7ZA-05

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

畑 智裕† 最所 圭三‡

香川大学†‡

1. はじめに

我々の研究室では，クラウド環境を対象にした負荷量に応じて動的にキャッシュサーバ(VC 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 システムを開発している[1][2]．先行研究[2]において，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していることを示した．しかし，多量の画像ファイル等が添付されている Web サイトでは，スループットが期待したほど増加せず，その原因は，NIC の通信帯域が限界値に達したことであった．以上のことから，スループットのみではサーバの正確な負荷を知ることができないことが分かった．本研究では，CPU 使用率等のサーバの負荷を表す情報も取得し，それらを組み合わせたオートスケールアルゴリズムの提案を行う．

2. 分散 Web システムの概要

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 サーバの台数の増減に合わせてアクセスの振分先を更新する振分先更新機能を持つ拡張プログラムを追加したものである．負荷量の監視及び 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．

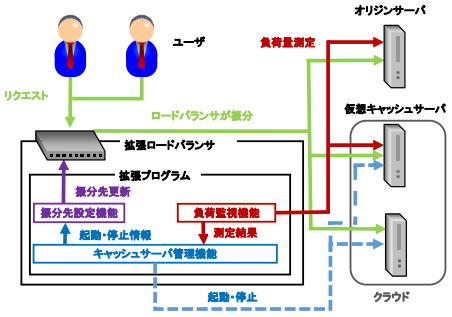

図 1 分散 Web システムの概要図

Autoscale Algorithm with Multiple Parameters in Distributed Web System

†Hata Tomohiro・Kagawa University

‡Saisho Keizo・Kagawa University

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かかっている負荷量(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台数の 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. 負荷情報の取得

サーバの負荷情報として，CPU 使用率，メモリ使用率，ディスク I/O，ソケット数，送信・受信バイト数を用いることにした．それらの情報が影響されるサービスの例を以下に示す．

・CPU 使用率:CGI プログラムを実行する場合等
・メモリ使用率:複数のリソースをメモリ中にキャッシュした場合等
・ディスク I/O:複数のリソースの読み出し，書き込みを行う場合等
・ソケット数:コネクションを維持する場合等
・送信・受信バイト数:大量のデータ，大きいデータの送受信する場合等

負荷情報を取得するためにサーバ上で動作するエージェントを開発した．開発したエージェントについて述べる．開発を行っている環境が Linux のため，エージェントは Linux 対応になっている．Linux では，/proc の下のファイルの形で各情報が提供されている．エージェントは，必要な情報に対応したファイルから該当箇所を抜き出す．Web サーバプログラムとして Apache を対象としており，Apache の server-status にローカルでアクセスすることで Apache の稼働情報を

取得する．取得した情報は JSON 形式で送信し，メモリ使用率，ディスクの総読み書き量，使用 CPU 時間，待機 CPU 時間，Apache の稼働率，Apache の総処理量，Apache 計測での秒間リクエスト数，総送信バイト数，総受信バイト数，上限ファイルディスクリプタに対するソケット数の割合，取得時刻，エラー情報の順番で送信される．CPU 時間等の累積値で格納されている情報に関しては，負荷監視機能が前回値との減算を行ってから使用率等の計算を行う．

エージェントは指定ポートを Listen 状態で待機しており，拡張ロードバランサの負荷監視機能は対象のサーバの該当ポートにアクセスすることで負荷情報を取得する．

DokuWiki をサービスしている Web サーバにエージェントを稼働させ，負荷をかけていない状態で 3 回，負荷をかけた状態で 3 回行った情報取得の様子を図 2 に示す．各値が増えていることが分かり，負荷の情報を取得できていることが分かる．使用 CPU 時間の増加率が一番大きいことから，CPU 負荷のサービスであることが分かる．

図 2 情報取得時の送信内容

4. 複数パラメタを用いたアルゴリズム

1.にて述べた通り，いずれかのリソースが上限値に達した場合，他のリソースが上限値に達していなくても，過負荷状態になり，それ以上の応答が不可能になる．そのため，単一のリソースのみで負荷状況の判定を行っていると提供サービスの切り替えを行うようなサービスや設定をそのまま利用して違うサービスの運用を行おうとした場合に正確な負荷の判定が難しくなる．以上のことから，複数のリソース情報を利用する負荷判定アルゴリズムが必要である．提案するアルゴリズムについて述べる．

3.にて，追加した複数の負荷情報にそれぞれの閾値を設定する．全ての負荷情報に対して，閾値を超えているかの判定を行い，全ての成否の論理和を取る．こうすることにより，いずれかのリソースが上限値に達したことを検知することができる．スループットを用いる先行研究[2]のアルゴリズムでは，上限スループットを調べるために事前実験が必要であり，サービスの種類が変わることによる上限スループットの変動に追従できなかった．本稿で提案するアルゴリズムではそのような問題はないと考えている．

CPU 負荷のサービスである DokuWiki に対して，Gatling に最大同時リクエスト数を 3,500 に設定して実験を行った結果のスループット，CPU 使用率，DiskIO，メモリ使用率，送信バイト数，ソケット数の割合をまとめたグラフを図 3 に示す．図 3 から増加の様子は少し異なるが CPU 使用率が 100%近くに到達している 350 秒付近にてスループットも増減し，限界を迎えている．他の DiskIO 定期的にスパイクが発生しており，メモリ使用率は 8 割程度で横ばい，送信バイト数は限界が来た時に急激に増加しているが，そのときでも通信帯域の 75%以下になっている．ソケット数の割合はとても小さい値となっている．このことから，CPU 負荷のサービスであることが分かる．このように，負荷と伴って増加する情報を判別し，負荷量として用いれば，どのような負荷にも対応できるようになると考える．

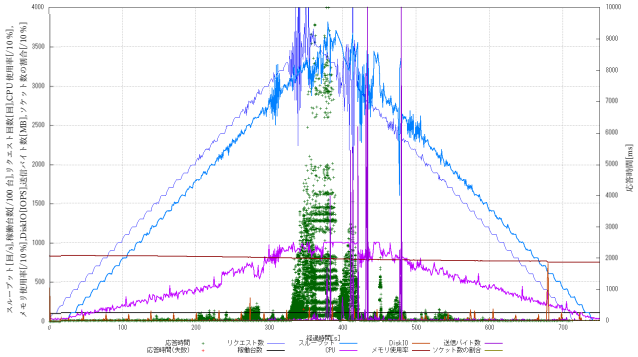

図 3 実験結果

5. おわりに

先行研究の分散 Web システムに実装されているアルゴリズムの問題点を踏まえて，負荷監視機能にて取得する情報を増やし，それらを用いたアルゴリズムの提案を行った．

参考文献

[1] 松田正也，最所圭三，“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”，情報処理学会第 81 回全国大会講演論文集，2W-07，pp. 3-133 － 3-134，2019.

[2] 畑智裕，最所圭三，“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” 令和 2 年度電気・電子・情報関係学会四国支部連合大会，16-1，2020.