

分散 Web システムにおける複数パラメタを用いたオートスケールアルゴリズム

20G477 畑 智裕 (最所研究室)

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動的に増減させるオートスケールアルゴリズムを開発している。これまで、スループットを負荷量として用いたが、負荷が増え始めている際に起動が間に合わず応答性が低下しているという問題があった。さらに、起動・停止を実際にしておらず起動時間分待ち、その後振り分け先に追加するという実際の環境に即していない実装であった。本研究では、サーバ群から取得する負荷情報を増やし、増やした負荷情報を用い、実際にサーバを起動・停止するオートスケールアルゴリズムの開発と評価について述べる。

1 はじめに

当研究室では、クラウドで提供されている仮想マシンをキャッシュサーバ (VCサーバ) として用い、負荷量に応じて動的に VCサーバ数を増減させることで、応答性を確保する分散 Web システムを開発している [1][2]。先行研究では、負荷量として稼働率 (同時にサービスできる最大数に対する実際の同時サービス数) を用いる方法とスループットを用いる方法で比較実験を行い負荷量としてスループットを用いるほうが適切であると示した [1]。本研究では、先行研究で開発されたシステムの問題点を解決し、複数パラメタを用いたオートスケールアルゴリズムの提案を行う。

2 分散 Web システム概要

図 1 に当研究室で開発している分散 Web システムの概要を示す。本システムは拡張ロードバランサ、コンテンツを提供するオリジンサーバ、オリジンサーバから取得したキャッシュを提供する VCサーバ群から構成される。拡張ロードバラン

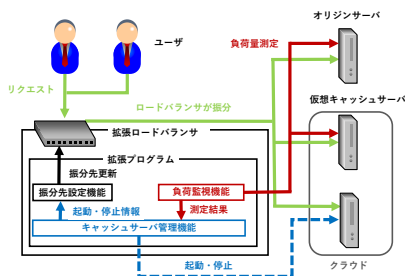


図 1: 分散 Web システムの概要図

サは、ソフトウェアロードバランサに、サーバの

負荷量を監視する負荷監視機能、負荷量に応じて VCサーバを起動・停止するキャッシュサーバ管理機能、VCサーバ数の増減に合わせてアクセスの振り分けを更新する振り分け設定機能を持つ拡張プログラムを追加したものである。負荷量の監視および VCサーバの増減は拡張プログラムの機能で行い、リクエストの振り分けはソフトウェアロードバランサの機能を用いて行う。

スループットを用いる負荷量の計算式を式 (1) に示す。式 (1) では、一定区間 (i) の全稼働サーバの合計スループットの移動平均 (TP_{MAiAll}) をそれぞれのサーバでの上限スループットの合計値 ($TP_{HighAll}$) で割ることにより現在の全体処理能力に対して掛かっている負荷量 (LO) を算出している。

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

負荷量がスケールアウトの閾値を超えると現在のサーバ数では処理することができないと判断し、VCサーバを起動し、振り分けを開始する。スケールインの閾値を下回ると余分な VCサーバが動作していると判断し、振り分けを停止し、VCサーバを停止する。

3 先行研究での問題点

先行研究にて負荷実験を行った際に、サーバ台数が少ない状態で負荷が増加すると起動が間に合わず応答性が低下しているという問題があった。さらに、先行研究でのシステムは実際にサーバの起動・停止をしておらず、事前実験にて調査した起動時間の最大時間分待ち、サーバを振り分け先

に追加するという実際の運用状況に即していない状態であるという問題があった。

そこで本研究では、負荷量がスループットのみではサービスに変更が起きる場合対応しきれないため、サーバから取得する負荷情報を増やし、負荷量として用いることと実際に起動・停止を行う機能の作成を行うことにより解決する。

負荷量として追加して取得する負荷情報を以下に示す。

- CPU 使用率
- メモリ使用率
- ディスク I/O
- ソケット数
- 送信・受信バイト数

4 起動・停止機能

起動に関しては、キャッシュサーバの起動、その後、サービスが提供可能になった状態を起動完了とする。そのため、起動のアルゴリズムは以下に示すようになる。先行研究では、既に起動していたためそのまま振り分け先に追加していた。しかし、本研究では、実際に起動しているため、起動後サービスが開始されるまで時間が必要である。疎通確認に関しては、失敗すると1秒待機を行っている。HEAD アクセスであるのは、Body は必要でなく疎通の確認をより早く行うためである。停止に関しては、既に振り分けられたリクエストの処理が完了する時間待ち停止に入るため、停止のみを行う。

1. 各仮想環境に応じたキャッシュサーバの起動完了
2. Apache に対して、HEAD アクセスで疎通確認
3. 規定回数 2. を繰り返す。(疎通できれば 4. へ、出来なければ停止・失敗)
4. エージェントに対して、HEAD アクセスで疎通確認
5. 規定回数 4. を繰り返す。(疎通できれば成功、出来なければ停止・失敗)

5 複数パラメタを用いたオートスケールアルゴリズム

変更したスケールアウトの判定式を以下に示す。CPU 負荷情報、スループットのスケールアウト

判定の論理和を用いたもので行う。論理和を用いるものは応答時間の改善を図るため、スケールアウトの条件を緩和するために論理積でなく論理和を用いることにした。

$$LO_{CPU} > CPU_{high} \cup LO > Th_{high} \quad (2)$$

6 上限スループットの予測

上限スループットの予測アルゴリズムについて述べる。本論文では、対象としているサービスが DokuWiki であり、CPU 負荷のサービスであるため、CPU 負荷のサービスの上限スループットの予測を行った。予測した上限スループットを $TP_{highpre}$ 、スループットの移動平均を TP_{MA} 、CPU 使用率の移動平均を CPU_{MA} とし、算出式を以下に示す。

$$TP_{highpre} = \frac{TP_{MA}}{CPU_{MA}} \times 0.7 \quad (3)$$

0.7 をかけているのは、実験にて、上限スループットに達した際の CPU 使用率が 0.7 であったためである。

7 まとめ

先行研究にて発見された問題点を解決するため複数の負荷情報を取得するエージェントの開発を行った。実装されていなかった起動・停止機能の開発を行い、振り分け先追加までのアルゴリズムを開発し、機能評価を行った。対象サービスが追加した負荷情報から CPU の負荷であることが分かったため、CPU 使用率も負荷量として使用するアルゴリズムを提案・評価し、負荷を分散できていることを確認した。取得した負荷情報から上限スループットを推定するアルゴリズムの提案・評価を行い有効であることを確認した。

参考文献

- [1] 松田正也, 最所圭三, “クラウドを用いた分散 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, 2018.
- [2] 畑智裕, 最所圭三, “分散 Web システムにおけるロードバランサを用いたオートスケールアルゴリズムの改良と評価“, 香川大学学士 論文, 2019.