

分散 Web システムにおけるスケーリングアルゴリズムの改良と評価

松田正也¹

最所圭三²

香川大学^{1,2}

1 はじめに

我々は、Web サーバの負荷量に応じてクラウド環境上の仮想キャッシュサーバ (VC サーバ) 数を動的に増減させることで、応答性を確保しつつ運用コストを低減する分散 Web システムの実現を目指している。本研究では、大本の Web サーバ (オリジンサーバ) と VC サーバを監視し、負荷量に応じて VC サーバを起動・停止させ、それらのサーバにリクエストを振り分ける機能を持つ拡張ロードバランサを開発している [1]。

拡張ロードバランサは、稼働中のサーバの負荷量が上限を超えると VC サーバを起動し (スケールアウト)、下限を下回ると停止させる (スケールイン) が、先行研究では、スケールアウトのトリガーとなる閾値 (負荷量の上限) を固定して評価していた [2]。しかし、閾値により VC サーバの稼働台数や稼働率、クライアントへの応答時間、運用コストなどが影響されると考えられる。本稿では、閾値を変動させたときの応答時間や稼働中のサーバの台数や稼働率、さらにはコスト (サーバの起動時間の合計値) の評価と、それに基づいて改良したスケーリングアルゴリズムについて述べる。

2 分散 Web システムの概要

本研究では図 1 に示すように拡張ロードバランサ、オリジンサーバ、取得したキャッシュを提供する VC サーバ群からなる分散 Web システムを実現する。負荷量の監視及びキャッシュサーバの増減は拡張プログラムを用いて行い、実際のリクエストの振り分けはロードバランサの機能を用いて行う。拡張プログラムの機能を以下に示す。

- A 負荷監視機能: サーバの負荷量を監視する。
- B キャッシュサーバ管理機能: 負荷量に応じて、仮想キャッシュサーバの起動・停止を行う。
- C 振分機能: アクセスの振り分け先を設定する。

3 スケーリングアルゴリズム

3.1 スケールアウトアルゴリズム

スケールアウトアルゴリズムは、負荷監視機能で観測された負荷量から負荷予測を行い、予測値が上限になる前にスケールアウトを行う。起動する台数 M は $TTLOR_t$, $TTLOR_{t-9}$, Th_{high} , S , N をそれぞれ現在の合計稼働率, 9 秒前の合計稼働率, スケールアウト

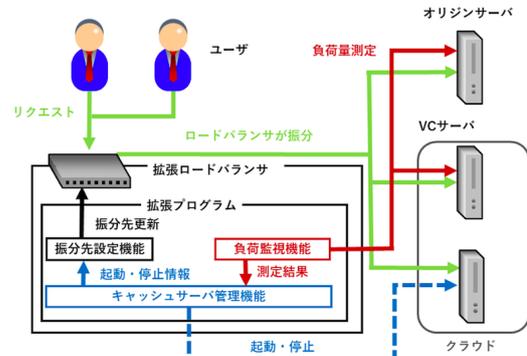


図 1 システムの構成

時の閾値、サーバを起動してから振分開始までにかかる時間、現在の稼働台数として以下の式 (1) で求める。

$$M = \frac{TTLOR_t + \frac{TTLOR_t - TTLOR_{t-9}}{9} \times S}{Th_{high}} - N \quad (1)$$

右辺の第一項は、 S 秒後に必要なキャッシュサーバ台数の予測値である。そこから現在の稼働台数 N を引くことで起動する台数 M を決定する。

3.2 スケールインアルゴリズム

スケールインアルゴリズムは、 $TTLORS_6$, Th_{low} , N , $Margin$ をそれぞれ最新 6 秒間の合計稼働率, スケールインの閾値, 現在のサーバ台数, マージンとして以下の式 (2) で決定する。

$$\frac{TTLORS_6}{6} < Th_{low} \times N - Margin \quad (2)$$

この式は各サーバの稼働率の平均が Th_{low} を下回ると真となり、サーバを一台停止する。マージンは測定誤差の影響を抑えるためのものである。

4 閾値評価

4.1 スケールアウトの閾値評価

スケールアウトのトリガーとなる閾値が、運用コストとクライアントへの応答時間に与える影響を調査するための実験を行った。実験環境として、仮想環境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、VC サーバ 9 台、クライアント 15 台を用いた。事前実験の結果より $S = 30$ とし、 Th_{high} を 0.1 から 0.9 まで変化させて実験を行った。

実験結果を図 2 に示す。実験中のサーバ起動時間の合計をコストとしてグラフ化している。 Th_{high} が小さいほどコストが高くなっているが、これは、 Th_{high} が低

Improvement and Evaluation of Scaling Algorithm for Distributed Web System

¹Masaya Matsuda, Kagawa University

²Keizo Saisho, Kagawa University

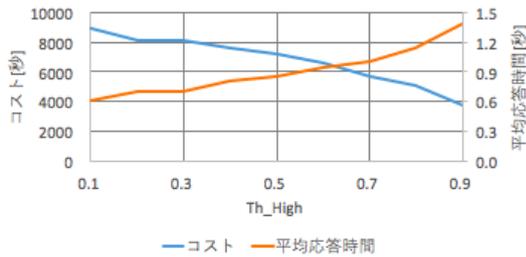


図2 スケールアウトの閾値を変動した実験結果

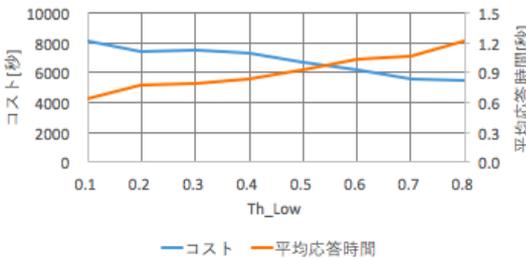


図3 スケールインの閾値を変動した実験結果

いほど、負荷上昇に対して素早く VC サーバを起動するためである。 Th_{high} が大きいほど平均応答時間が高くなっているが、 Th_{high} が 0.1 と 0.7 のときの平均応答時間の差と、 0.7 と 0.9 の差が 0.4 とほぼ同じ値になっており、 閾値が上がるほどに急激に応答時間が伸びることが分かる。 このことから、 コストを抑えた運用をしたい場合は Th_{high} を大きく設定し、 応答時間を重視した運用をしたい場合は Th_{high} を小さく設定することが適切であることが確認できた。

4.2 スケールインの閾値評価

スケールインのトリガーとなる閾値が、 運用コストとクライアントへの応答時間に与える影響を調査するための実験を行った。 実験環境はスケールアウトの閾値評価の実験と同一である。 スケールインの閾値の影響のみを見るため、 スケールアウトの閾値を 0.8 に固定し、 スケールインの閾値を 0.1 から 0.8 まで変化させて実験を行った。

実験結果を図 3 に示す。

Th_{low} が大きいほどコストが低く、 平均応答時間が長くなっているが、 これは Th_{low} が大きいほど、 負荷低下に対して素早く VC サーバを停止させるためである。 反面、 稼働台数が減ってしまうため平均応答時間が大きくなってしまふ。 このことから、 コストを抑えた運用をしたい場合は Th_{low} を高く設定し、 応答時間を重視して運用をしたい場合は Th_{low} を低く設定することが適切であることがわかる。

5 台数依存スケールリングアルゴリズム

各閾値の実験から、 運用方針に合わせてコストを重視するか応答時間を重視するか選択できることがわかった。 しかし、 サーバ台数が少ないとき急激な負荷増加に対して素早くスケールアウトをしなければ応答時間の

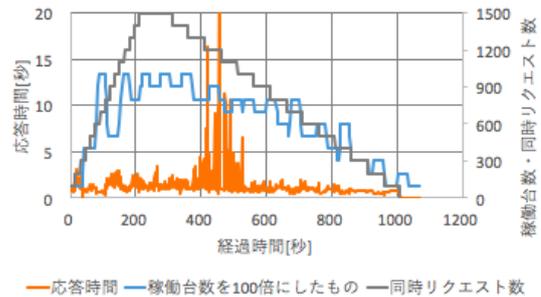


図4 台数依存スケールリングアルゴリズムの実験結果

低下が大きくなる。 これに対応するために、 サーバ台数に応じて各閾値を変動するアルゴリズムを提案する。 スケールアウトアルゴリズムは式 (1) を使用し、 スケールインアルゴリズムでは式 (2) に基づき、 Th_{low} の代わりに Th_{high} から差分 $Diff$ を引いた式 (3) を用いる。

$$\frac{TTLORS_6}{6} < (Th_{high} - Diff) \times N - Margin \quad (3)$$

式 (3) のアルゴリズムを評価するための実験を行った。 実験環境はスケールアウトの閾値評価の実験と同一である。 $Diff$ を 0.2 に、 台数ごとの閾値を表 1 に示したものに設定した。

表 1 各台数における閾値

台数	~3	3~5	5~7	7~9	9~10
Th_{low}	0.5	0.6	0.7	0.8	0.9

実験結果を図 4 に示す。 コストは最大で 5000 前後に抑えられていた。 0 秒から 200 秒の負荷増加中の応答時間は短くなったが、 500 秒以降の負荷減少中の応答時間は長くなった。 これは、 スケールイン・スケールアウトを繰り返していることが原因であると考えられる。

6 おわりに

以上、 負荷予測を組み込んだオートスケールリングアルゴリズムについて、 スケールアウトの閾値とスケールインの閾値の評価を行い、 それらを踏まえた改良スケールリングアルゴリズムについて述べた。 今後の課題として、 運用方針に基づく Th_{high}, Th_{low} の設定機能の追加、 台数に応じて閾値を変化するスケールリングアルゴリズムの改良、 ヘテロなクラウド環境における実験などがある。

参考文献

- [1] 堀内辰彦, 最所圭三, "クラウドに適した Web システムにおけるキャッシュサーバの負荷監視および負荷分散", 信学技報, vol.114, no110, IN2014-20, pp.79-84, 2014
- [2] 松田正也, 堀内辰彦, 最所圭三, "負荷予測を用いた分散 Web システムにおけるキャッシュサーバ管理機能の評価", 平成 28 年度 電気関係学会電子情報通信学会四国支部連合大会論文集, 16-2, p.211, 2016.9