

## クラウドに適した Web システムの負荷監視機能の改善について

Proposal of a Function to Improve Load Observation of Web System Fit for Cloud

堀内 晨彦, 最所 圭三 Akihiko Horiuchi, Keizo Saisho

香川大学 Kagawa University

### 1 まえがき

クラウド環境の発展に伴い、キャッシュサーバをクラウド上に構築することで Web サーバのサービス能力を向上させることが容易になった。我々は、クラウド環境において負荷量に応じて仮想キャッシュサーバ数を増減させることで、運用コストを最適化する Web システム<sup>[1]</sup>を開発している。ここで、キャッシュサーバ数を最適化するためには、Web サーバにかかる負荷を正確に把握する必要がある。HTTP の応答時間を負荷量としてシステムを試作したところ、測定精度が非常に悪かった。本稿では Web サーバの CPU 使用率と処理数を、Web サーバデーモンと外部プログラムで測定する方法について述べ、その結果を示す。

### 2 システムの概要

提案する Web システムの概要を図 1 に示す。拡張 L7 スイッチはユーザからのアクセスを大元のサーバ(オリジンサーバ)や仮想キャッシュサーバに振り分けると共に、稼働中の Web サーバの負荷量に応じて仮想キャッシュサーバ数を増減させる。本研究では、ソフトウェア L7 スイッチに以下の機能を拡張することで提案のシステムを実現する。また、Web サーバデーモンとして代表的な Apache を用いる、

- A. サーバの負荷量を監視する。
- B. 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. 仮想キャッシュサーバ数の増減に合わせて、サーバ数に応じたアクセスの振り分け先を設定する。

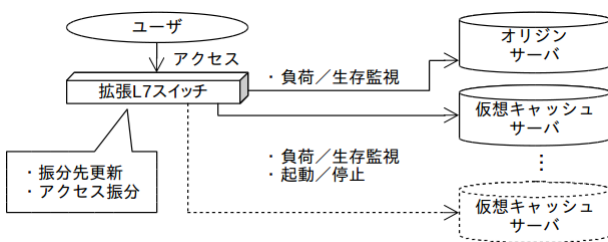


図1 クラウドに適した Web システムの構成

### 3 システムの評価

機能 A と C を実装したプロトタイプを作成した。ここでは負荷量として HTTP の応答時間を、L7 スイッチとして UltraMonkey を用いている。HTTP の応答時間が 800ms を上回ると過負荷、200ms を下回ると過負荷を脱したとみなす。同時アクセス数をオリジンサーバのみで処理可能な 1000 から、処理不可能 2500 に増加させた場合の実験結果を図 2 に示す。なお、機能 B は未実装であるため、仮想キャッシュサーバは常に起動している。

図 2 の①では、アクセス数の増加に伴いユーザから見

た応答時間が 800ms を超えたため、キャッシュサーバへの振り分けを開始し、応答時間が早くなった。しかし、②では負荷がかかっているにも関わらず応答時間が 200ms 以下になったため、キャッシュサーバへの振り分けを停止した③。この結果から、試作システムではサーバの負荷量を正確に測定できないことが分かった。

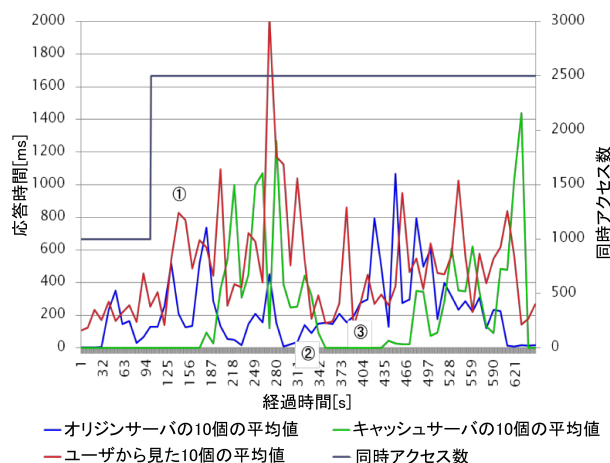


図2 HTTPの応答時間による負荷量の測定

### 4 負荷量の測定方法の検討

以下を負荷量のパラメタとして用いることが可能であるか検討する。以降、a を CPU 使用率、b を稼働率とする。

- a. Web サーバの CPU 使用率
- b. Apache の最大処理数に対する現在の処理数の割合

以下の方法を用いて、これらの負荷量を測定する。

- i. Apache の server-status で負荷量を測定
- ii. 外部プログラムで負荷量を測定

方法 i の server-status は CPU 使用率やプロセスの状態など、Apache の動作状況を Web ページとして出力する。ここで出力される CPU 使用率は Apache が起動してからの CPU 使用率の平均値である。この Web ページを取得し解析することで、負荷量を得る。

方法 ii では、Web サーバ上で負荷監視機能からのアクセスに応じて /proc/stat を解析して CPU 使用率を測定したり、Apache の処理数を数えて稼働率を測定したりする Ruby のスクリプトを作成し、負荷量を得る。

### 5 負荷量の測定方法の評価

前節で示した負荷量の測定方法により、サーバの負荷(アクセス数)の増減に伴ってどの程度追従できるかを調べる実験を行った。

図3に示すように Web サーバに対して負荷テストツールである Apache JMeter を用いて負荷をかけ、その時の負荷量を負荷測定プログラムを用いて測定する。実験の環境及び条件は以下の通りである。

- CPU を 2 コア、メインメモリを 1GB 割り当てた KVM 上の仮想 Web サーバに対して行う。
- Web サーバデーモンの最大同時処理数 (MaxClient) を 200 に設定する。
- Apache JMeter を用いてアクセス数を 1 秒当たり 5 個ずつ増加させ、合計 600 個になるまでリクエストを発生し続ける。
- テストページとして、PHP 製の Wiki に対してアクセスを行う。
- Web サーバデーモンを起動しアクセスが無い状態から 10 秒後にアクセスを開始する。

方法 i での測定結果を図 4 に、方法 ii での測定結果を図 5 に示す。実験では、前の測定が終了してから 1 秒後に次の測定を開始する。そのため、プロットの間隔は負荷量の測定にかかった時間+1 秒になる。

どちらの方法でもスレッド数の増加に伴い、CPU 使用率及び稼働率が上昇しているが、図 5 の CPU 使用率は CGI を処理するため、すぐに 100% に達している。図 4 では稼働率が 1.0 に達すると、次の測定までに 1 分以上かかっている。これは Web サーバデーモンの最大同時処理数以上のアクセスがあり、過負荷になっているためである。また、どちらの図でもアクセスが終わってから負荷量が低下するまでに 20 秒程度かかっているが、これは Web サーバが各スレッドの行った最後のアクセスを処理しているためである。

以上の結果から、アクセス数に応じて増減する稼働率が負荷量に適していることが分かる。CPU 使用率は同時アクセス数が少なくとも 100% に達するため、過負荷の判別ができない。また、稼働率の測定方法としては、Web サーバが過負荷になった場合でも短時間で測定が行える方法 ii が適しているが、過負荷を脱してから稼働率が低下するまでに時間がかかる。反対に方法 i では過負荷を脱すると直ぐに稼働率が低下するが、過負荷になった場合の測定に時間がかかる。

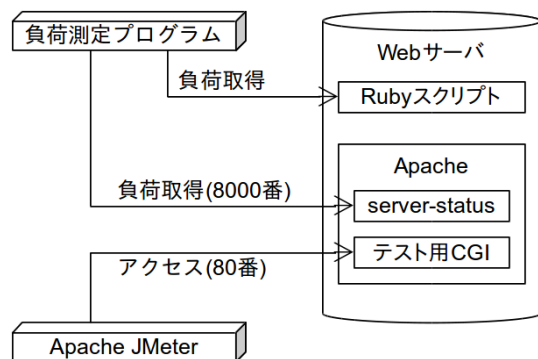


図3 負荷量の測定実験の概要

## 6 まとめ

以上、仮想キャッシュサーバを用いるクラウド適した Web システムの概要及び試作システムの評価について述べ、その負荷量の監視機能の改善方法を提案・評価を行った。実験から稼働率が負荷量に適していると分かったが、測定方法はどちらも課題があるという結果になった。今後は改善のために以下の方法を検討する。

- 過負荷でない平常時には方法 i を、過負荷になると方法 ii を用いる測定。
- 本研究室で開発している次回アクセスを保証する Web システムに実装中の優先アクセス機構<sup>[2]</sup>を用いて server-status へのアクセスを優先した場合の方法 i による測定。

最終的には、改善した負荷量の測定方法をシステムに組み込み、仮想キャッシュサーバの起動・停止を含めた評価を行う。

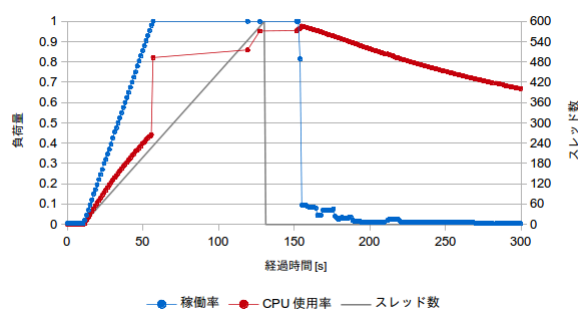


図4 Apache の server-status による負荷量の測定

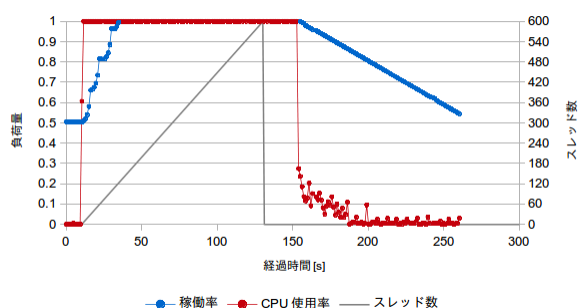


図5 Ruby のスクリプトによる負荷量の測定

## 謝辞

本研究は JSPS 科研費 25330082 の助成を受けた。

## 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, 2012, p.360
- [2] 山田茂和, 最所圭三, 松浦正尚, "次回アクセスを保証する NAP-Web の予測精度の評価", 平成 24 年度 電気関係学会四国支部連合大会, 2012, p.361