L-002

キャッシュサーバを用いる分散 Web システムにおける応答性の改善
Improvement of Responsiveness in Distributed Web System Using Cache Servers

河野彰吾 † 最所圭三 †
Shogo Kono Keizo Saisho

1. はじめに

クラウド環境の発展により，クラウド上にキャッシュサーバを仮想マシンとして構築することが容易になり，これらを用いることで Web サービスの応答性を向上させることが可能となった．我々は，負荷量に応じて動的にキャッシュサーバ数を増加 (スケールアウト)・減少 (スケールイン) 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 Web システムを開発している [1][2]．開発中のシステムでは，Web システムのリクエストを処理しているサーバの負荷に応じてキャッシュサーバを起動し，アクセスを振り分けるが，キャッシュサーバの起動が完了するまではアクセスを振り分けることができない．そのため，キャッシュサーバの起動からサービスを提供できるようになるまでにタイムラグが生じ，その間の応答性が大きく低下していた．これを改善するために，本研究では，常時起動している 1 台のキャッシュサーバを用意し，起動中のキャッシュサーバの肩代わりをすることによって応答性を改善する機構を開発している．

2. 分散 Web システム

2.1. システム構成

文献 [2] では，図 1 に示すような，拡張ロードバランサと, キャッシュ元のコンテンツを持つオリジンサーバ，取得したキャッシュを提供する仮想キャッシュサーバ群を用いて分散 Web システムを実現している．拡張ロードバランサは，既存のロードバランサをベースにして以下の拡張プログラムが追加されている．

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: 仮想キャッシュサーバ数に応じたアクセスの振り分け先を設定する．

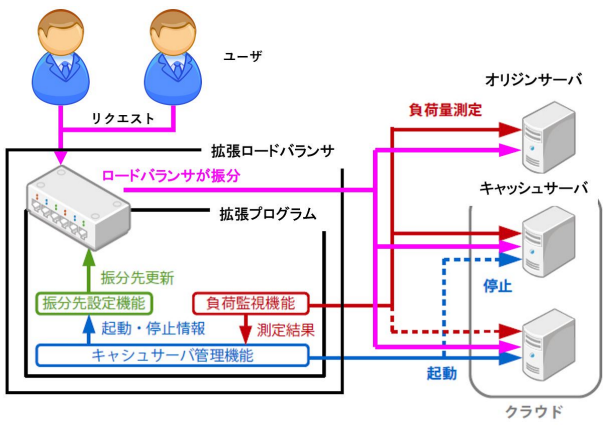

図 1: 分散 Web システム

負荷量の監視およ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，リクエストの制御はロードバランサの機能を用いて行う．本研究では，Web サーバプログラムとして Apache を用いており，稼働率を Apache の最大処理数に対する現在処理しているリクエスト数の割合とし，振分中のサーバ群の稼働率の平均値を負荷量として用いる．仮想キャッシュサーバの起動・停止の制御には，Th_{high} と Th_{low} の 2 つの閾値を設けている．

- スケールアウト

  測定した負荷量の増加率から，仮想キャッシュサーバの起動完了時の全体の負荷量を予測し，それを Th_{high} で割った値をそのときに必要な台数を求め，現在の起動台数よりも少ない場合は，(予測値 – 現在の起動台数) 台のキャッシュサーバを起動する．

- スケールイン

  合計稼働率が現在の稼働台数 $\times Th_{low}$ を下回るとキャッシュサーバを 1 台停止する．このとき，停止の対象となった仮想キャッシュサーバへのリクエスト振分を停止し，処理中のリクエストをすべて終了するまで待ち，仮想キャッシュサーバを停止する．

2.2. 負荷実験

本節では，先行研究のスケーリングアルゴリズムを用いた負荷実験の結果について述べる．実験には，クライアントから 60 秒かけて秒間リクエスト数が 0 から 700 になるまで分散 Web システムにリクエストを行った．分散 Web システムで用いているオリジンサーバおよびキャッシュサーバでは，最大で 1 秒あたり 200 のリクエストを処理できる Web サーバを用いている．仮想キャッシュサーバの起動・停止のしきい値である Th_{high} は 0.5，Th_{low} は 0.1 とする．また，仮想キャッシュサーバの起動には 30 秒かかる．

実験結果を図 2 に示す．稼働率 1 がオリジンサーバの稼働率，稼働率 2〜5 がそれぞれの仮想キャッシュサーバを稼働率を示す．図 2 の 20 秒付近で，オリジンサーバの稼働率が Th_{high} を超えているので，仮想キャッシュサーバを起動開始した．仮想キャッシュサーバが立ち上がるまでの 50 秒付近までは，オリジンサーバのみでリクエストを処理しているため，応答に時間がかかっており，失敗もいくつか見られる．仮想キャッシュサーバの起動後は，リクエストが振り分けられているため，そ

†香川大学, Kagawa University

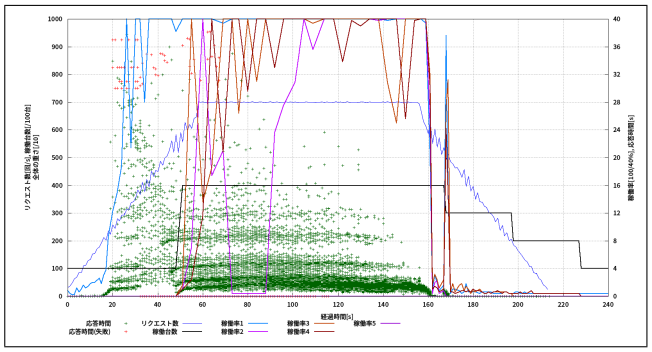

図 2: 先行研究の実験結果

の後の応答性は少しずづ改善されているが，仮想キャッシュサーバへリクエストが振り分けられるまでのタイムラグによって滞ってしまったリクエストの処理があるため，応答性の改善に時間がかかっている．本研究では，共有キャッシュサーバを用いることでスケールアウト時の応答性の改善を図る．

3. 共有キャッシュサーバを用いた分散 Web システム

3.1. システム概要

2.2 節より，負荷の上昇に応じて仮想キャッシュサーバの起動処理を行っても，起動が完了してリクエストが振り分けられるようになるまでには時間がかかるため，それまでリクエストを処理しているサーバの過負荷に対処できないということが確認できた．

本研究では，この問題に対し，仮想キャッシュサーバが起動し，リクエスト振分が開始されるまでの間，アクセスを肩代わりするキャッシュサーバを常時起動しておき，スケールアウトされるまでのリクエストをこのキャッシュサーバに振り分けることで，応答性の改善を図る．常時起動しているキャッシュサーバが存在する分，コストが増加するという問題が新たに発生するが，図 3 に示すように，常時起動しているキャッシュサーバを複数の Web システムで共有することで，分散 Web システム 1 つあたりにかかるコストの低減が可能である．このアクセスを肩代わりするキャッシュサーバを本稿では共有キャッシュサーバと呼ぶ．

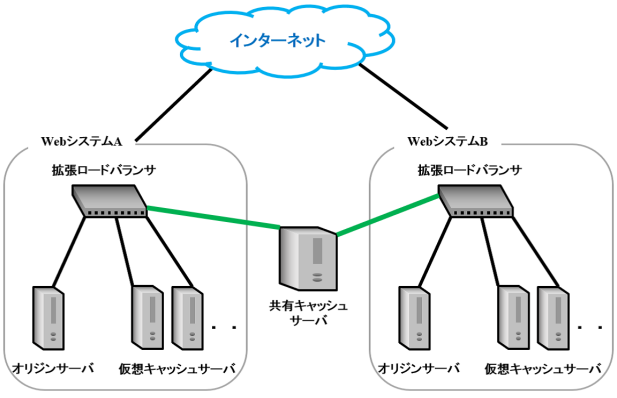

図 3: 共有キャッシュサーバを用いた分散 Web システム

3.2. スケールアウトアルゴリズム

本節では，共有キャッシュサーバを用いるスケールアウトアルゴリズムについて述べる．基本的には，先行研究のスケールアウトアルゴリズムを，仮想キャッシュサーバが起動するまでの間，共有キャッシュサーバにリクエストを振り分けるように改良する．しかし，仮想キャッシュサーバへのリクエスト振分の直後はキャッシュを保持していないため，オリジンサーバへのアクセスが定常状態よりも増えることに急激な負荷の増加が考えられるため，仮想キャッシュサーバへのリクエスト振分と同時に共有キャッシュサーバへのリクエスト振分を停止するのではなく，設定した時間後に振分を停止するようにする．これらを考慮したスケールアウトアルゴリズムは以下の通りである．

- 仮想キャッシュサーバ起動要求：
  - 共有キャッシュサーバへのリクエスト振分開始
  - 仮想キャッシュサーバの起動開始
- 仮想キャッシュサーバの起動完了：
  - 仮想キャッシュサーバへのリクエスト振分開始
- 仮想キャッシュサーバへのリクエスト振分開始から指定時間後：
  - 共有キャッシュサーバへのリクエスト振分停止

複数の Web システムで共有する場合は，2 つ以上の Web システムが同時に過負荷になったことで，同時に仮想キャッシュサーバの起動を行う場合もある．その場合は, それぞれの Web システムの負荷状況に応じて，共有キャッシュサーバへのリクエスト数を調整する必要があるが，本研究ではその前段階として，単一の Web システムにおいて共有キャッシュサーバを用いた分散 Web システムの開発を行う．

4. 共有キャッシュサーバを用いた評価実験

4.1. 先行研究との比較

共有キャッシュサーバの有効性を調べるため，本研究のアルゴリズムを用いた分散 Web システムの負荷実験を行い，先行研究の実験結果と比較する．実験環境は，先行研究の負荷実験と同様である．共有キャッシュサーバは，仮想キャッシュサーバが起動しリクエスト振分が行われるようになってから 30 秒後にリクエスト振分を停止する．

実験結果を図 4 に示す．稼働率 1 がオリジンサーバ，稼働率 2 が共有キャッシュサーバ，稼働率 3〜5 が仮想キャッシュサーバの稼働率をそれぞれ示す．図 4 の 20 秒付近で，オリジンサーバの負荷が上昇したため，仮想キャッシュサーバの起動処理が開始され，共有キャッシュサーバへのリクエスト振分が行われている．図 2 と比較すると，スケールアウト時の応答性が改善できていることが分かる．しかし，120〜180 秒付近で応答性が低下してしまっている．これは，応答性が改善されたことにより，リクエスト振分が不要と判断された仮想キャッシュサーバの停止が必要以上に多くなってし

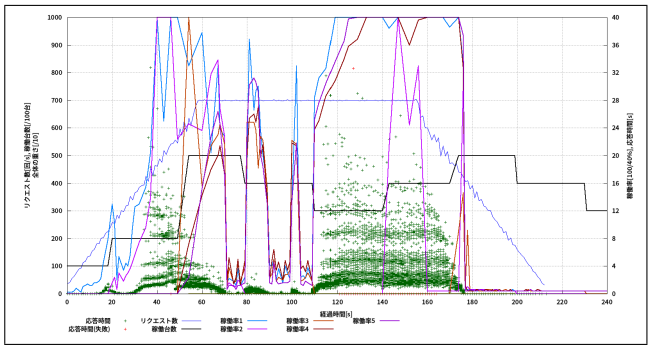

図 4: 共有キャッシュサーバを用いた実験結果

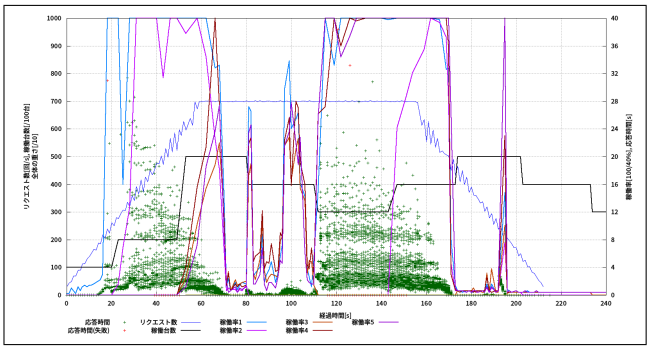

図 5: 共有キャッシュサーバへの振分割合:0.5

まっているために起きている．スケールイン時のアルゴリズムについては先行研究のものと変わらないため，これについても改良する必要である．

4.2. 共有キャッシュサーバへのリクエスト振分割合の評価

4.1 節の実験では，共有キャッシュへのリクエスト振分をオリジンサーバや仮想キャッシュサーバと同じ割合にしていたが，複数の Web システムで共有キャッシュサーバを用いる場合，同時に 2 つ以上の Web システムで過負荷になった時に，それぞれの Web システムに対して共有キャッシュサーバへ振り分けるリクエストの割合を考慮する必要がある．このため，共有キャッシュサーバへの振分割合を減らしたときの影響を調べることにした．実験では，オリジンサーバと仮想キャッシュサーバへのリクエスト振分割合を 1 として，共有キャッシュサーバへのリクエスト振分割合を 0.5 に設定した．

実験結果を図 5 に示す．図 4 と比較すると，あまり変わらないように見えるが，20〜40 秒付近のスケールアウトを行う部分の応答性が図 5 の方が悪くなっているが，共有キャッシュサーバを用いない図 2 と比較すると，十分な効果があることが分かる．

実際に応答性がどの程度変化しているかを調べるために，平均応答時間を比較する．図 6 に平均応答時間を比較したグラフを示す．ここで用いたデータは，図 2，図 4，図 5 と同一である．20〜40 秒付近を見ると，共有キャッシュサーバを用いない場合にと比べ，応答性が大きく改善できていることが分かる．また，20 秒〜35 秒の間の平均応答時間を見ると，共有キャッシュサーバへの振分割合が 0.5 の場合は共有キャッシュサーバを用いない場合と共有キャッシュへの振分割合が 1 の場合のほぼ中間の応答時間になっており，おおよそ割合通りの応答時間になっていた．スケールアウト時の応答時間が改善されており，共有キャッシュサーバの有効性が確認できた．

本実験では，リクエスト振分割合を静的に設定して応答性を確認したが，複数の Web システムでは，それぞれの負荷量やサーバの性能によって，必要な割合が変動することが予想される．そのため，複数の Web システムで運用する際には，共有キャッシュサーバへのリクエスト振分割合を動的に設定することによって，負荷に対してより柔軟に対応することが可能であると考える．

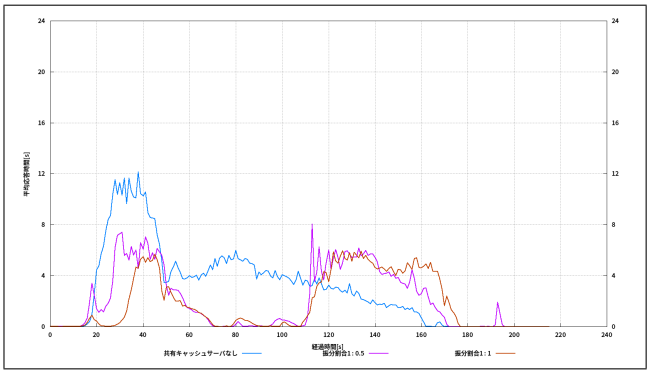

図 6: 平均応答時間

4.3. 急激な負荷に対する評価実験

これまでの実験では，負荷量の増加が比較的緩やかであり，実際の Web システムの負荷量は，大量のアクセスなどによって急激に負荷が増加することが予想される．そこで，急激な負荷が起きた場合の応答性を，共有キャッシュサーバを用いない場合，共有キャッシュサーバを用いたリクエスト振分割合が 1 の場合，共有キャッシュサーバを用いたリクエスト振分割合が 0.5 の場合とで比較する．これまでの実験では，60 秒かけて秒間リクエスト数を 0 から 700 になるまで増加させてきたが，今回の実験では，その時間を 10 秒 (6 倍の増加量) にした．

共有キャッシュサーバを用いない場合の実験結果を図 7 に，共有キャッシュサーバへのリクエスト振分割合が 0.5 の実験結果を図 8 に，共有キャッシュサーバへのリクエスト振分割合が 1 の実験結果を図 9 にそれぞれ示す．図 7 では，40 秒付近までオリジンサーバのみで大量のアクセスを処理しているため，応答に非常に時間がかかり，失敗してしまうものも多くなっている．図 8 と図 9 では，共有キャッシュサーバへのリクエスト振分がすぐに行われているが，共有キャッシュサーバがあっても負荷に対して処理できる限界を超えているため，応答性が改善されていないように見える．しかし，80 秒以降の応答時間を見ると，図 9 が最も良い応答時間の結果が出ていた．

次に平均応答時間で比較する．図 10 に平均応答時間を比較したグラフを示す．0〜40 秒付近を見ると，共有キャッシュサーバを用いない場合の平均応答時間に対して共有キャッシュサーバを用いる場合の平均応答時

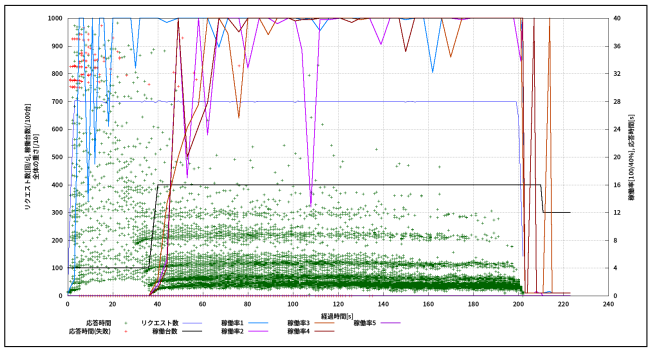

図 7: 急激な負荷実験 (共有キャッシュサーバなし)

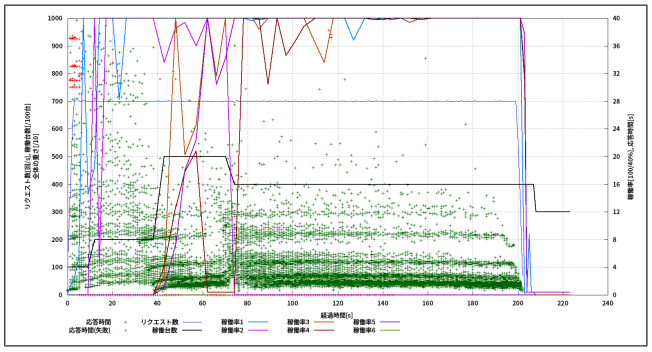

図 8: 急激な負荷実験 (共有キャッシュサーバ振分割合:0.5)

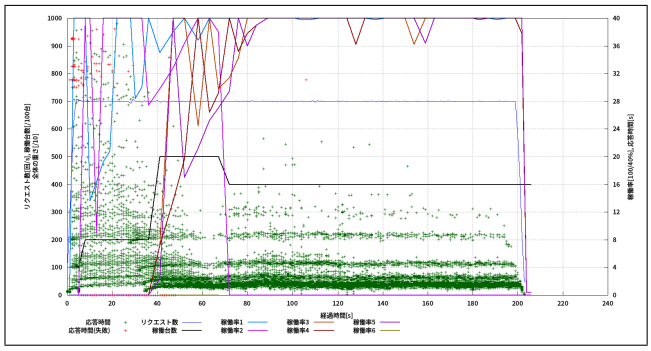

図 9: 急激な負荷実験 (共有キャッシュサーバ振分割合:1)

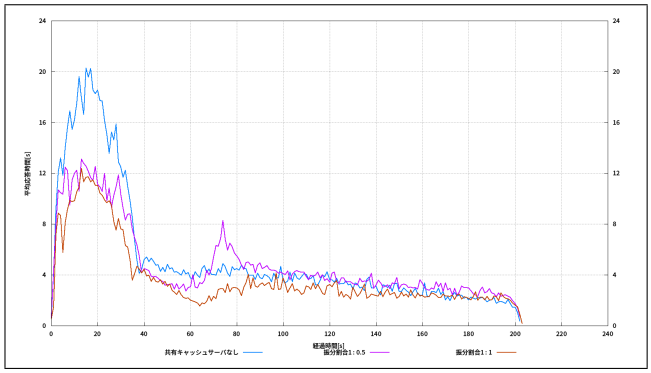

図 10: 平均応答時間 (急激な負荷実験)

間の方が短く，急激な負荷増加に対しても一定の効果があることが確認できた．共有キャッシュサーバへのリクエスト振分割合が 1 の場合と 0.5 の場合を比較すると，1 の方が短くはなっている．実験前は，図 6 に示すような差が出ると予測していたが，予測通りにはならなかった．この原因としては，急激な負荷増加の場合，共有キャッシュサーバへの振分割合を変化させても，割合通りに振り分けられない可能性があると考えられる．

複数の Web システムで共有キャッシュサーバを用いる場合，2 つ以上の Web システムが過負荷になったとき，共有キャッシュサーバの性能がそれらのリクエストを十分に処理できるかを考慮しなければならない．また，共有キャッシュサーバに要求される性能についても考える必要がある．

5. まとめ

先行研究の分散 Web システムのスケーリングアルゴリズムの評価により判明した問題点を踏まえて，共有キャッシュサーバを用いたスケーリングアルゴリズムを提案した．実験の結果，共有キャッシュサーバを用いることおよび提案したアルゴリズムの有効性を確認できた．今後の課題として以下の課題がある．

- スケールイン時の仮想キャッシュサーバを必要以上に停止してしまう問題の解決
- 複数の Web システムでの共有を行うための共有キャッシュサーバへの振分割合を動的に決定するアルゴリズムの開発
- 複数の分散 Web システムを用いた評価

参考文献

[1] 小笹光来，最所圭三 “クラウドに適した Web システムについて」，平成 24 年度 電気関係学会四国支部連合大会論文集”，17-14，p.360，2012.9

[2] 堀内晨彦, “分散 Web システムにおけるオートスケールアルゴリズムの改良と評価”, 香川大学 修士論文, 2016