

# クラウドに適したキャッシュサーバを用いる Web システムにおける 管理機構の開発及びその評価

11G472 小笹光来(最所研究室)

本稿では、機能を拡張した L7 スイッチとキャッシュサーバを用いる事で、クラウド環境において負荷に応じて動的なキャッシュサーバの増減とリクエストの振分けを行い、運用コストを低減しつつ負荷の変動に柔軟に対応できる Web システムを提案しその評価について述べる。

## 1. はじめに

現在、インターネットでのサービスは増大し続けており、負荷分散の手段としてキャッシュサーバを用いる方法が一般的に見られる。近年ではクラウドを用いて容易にキャッシュサーバを構築出来るようになったが、キャッシュサーバ数を負荷量に応じて動的にアップダウンできなければ無駄なコストがかかる。本研究では、クラウド環境において負荷量に応じ動的にキャッシュサーバ数をアップダウンさせ、リクエストを適切に振分けることで運用コストを抑制しサービスを柔軟に管理することができる Web システムを設計し評価を行う。

## 2. システムの概要

1 節で述べた様な Web システムを構築する際の問題点として、どの様にして負荷量に応じたキャッシュサーバ数に制御するかという管理の問題や、アップダウンさせたサーバに対しどの様にリクエストを振分けるかというリクエスト振分けの問題がある。

これらの問題を解決する方法として DNS ラウンドロビンを用いる方法や、L7 スイッチを用いる方法がある。しかし例えば、DNS ラウンドロビンはヘビーユーザが一つの通信先に集中してアクセスすると負荷を分散する際に予期せぬ偏りが起こってしまったり、分散先サーバの障害検知が不可能であるという問題がある。そこで、本研究ではそれらの問題に対応出来る、ソフトウェア L7 スイッチを用いる事にした。提案システムの実現性を高められ、将来的に追加機能の実装や、既存機能の変更を行うことも可能である。

本研究では、ソフトウェア L7 スイッチである UltraMonkey[1]を機能拡張するプログラムを開発・評価し、提案システムを構築する。UltraMonkey と機能拡張プログラムの組合せを拡張 L7 スイッチと呼ぶ。

## 3. 拡張 L7 スイッチの機能

拡張 L7 スイッチに必要な機能は以下のとおりである。C)は UltraMonkey の機能を利用し、A),B)の機能を機能拡張プログラムとして実装した。

### A)監視機能

サーバを監視する機能であり生存監視と負荷監視がある。生存監視はサービスやシステムが過負荷などによ

りダウンしていないか UltraMonkey の機能を用い監視する機能であり、負荷監視はキャッシュサーバ増減機能に対しサーバの増減を指示する機能である。サーバ負荷量の判定はサービスの種類による影響が少ないレスポンスタイムを用いる。HTML ファイルの受信にかかる時間だけでなく、接続にかかる時間も考慮しユーザから見たレスポンスタイムを測定するため HTTP\_Ping と呼ぶ測定プログラムを実装する。

負荷監視は測定したレスポンスタイムが指定した閾値に応じてサーバのアップダウンを行う。しかし閾値が1つだと図 1(a)の様にレスポンスタイムのばらつきによりアップダウンが繰り返されてしまい、安定したサービスを提供出来ない。この為図 1(b)のグラフの様に 2 の閾値 (Th\_high, Th\_low)を用いヒステリシスを設ける事で問題を解決する。

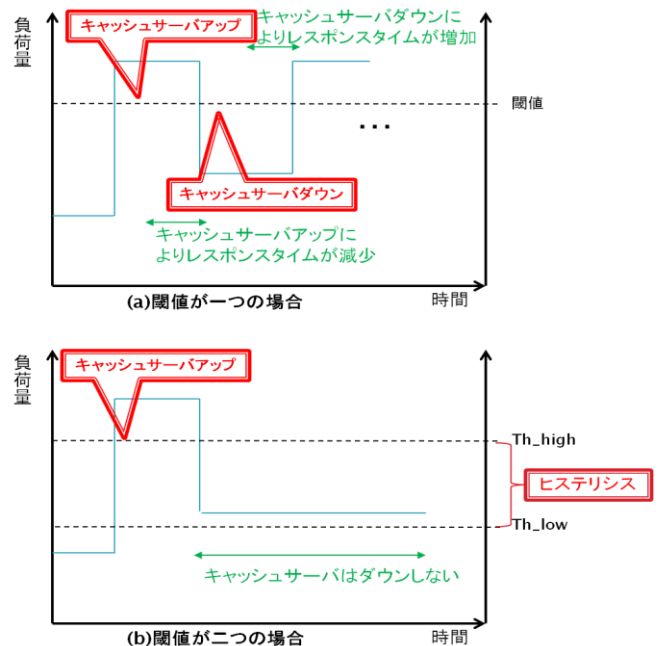


図1. 閾値とヒステリシス

### B)キャッシュサーバ増減機能

キャッシュサーバをアップダウンさせるために仮想マシンの起動停止を行う。サービスが可能になるまでの時間は仮想マシンを起動/停止する操作により異なるので ON, シャットダウン, サスペンドを行う機能を実装し、仮想マシンを状況に応じて制御する。

### C)リクエストコントロール機能

キャッシュサーバのアップダウンに伴いリクエストを振分ける機能である。UltraMonkey の持つ振分けアルゴリズムに従い振分けを行う。負荷監視機能がキャッシュサーバ数を変更すると共に UltraMonkey の設定ファイルを書き換えて振分け先を設定する。振分けアルゴリズムにはラウンドロビンや重みつきラウンドロビンがあり、通常ラウンドロビンを用い、サーバに性能差がある場合重みつきラウンドロビンを用いる。

### 4. 評価

図 2 はプログラムの関係図である。機能拡張プログラムは監視処理を行う Load watch program や、キャッシュサーバのアップダウン処理を行う UP DOWN program からなる。キャッシュサーバのホスト名や IP アドレスなどの情報は設定ファイルに記述されている。

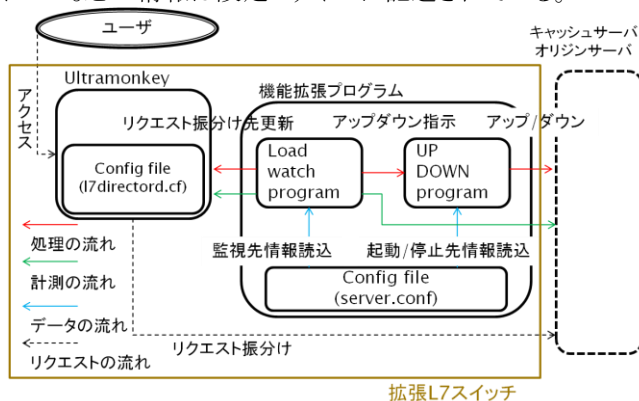


図 2. プログラム関係図

負荷をかけるクライアント、オリジンサーバ、キャッシュサーバをそれぞれ 1 台ずつ用いる。初めにオリジンサーバ単独で十分に処理できる量の負荷をかけておき、一定時間後にオリジンサーバ単独では処理できない量の負荷をかけた。Th\_high が 800[ms]を上回るとキャッシュサーバをアップし、Th\_low が 200[ms]を下回るとキャッシュサーバをダウンさせる。この時 HTTP\_Ping を用い、オリジンサーバ、キャッシュサーバ、拡張 L7 スwitch のレスポンスタイムを測定した。拡張 L7 スwitch のレスポンスを負荷監視機能から測定する事で全体の負荷を測定している。図 3 は各サーバのレスポンスタイム測定結果である。

図 3 の①では Jmeter のスレッド数の上昇に伴い、平均レスポンスタイムが増加し 800[ms]を超えたのでキャッシュサーバがアップしていた。キャッシュサーバでの処理が始まり、キャッシュサーバへの振分けが始まると共にオリジンサーバの平均レスポンスタイムが減少している事が分かる。キャッシュサーバに負荷が分散された事により、②に達すると、ユーザから見た平均レスポンスタイムが小さくなり、200[ms]を下回った。これによりキャッシュサーバがダウンし、オリジンサーバの平均レスポンスタイムが増加した。

設定した負荷量でキャッシュサーバのアップとダウンが行われ、オリジンサーバとキャッシュサーバの負荷は各サーバ単体を同条件で負荷テストした場合よ

りも軽減されていた。この事から動的管理と、負荷分散が行われている事が確認できた。しかし、③では負荷がかかっているにも関わらず負荷が減少しており、調査を行う必要がある。計測精度に問題がある可能性があり、今後の課題とした。

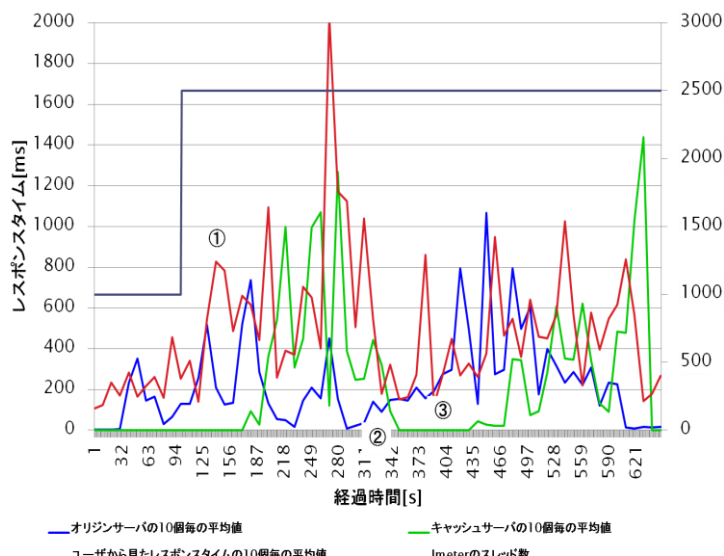


図 3. 各サーバのレスポンスタイム

### 6. まとめ

システム要件をまとめて設計を行い、監視機能、キャッシュサーバ増減機能を実装・評価した。負荷監視機能、キャッシュサーバ機能それぞれの機能の正常な動作を確認し、キャッシュサーバを新たにアップ・ダウンさせた後にレスポンスタイムが変化する事を確認した。

今後は未実装機能の実装や課題解決の為の機能追加を行う必要がある。今後の課題としては主に以下の様なものがある。

- ・閾値決定の自動化  
今回は予備実験を行いアップダウンに適切な閾値を計測した。アップダウンに適切な閾値をサーバ毎に自動で設定する機能が必要である。
- ・システムのクラスタ構成  
負荷の集中により拡張 L7 スwitch の処理がボトルネックとなる事を防ぐため、拡張 L7 スwitch とキャッシュサーバをクラスタ構成する機能が必要である。
- ・リクエスト振分けアルゴリズムの設計と実装  
オリジンサーバでしか処理できないリクエストをオリジンサーバへ振分ける機能が必要である。
- ・レスポンスタイム測定の精度向上  
本研究では直近 10 個のレスポンスタイムの平均を取り負荷監視機能で利用したが、レスポンスタイムのばらつきを考慮した測定を行う機能が必要である。

### 参考文献

- [1]UltraMonkey-L7,  
<http://sourceforge.jp/projects/ultramonkey-l7>  
2013 年 2 月 21 日