

キャッシュの影響を考慮したリソースサイジングに関する検討

日野 泰子[†] 松浦 陽平[†][†]三菱電機株式会社 情報技術総合研究所

1. はじめに

システム開発の上流工程において行われるサイジングは、性能目標を達成するために必要なリソース量を見積もる作業である。負荷試験とは異なり得られる情報が限られているため、性能指標値が従う性能モデルを仮定し、流用元やPoCの計測結果をもとに性能指標値を見積もる。一般的な性能モデルとして、待ち行列理論に基づくモデルが挙げられる。このモデルでは、解析の容易性からサービス時間が指数分布に従うと仮定することが多い(例えば, [1])。一方で、キャッシュ等の影響により、リソースのレイテンシは多峰分布になりやすいことが知られており[2]、現実とモデルの間にずれが生じている。

本稿は、サービス時間が2峰型の分布に従うとみなすことによって、キャッシュの影響を反映した性能モデルを提案するものである。

2. 先行研究

参考文献[1]で提案している性能モデルは、評価対象サーバのCPU, I/O, アプリをそれぞれM/M型の待ち行列モデルとし、それらを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。ここでM/M型とは、リクエストの到着過程をポアソン到着、サービス時間分布を指数分布と仮定した待ち行列モデルである。サービス時間分布を固定しているため、必要な計測値が各ノードの平均サービス時間(評価対象の処理1件当たり消費されるCPU時間・I/O時間)のみと少ないことが特長である一方で、サービス時間分布が指数分布から外れる対象には適さない。

3. 目的

先行研究の手法は、適用は容易なもの、キャッシュの影響を考慮することができない。そこで、キャッシュの影響を考慮した性能モデルを提案することを本稿の目的とした。ただし、性能モデルに含まれるパラメータは、計測によって特定可能なもののみとする。

4. 提案する性能モデル

4.1 全体像

本稿では、サイジングの対象をCPUに限定するものとする。性能モデルには待ち行列モデルのM/G/Nを採用し、サービス時間分布はキャッシュの影響をモデル化した分布とする。

M/G/Nは解析解が得られないため、性能指標の計算には近似式を用いる。近似式には、サービス時間分布の特性が変動係数 c_s として残るため、4.2においてモデル化したサービス時間分布の変動係数を定式化する。また、サービス時間分布に含まれるパラメータの特定方法を4.3に示す。

4.2 サービス時間分布のモデル化

キャッシュにヒットした場合、ヒットしなかった場合、それぞれのサービス時間が従う単峰型の分布(以下、分布 H_{hit} 、分布 H_{miss})を用意し、キャッシュミス率を R で重み付けし混合した分布をサービス時間の分布(以下、分布 H)とする(図1)。本来、キャッシュは数段階に分けられるが、ここではまとめて扱う。

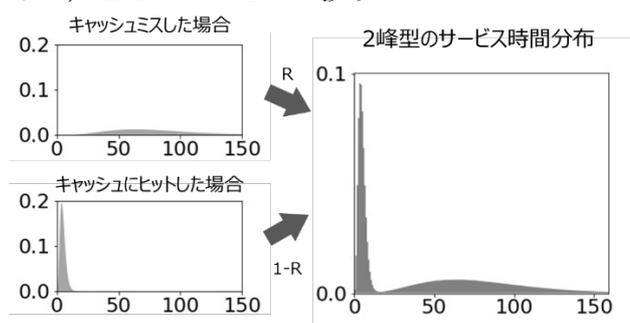


図1 サービス時間分布のモデル化の考え方

分布 H_{hit} 、 H_{miss} は、ガンマ分布 $\Gamma(\alpha, \beta)$ で定式化し、 $\alpha = 5$ は共通、 β は $E[H_{miss}] = X E[H_{hit}]$ となるよう定める。

上記のようなモデル化を行うと、分布 H の平均は次式で求められる。

$$\begin{aligned} E[H] &= (1-R)E[H_{hit}] + R E[H_{miss}] \\ &= (1-R) \frac{\alpha}{\beta} + R \frac{X\alpha}{\beta} = \frac{5(1-R + RX)}{\beta} \end{aligned} \quad \dots(\text{式 } 1)$$

また、分散は次式で求められる。

$$\begin{aligned} V[H] &= E[H^2] - \{E[H]\}^2 \\ &= (1-R)E[H_{hit}^2] + RE[H_{miss}^2] - \{E[H]\}^2 \end{aligned} \quad \dots(式2)$$

分布 $H_{hit} \cdot H_{miss}$ はガンマ分布であることから、

$$E[H_{hit}^2] = \frac{20}{\beta^2}, E[H_{miss}^2] = \frac{20X^2}{\beta^2} \quad \dots(式3)$$

となる。 (式1)および(式3)より、 $V[H]$ は β, X, R の関数と分かる。さらに、分布 H の変動係数 c_H も、

$$c_H = \frac{E[H]}{\sqrt{V[H]}} \quad \dots(式4)$$

より、 β, X, R の関数として求められる。よって、分布 H の変動係数 c_H を特定するために必要なパラメータは、 β, X, R となる。

4.3. パラメータの計測方法

4.3.1 キャッシュミス率R

CPUに実装されているパフォーマンスカウンタからキャッシュミス率を取得することが可能である。Linux系のマシンであれば、perfコマンドを用いて、値を取得できる。ただし、仮想環境では、perfコマンドで取得できる値に制限がありキャッシュミス率を取得できないことがある。

まず、以下のコマンドで、取得できる項目を出力し、キャッシュミス率に当たる項目の名称を確認する。

```
# perf list
```

確認した名称および計測対象の処理を実行するコマンドを指定して、キャッシュミス率を計測する。

```
# perf stat -e <cache-misses> <command>
```

なお、計測対象は、コマンドでなくプロセスIDで指定することも可能である。

4.3.2 キャッシュ性能係数X

処理は同等だが、キャッシュミス率の異なるサンプルプログラムを用意し、その計測結果からキャッシュ性能係数 X を算出する。

まず、サンプルプログラムとして、アクセスする要素の順番を変えた配列処理のプログラムを2種類用意する。それぞれを実行したときの平均キャッシュミス率(R_A, R_B)および平均処理時間(T_A, T_B)を計測する。キャッシュミスがない場合の処理時間を T とすると、

$$\begin{aligned} T_A &= (1 - R_A)T + R_A X T \\ T_B &= (1 - R_B)T + R_B X T \end{aligned}$$

となるため、計測した T_i および R_i を代入し連立方程式を解くことで、キャッシュ性能係数 X が算出できる。

4.3.3 サービス時間分布のパラメータ β

キャッシュミス率と同時に、十分に低い処理頻度における平均CPU使用率を計測し、処理1件で消費するCPU時間の平均 $E[T_s]$ を求めておく(従来手法と同様であるため、具体的な方法は省略する)。 T_s は平均サービス時間と見なせるため、(式1)より、

$$\begin{aligned} E[H] &= E[T_s] \\ \beta &= \frac{5(1-R+XR)}{E[T_s]} \end{aligned} \quad \dots(式5)$$

となり、 β を求めることができる。

5. 提案モデルを用いたサイジング方法

4章で求めた c_H をM/G/Nの近似式に適用することで、CPUの使用量を見積もることができる。

平均待ち時間の近似式としてKimura(1986a)の近似[3]を採用すると、平均CPU使用時間 $E[T_R]$ は、M/M/Nにおける同条件での平均待ち時間を $E[W(M)]$ 、同様にM/D/Nの平均待ち時間を $E[W(D)]$ として、以下のように求められる。

$$\begin{aligned} E[T_R] &= E[T_s] + E[T_w] \\ &= E[T_s] + \frac{1 + c_H^2}{\frac{2c_H^2}{E[W(M)]} + \frac{1 - c_H^2}{E[W(D)]}} \end{aligned} \quad \dots(式6)$$

上記 $E[T_R]$ は、処理要求の到着頻度の関数となる。従って、性能目標で定められた処理要求の到着頻度における $E[T_R]$ を(式6)によって算出し、許容されるCPU使用率と照らし合わせることで、必要な計算機台数を見積もることができる。

6. まとめと今後の展望

本稿では、キャッシュの影響を受けたサービス時間分布を2峰型の分布でモデル化した性能モデルを提案するとともに、性能モデルに含まれるパラメータの計測の方法を示した。

今後は、実機を用いてサイジング精度の評価を行うとともに、仮想環境での計測方法を検討する。

参考文献

- [1] Brendan Gregg, “詳解 システム・パフォーマンス”, オライリー・ジャパン(2017)
- [2] 小杉 優等, “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).
- [3] 木村 俊一, “確率光学シリーズ1 待ち行列の数理モデル”, 朝倉書店(2016)