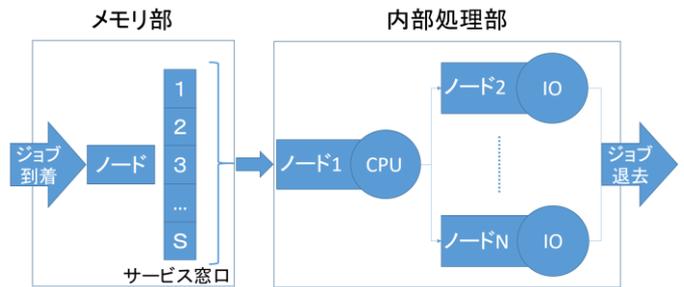


図 1：1 台当りのサーバ内の性能モデル

提案モデルはメモリ部と内部処理部で構成している。メモリ部では、1 プロセスにかかるメモリ消費量を基にサービス窓口数を設定する。メモリ部に到着したジョブはサービス窓口を通して内部処理部へ向かう。この時、内部処理部の滞在ジョブ数がメモリ部の窓口数よりも多い場合は、メモリ部によるボトルネックが発生したと判断する。ボトルネック発生時には、内部処理部に空きが出るまで待機するため、メモリ部のサービス率は内部処理部の処理時間に等しくなる。この時の変化を数式 1 に示す。

### 数式 1 提案モデルにおけるサービス率変化

$$\text{メモリ部のサービス率 } \mu_n = \begin{cases} \mu_{init}, & n < S \\ W, & n \geq S \end{cases}$$

Investigation of the sizing in consideration of memory influence and CPU processing performance

<sup>†</sup> Information Technology R&D Center, Mitsubishi Electric Corporation.

#### 4.2. CPU 性能を考慮する性能モデル

文献 [3]から、性能が異なる CPU がある場合、一方の CPU を 1 としたときの相対値から性能を計算するシングルプロファイル法を参考にした。

本稿では、計測環境と本番環境の性能比から推定平均処理時間を計算することでサービス率を求める。計測環境と本番環境との性能比はベンチマークテスト等のカタログスペックから算出する。計算した性能比を基に本番環境の推定平均処理時間を数式 2 にて算出する。

#### 数式 2 環境変化時の性能比を利用した計算式

$$\begin{aligned}
 & \text{本番環境の処理時間 } W_r \\
 &= \text{計測環境の処理時間 } W_p \\
 & \times \frac{1}{\text{計測環境との性能比}}
 \end{aligned}$$

サービス率は平均処理時間の逆数であることから、本番環境の平均サービス率は  $1/W_r$  となる。よって、推定されたサービス率  $1/W_r$  を用いて、本番環境のサイジングが可能となる。

### 5. 評価

本稿では CPU の性能差とメモリに対する考慮がどの程度サイジングに影響を与えるか検討するために、従来の性能モデルと提案する 2 つの性能モデルを比較する。今回は Python の離散型シミュレーションライブラリである SimPy を用いて作成したシミュレーションにて評価を行った。シミュレーションでは、到着率毎の平均滞在時間を計算する。その結果から従来モデルと提案モデルの比較を行う。この時、ボトルネックが発生するようなパラメータを設定する。

#### 5.1. メモリを考慮したシミュレーション

4.1 節のモデルは、到着率が上がる毎にボトルネックの影響から平均応答時間が大きくなると考えられる。これを判断するために、4.1 節で述べた提案モデルと従来モデルをシミュレーションで実行した。その結果である各到着時間のシステム内滞在時間を比較したものを図 2 に示す。

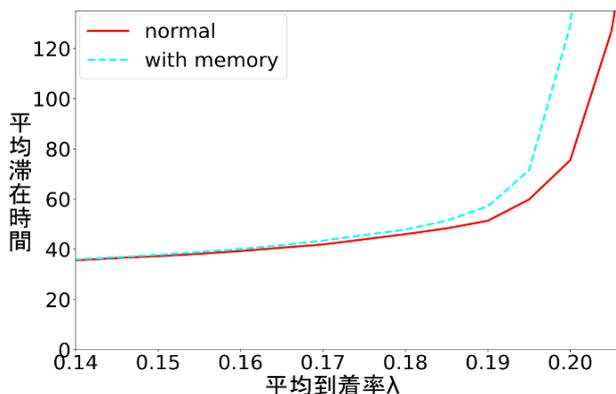


図 2 : メモリを考慮したモデルと従来のモデルによる処理時間比較 ( $\mu = 10, S = 7$ )

この時のメモリ部に当るノードのパラメータは共にサービス率が 10、窓口数が 7 とした。破線

がメモリリソースを考慮した性能モデルであり、実線の従来モデルと比較して、ボトルネックが発生するような到着率になった際に平均滞在時間が大きくなった。

#### 5.2. CPU の相対値によるシミュレーション

CPU 性能を考慮するモデルでは、性能の差が反映されるため、平均応答時間が相対値に比例して変化すると考えられる。これを判断するために、環境の変化として図 1 のノード 1 に当るノードのサービス率を 2 倍となるように設定しシミュレーションを行った。この際の通常時と 2 倍に変更した時の各到着時間における処理時間比較を図 3 に示す。従来モデルと提案モデルの滞在時間が発散する到着率はおおよそ 2 倍となり、相対値に比例していることが示された。

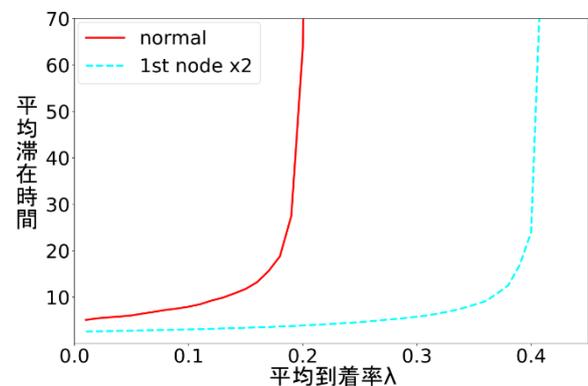


図 3 : ノードのサービス率を 2 倍とした際の従来モデルとの処理時間比較

### 6. まとめ

本稿では、処理性能の差とメモリボトルネックを反映するサイジング手法のための性能モデルについて検討した。5.1 節の結果から、メモリリソースを考慮することによって、メモリボトルネックが発生するような処理量が多い時にシステム滞在時間が大きくなることを示した。このことからボトルネックが発生した場合、提案モデルではより正確に推定できると考えられる。また、5.2 節では CPU 性能を相対値で反映することで相対値に比例して処理性能に影響が出ること示した。このことから、相対値によって異なる性能を持つ場合、提案モデルはより正確に推定できる可能性がある。今後は検討した性能モデルを実機にて反映し、サイジング精度を検証する。

### 7. 参考文献

1. 小杉優, ほか. 大規模 IoT システムにおける計算機リソースサイジングの研究. : FIT2015(第 14 回情報科学技術フォーラム), 2015.
2. 木下俊之, 高秀梅. メモリ資源のある計算機システムの待ち行列網による近似性能評価法の提案と評価. : 情報処理学会, 2010.
3. 紀一誠. 待ち行列ネットワーク. : 朝倉書店, 2012.