

キャッシュサーバを用いる分散 Web システムにおけるスループットを用いたオートスケールアルゴリズムの開発と評価

松田正也¹最所圭三²香川大学^{1,2}

1 はじめに

クラウド環境を用いた Web サービスを運用するうえにおいて、運用コストを抑えつつ、応答性を確保することが常に求められている。我々は、クラウド環境において負荷量に応じてキャッシュサーバ数を増減させることで、応答性の確保と運用コストの低減を目指す分散 Web システムの開発を行っており、キャッシュサーバの稼働率を用いたオートスケールアルゴリズム (以下、稼働率アルゴリズム) を開発した [1]。ここでの稼働率とは、本研究で用いている Apache の最大同時処理数に対する現在の同時処理数の割合を指す。このアルゴリズムでは過負荷状態と判断した時点で応答時間が既に延びていることが多かった。これに対して、このときのスループットは秒間リクエスト数の増加に比例して増加していた。このことから、本研究ではスループットを用いたオートスケールアルゴリズムを開発することにした。本稿では、スループットを用いたオートスケールアルゴリズムを開発、および先行研究で開発した稼働率アルゴリズムとの比較実験の結果について述べる。

2 稼働率アルゴリズムの問題点

文献 [1] で提案した稼働率アルゴリズムでは、稼働しているサーバの合計稼働率がスケールアウトの閾値を超えると稼働台数を増やし、スケールインの閾値を下回ると稼働台数を減らす。アクセスの振り分けにはロードバランサを用い、スケールアウト時の追加のサーバは起動が完了するまで待ってから振り分けの対象とする。

図 1 に、稼働率アルゴリズムを用いたオートスケールの結果を示す。なお、実験環境として、Azure 西日本リージョン上に構築した拡張ロードバランサ 1 台、オリジンサーバ 1 台、仮想キャッシュサーバ 9 台、アクセスを行うクライアント 4 台を用いた。拡張ロードバランサは Standard D4(vcpu 数 8, メモリ 28GB)[2]、それ以外は DS1.v2 Standard(vcpu 数 1, メモリ 3.5GB) を用いた。アクセスするページとして DokuWiki のフロントページを用意し、スケールアウトの閾値を 0.5、スケールインの閾値を 0.1 に設定して、クライアントから負荷テストツール (Gatling) を用いて、秒間リクエスト数を 50 から 2,000 まで増加させながらアクセスした。

図 1 の 50 秒あたりで稼働率の上昇によりスケールアウトしたが、追加のサーバの起動までの応答時間が非常に長くなっていた。その後、リクエスト量が増加している

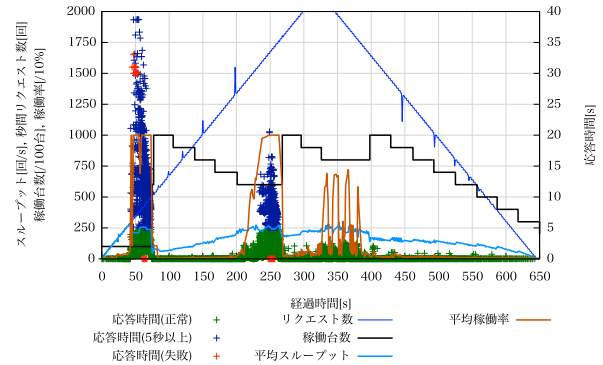


図 1 稼働率アルゴリズムの実験結果

にもかかわらず連続してスケールインしたことで稼働率が上がり、250 秒あたりで再度スケールアウトしている。これらは、稼働率ではサーバの負荷状況を十分に反映できないことを表している。これに対して、スループットは秒間リクエスト数にほぼ比例して上昇しており、平均スループットが 250 あたりで、スケールアウトが発生している。このことから、スループットを用いることにより早期に負荷上昇を検知できると考えられる。

3 スループットと応答時間の調査

スループットを負荷判定に用いることができるかどうかを確認するために実験を行った。調査対象のサーバは 2 節の仮想キャッシュサーバと同じものである。秒間リクエスト数を 100 から 400 まで 50 刻みで変更して実験を行った。図 2 に実験結果を示す。

図 2 より、秒間リクエスト数が 200 までの平均スループットは秒間リクエスト数と同数になっているが、250 から秒間リクエスト数を下回り始め、300 以上では約 270 で一定になった。このことから、秒間リクエスト数が 250 ではサーバが限界に達していると判断できる。応答時間に関しては、秒間リクエスト数が 200 までは平均値はそれほど増加は見られないが、標準偏差の増加が目立ち始めていた。以上のことから、スループットを負荷判定に用いることができ、スケールアウトすべきであると判断するスループット (今回の実験では 200) を本稿では上限スループットと呼ぶ。

4 スループットアルゴリズム

4.1 アルゴリズムの概要

3 節の結果に基づき、スループットを用いたオートスケールアルゴリズムを開発した。このアルゴリズムを本稿ではスループットアルゴリズムと呼ぶ。スループットアルゴリズムでは、処理能力推定機能と振分量調整機能、負荷判定機能からなる機構を用いる。各機能の役割を以下に示す。

Development and Evaluation of Autoscale Algorithm Using Throughput for Distributed Web System With Cache Server

¹Masaya Matsuda, Kagawa University

²Keizo Saisho, Kagawa University

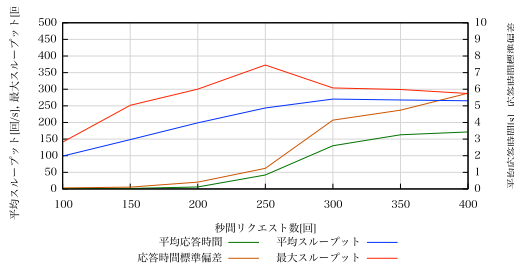


図2 スループット調査の結果

- A 処理能力推定機能: キャッシュサーバの処理能力(スループット)を推定する. キャッシュサーバに振り分けるリクエスト量を調整し, 応答時間が許容範囲内に収まる上限スループットを推定する.
- B 振分量調整機能: 異なる性能のキャッシュサーバを用いる場合に発生し得る負荷の偏りを調整する. 一部のキャッシュサーバが過負荷かつ, 他のキャッシュサーバが低負荷の場合, 過負荷キャッシュサーバの振り分け量を減らし, 低負荷キャッシュサーバの振り分け量を増やす.
- C 負荷判定機能: 上限スループット (TP_{High}) と, 一定区間 (i) のスループットの移動平均 (TP_{MAi}) に基づいて, キャッシュサーバ台数の増減の判定を行う. 稼働中のキャッシュサーバ全体を一つのクラスタとみなした移動平均を TP_{MAiAll} , 稼働中のキャッシュサーバと起動処理中のキャッシュサーバの上限スループットの合計値を $TP_{HighAll}$ とし, 負荷量 (LO) を以下の式で求める.

$$LO = \frac{TP_{MAiAll}}{TP_{HighAll}} \quad (1)$$

LO がスケールアウトの閾値を上回ると過負荷と判定し, スケールアウトが必要と判断する. LO がスケールインの閾値を下回ると低負荷と判定し, スケールインが必要と判断する.

4.2 比較実験環境

スループットアルゴリズムと稼働率アルゴリズムの比較評価実験を行った. なお, 処理能力推定機能と振分量調整機能は未実装のため, 上限スループットを3節で求めた200に固定して行った. 本実験では, 初期状態から秒間リクエスト数の上昇に合わせてスケールアウトさせ, 応答時間の増加・分散をどの程度抑えられるかという点で比較する. 負荷増加率に対するオートスケール性能を評価するために, 秒間リクエスト数が200増加する間隔が, 15秒のパターン(高)と30秒のパターン(中), 60秒のパターン(低)の3つのパターンで実験した. サーバは2節と同じ構成を用いる. スケールアウト性能のみを評価するために, 両アルゴリズムのスケールインの閾値は0.1に固定する. 事前実験の結果から, スループットアルゴリズムに用いる移動平均区間 i は3とし, スケールアウトの閾値 (Th_{High}) は稼働割合に応じて表1の値を用いる. なお稼働割合とは, 利用するキャッシュサーバ数の上限に対する稼働中と起動処理中のキャッシュサーバの台数の割合を指す. 利用するキャッシュサーバ数の上限は10台とした. 稼働率アルゴリズムの Th_{High} は事前実験の結果から, 表2の値を用いる.

表1 スループットアルゴリズムの Th_{High}

稼働割合 [%]	0 ~ 30	30 ~ 60	60 ~ 80	80 ~ 100
Th_{High}	0.3	0.5	0.7	0.9

表2 稼働率アルゴリズムの Th_{High}

稼働割合 [%]	0 ~ 30	30 ~ 50	50 ~ 70	70 ~ 90	90 ~ 100
Th_{High}	0.5	0.6	0.7	0.8	0.9

表3に各アルゴリズムの実験結果を, 図3に図1と同じ条件でのスループットアルゴリズムの実験結果を示す. 表3より稼働率アルゴリズムでも, 平均応答時間や標準偏差はいずれの場合でも1.5秒以下になっているが, 常に失敗リクエストが発生している. それに対して, スループットアルゴリズムでは, 常に平均応答時間と標準偏差が稼働率アルゴリズムよりも短く, 負荷上昇が中低では遙かに短く, 失敗リクエストは0になった. また, 図3よりスループットアルゴリズムではよりはやくスケールアウトしたことで応答時間のピークを下げる事が出来ていることが確認できる. 以上のことからスケールアウト性能において, 稼働率を用いたオートスケールアルゴリズムよりもスループットを用いたオートスケールアルゴリズムの方が優れているといえる.

表3 各アルゴリズムの実験結果

負荷上昇	アルゴリズム	平均 [s]	標準偏差	失敗 [%]
高	稼働率	0.960	1.288	2.22
高	スループット	0.111	0.540	0.00
中	稼働率	0.466	0.855	0.20
中	スループット	0.014	0.019	0.00
低	稼働率	1.024	1.506	2.37
低	スループット	0.014	0.024	0.00

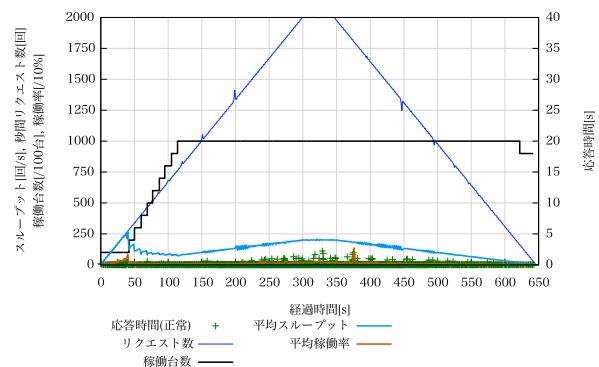


図3 スループットアルゴリズムの実験結果

5 おわりに

以上, スループットを用いたオートスケールアルゴリズムを開発し, 先行研究で開発した稼働率を用いたオートスケールアルゴリズムとの比較実験の結果, 優位性があることを確認した. 今後の課題として, 処理能力推定機能と振分量調整機能の開発がある.

参考文献

- [1] 松田正也, 最所圭三, "分散 Web システムにおけるスケールングアルゴリズムの改良と評価", 情報処理学会 第79回全国大会, 7T-03, p3-271, 2017.03
- [2] Cloud Services の価格, <https://azure.microsoft.com/ja-jp/pricing/details/cloud-services/>, 2018年12月28日閲覧