

移動平均による分散 Web システムにおける振分アルゴリズムの改良

Improvement of Distribution Algorithm for Distributed Web System Using Moving Average

堀内 晨彦, 最所 圭三 Akihiko Horiuchi, Keizo Saisho
香川大学 Kagawa University

1 はじめに

我々は、クラウド環境において負荷量に応じて動的に仮想キャッシュサーバ数を増減させることで、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。図 1 に示すように、キャッシュサーバと大元のサーバ(オリジンサーバ)の負荷状況を監視し、負荷量に応じてクラウド環境で提供される仮想キャッシュサーバを起動・停止させ、それらにリクエストを振り分ける機能を拡張した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では、稼働中のサーバの平均負荷量が上限を超えるとリクエストを振り分けるキャッシュサーバを増加し、下限を下回ると減少させている。初期バージョンでは下限を固定値にしていたが、負荷が軽くなったときにほぼ同時に複数のキャッシュサーバへの振分を停止していた。稼働中のサーバ台数に応じて振分先削除の下限を自動決定するようにすることでこの問題は解決できたが [2], 振分先の追加と削除が繰り返される問題が発生した。

本稿では、この問題を負荷測定値の移動平均を用いて解決する手法について述べる。

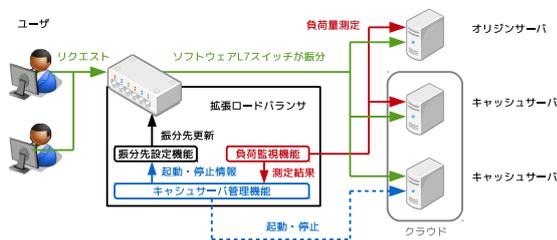


図 1 分散 Web システムの概要

2 負荷実験

本節では、文献 [2] で発生した問題点を説明する。

2.1 閾値の決定方式

キャッシュサーバは負荷量が $Th_{high} = 0.8$ 以上で 1 台追加、式 (1) 以下で 1 台削除する。例えば、稼働中のサーバ台数が n 台から $n-1$ 台に減少した場合、負荷量は $n/(n-1)$ 倍になる。この値が Th_{high} より小さくなければ、台数減少の反動による負荷の増加で再度キャッシュサーバを追加してしまう。そのため、測定値の揺らぎの影響を抑えるためのマージン m を引き、式 (1) を設定した。

$$Th_{low} = Th_{high} \times \frac{n-1}{n} - m \quad (1)$$

(n : 稼働サーバ台数, m : マージン)

2.2 実験環境

表 1 に示すような仮想環境において、システムの負荷実験を行った。Web サーバには Apache を用い、図 1 の負荷監視機能から Apache の最大処理数に対する現在の処理数の割合 (稼働率) を測定する。

ロードバランサが現在振分中の Web サーバの稼働率の平均値 (平均稼働率) を負荷量として用いる。また、各クライアントからの同時アクセス数は 100 とする (最大同時アクセス数 900)。

表 1 評価実験環境

	台数	スペック
ロードバランサ	2	CPU2 コア, メモリ 2GB
Web サーバ	9	CPU1 コア, メモリ 512MB
クライアント	9	CPU1 コア, メモリ 512MB

2.3 実験手順

下記に示すような手順で評価実験を行った。ロードバランサに対するアクセスを段階的に増減させ、その際の負荷量やサーバの稼働台数を調査する。また、図 1 のキャッシュサーバ管理機能は未実装のため、サーバは常に起動しており、振分先の切替のみを行う。

1. ロードバランサに対するアクセスが無い状態で開始
2. 30 秒ごとにアクセスするクライアントを 1 台追加
3. 全クライアントがアクセス中の状態で 30 秒維持
4. 30 秒ごとにアクセスするクライアントを 1 台削除
5. ロードバランサに対するアクセスが無くなると終了

2.4 実験結果

実験結果を図 2 に示す。150, 450, 500 秒付近で負荷量が変わらないのに稼働台数が上下していることが分かる。これは、負荷の測定値が閾値付近を上下し、ロードバランサに対して振分先の追加と削除が繰り返されているためである。その原因として、負荷の測定間隔が短すぎるために測定値のバラツキが大きいことが考えられる。また、実運用を考えると、サーバが起動直後にシャットダウンされる状態となり非効率である。

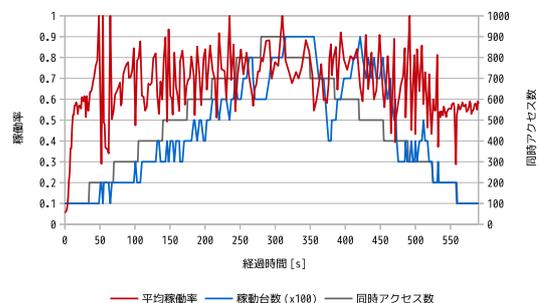


図 2 評価実験結果

3 移動平均機能

前節で示した振分先の追加と削除が繰り返される問題について、負荷測定値の移動平均を用いて解決する手法およびその評価について述べる。

3.1 移動平均

分散 Web システムは急激に負荷が増加した場合でも瞬時に対応し、ユーザへの応答性を維持することが求められる。そのため、サーバへの負荷の測定間隔は短くしたいが、測定値のバラツキが大きくなる問題がある。本稿では、負荷の測定結果を反映させつつバラツキを抑える(平滑化する)ために、移動平均を用いることにする。移動平均とは、直近の一定個数の測定値の単純な平均のことである。

3.2 実装方法

負荷測定値の移動平均を求める機能を実装した。一定期間の測定値を配列に保存し、その末尾から指定個数(平均区間)を取り出して平均値を求める。スケールアウト(サーバを追加)する場合とスケールイン(サーバを削除)する場合で、平均区間を区別する。(図3参照)

● スケールアウト

過負荷の場合は早急にサーバを追加し、負荷を分散する必要がある。そのため、平均区間を短くし、より早く測定値を反映させる。

● スケールイン

追加したサーバは一定期間稼働させる必要がある。そのため、平均区間を長くし、測定値のバラツキの影響を受けないようにする。これにより、振分先の追加と削除が繰り返されるのを防ぐ。



図3 移動平均の概要図

3.3 機能評価

実装した移動平均機能を評価する実験を行った。実験環境と実験手順は、第2.2節と第2.3節で述べたものと同じである。スケールイン時の平均区間は、スケールアウト時の平均区間の倍に設定した。

図4にスケールアウトで2回、スケールインで4回平均を用いた場合、図5にスケールアウトで4回、スケールインで8回平均を用いた場合の結果をそれぞれ示す。凡例の括弧内の数値は平均区間を表している。

図2と比較して、図4では稼働台数の上下が少ないことが分かる。しかし、150秒、250秒、400秒付近では、少数ではあるが台数増減の繰り返しが発生している。平均区間を大きくした図5は、図4よりも更に安定していることが分かる。問題であった台数増減の繰り返しも発生していない。

いずれの場合でも、移動平均によって測定値のバラツキが吸収され、稼働率のグラフの変化が緩やかになり、山や谷が小さくなっている。ただ、稼働台数と同時アクセス数(負荷)の変化を比較すると、平均区間が大きいほど稼働台数が負荷に遅れて反映されている。スケールアウト時の平均区間を大きくすると、負荷への追従が遅れるという副作用も確認できた。

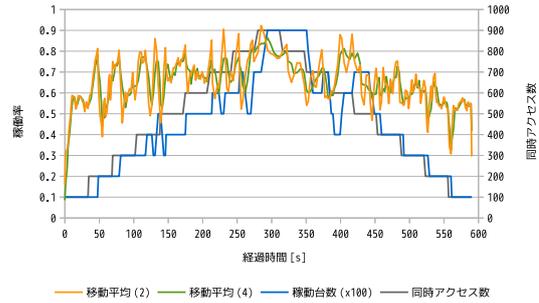


図4 スケールアウト2・スケールイン4での実験結果

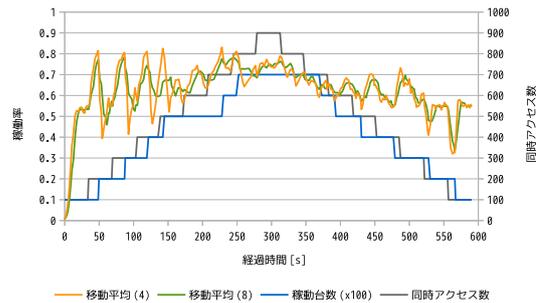


図5 スケールアウト4・スケールイン8での実験結果

4 おわりに

以上、移動平均を用いた分散 Web システムにおける振分アルゴリズムの改良について述べた。

サーバ台数を増やして実験を行ったところ、負荷の測定値が閾値付近を上下し、ロードバランサに対して振分先の追加と削除が繰り返される問題が発生した。その原因は、負荷の測定間隔が短すぎるために、測定値のバラツキが大きかったと考えた。そこで、一定回数ごとに測定値の移動平均を取ることで、そのバラツキを抑制する機能を実装した。振分先の追加時と削除時では求められる反応速度が異なると考え、平均を取る測定回数を変えて評価を行い、問題が解決できることを確認した。

今後の課題として、以下の点が挙げられる。

- 移動平均機能による HTTP の応答時間への影響を調査
- 他のシナリオ(急激なアクセスの増加や減少)による実験
- キャッシュサーバ管理機能の実装と評価

謝辞 本研究は JSPS 科研費 25330082 の助成を受けた。

参考文献

[1] 堀内農彦, 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年6月

[2] 堀内農彦, 最所圭三, "分散 Web システムにおけるキャッシュサーバへの振分アルゴリズムの改良", 平成26年度電気関係学会四国支部連合大会, 2014, p.254