

# キャッシュサーバを用いた分散 Web システムにおける キャッシュ更新機構の開発

## Development of Cache Update Mechanism in Distributed Web System Using Cache Server

16G463 島山 侑也(最所研究室)

### 1. まえがき

当研究室では、Web サーバの負荷量に応じて稼働するキャッシュサーバの台数を制御する分散 Web システムの開発を行っている[1]. 先行研究では、Web サーバが過負荷状態にある場合でも、キャッシュ更新のためのキャッシュサーバからのアクセスに対するオリジンサーバの応答性を確保することで、キャッシュの品質を向上させる優先アクセス機構の開発が行われている[2]. 本研究では、キャッシュ更新のためのリソースが制限されている場合に、できる限り高い品質のリソースを提供するためのキャッシュ更新機構の開発を目指す。本稿では、動的コンテンツのキャッシュを行う際の問題を抽出し、その解決についての検討及び動的コンテンツのためのキャッシュ更新機構の基本的な部分の実装に着いて述べる。

### 2. 動的コンテンツのキャッシュ問題

画像、文書などの静的なコンテンツであれば問題無くキャッシュ可能である。しかし、ユーザからの問い合わせやユーザ情報を元に Web ページを生成する動的なコンテンツをキャッシュしようとする場合、キャッシュの効果やキャッシュ可能性について考慮しなければならない。動的コンテンツは、ユーザ情報やリクエスト、時刻など、様々な条件に応じて変化するが、変化が少ないものについては短期間のキャッシュが可能であると考えられる。しかし、Web 検索サイトの検索結果については全てキャッシュすることが、非効率的になることもある。また、ユーザを識別するための情報、特に個人情報を扱うものは、キャッシュするとセキュリティの問題が発生するため、キャッシュ不可能である。クライアントが Web サーバにデータを送信するとき、GET メソッドか POST メソッドのいずれかが使用される。POST メソッドでは、サーバに渡す情報が URL に表れず、URL だけではユーザを識別するのに必要な情報を得ることができないため、本研究では、GET メソッドを用いたコンテンツ及び静的コンテンツをキャッシュ対象とする。

### 3. キャッシュ更新機構の概要

本節では、キャッシュ更新を最適化するための、キャッシュ更新機構の概要について述べる。図1に更新機構での処理の流れを示す。分散Webシステムは、オリジンサーバ、管理サーバ、キャッシュサーバの3つから構成される。オリジンサーバの静的コンテンツに該当するファイルが更新されると、

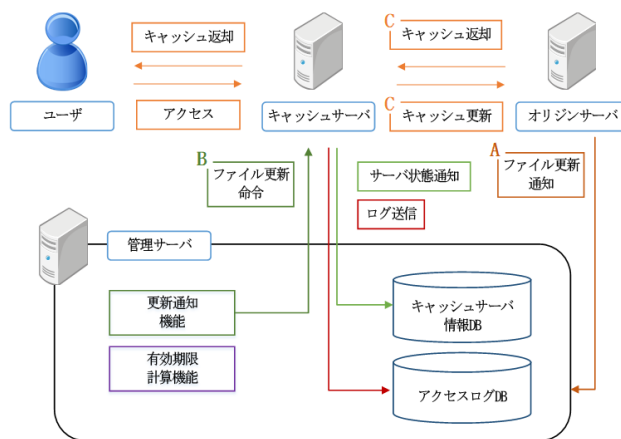


図1 更新機構概要

管理サーバへ通知される(A)。管理サーバは、通知を受け取ると、キャッシュサーバへと更新されたコンテンツをキャッシュし直すよう命令を送る(B)。この命令により、キャッシュサーバは対象のファイルを削除する。これにより次回コンテンツにアクセスされた際にキャッシュが更新されるようになる。その後オリジンサーバへコンテンツを取得するためアクセスし、取得したコンテンツをキャッシュとして保存する(C)。またそれとは別に、ユーザがキャッシュされていないコンテンツを要求した場合も、キャッシュの取得が行われる。キャッシュサーバ情報DBは、キャッシュサーバの起動停止の情報を蓄積する。アクセスログDBは、キャッシュサーバのアクセスログをもとに、クライアントのIPアドレス、アクセス時刻、URLといった情報を抽出して蓄積する。動的コンテンツのキャッシュは、未実装ではあるが、アクセスログを元にアクセス頻度を算出し、頻度に基づきアクセスの集中するコンテンツに対してキャッシュ有効期限を動的に変更する手法を検討している。現段階では、キャッシュのための基礎部分の実装のみに留まっている。

### 3. キャッシュ更新機構の評価

キャッシュ更新機構を用いて動的コンテンツのキャッシュの管理を行い、キャッシュを行わない場合とのパフォーマンスの差を調査する。図2に実験環境を示す。クライアント A 及び B では、各負荷ツールがそれぞれ 64 スレッド並列でオリジンサーバとキャッシュサーバの指定 URL へアクセスする。これにより両方のサーバが高負荷状態となる。動的コンテンツとして、入力した文字列をそのまま表示し、選択したラ

ジオボタンに対応する画像を表示する簡易な CGI を用意し、キャッシュの有効期限を 5 秒に固定し実験を行った。実験結果を図 3~5 に示す。図 3 と図 4 と照らし合わせると、キャッシュ更新機構有りの場合、キャッシュ更新のタイミングでアクセスを行うため、スループットと応答時間が繰り返し上昇していることが分かる。一方でキャッシュ更新機構無しの場合には実験開始から 20 秒ほどで 14000 ミリ秒を記録し、そのまま実験終了まで維持している。キャッシュ機構無しの場合の応答時間が跳ね上がるのは、クライアントからの 128 スレッドのアクセスとキャッシュサーバからの同様のアクセスがオリジンサーバへ向けられるためである。キャッシュ更新機構有りの場合でも、応答時間が 4000 ミリ秒を記録する瞬間的がある。これは、キャッシュサーバでキャッシュ有効期限が切れ、キャッシュ更新を行う際、一つのオリジンサーバへのアクセスでキャッシュ更新を行うのではなく、キャッシュサーバにアクセスしているクライアントの数だけキャッシュ更新のためのアクセスが発生しているためだと考えられる。キャッシュ鮮度とは、キャッシュの取得時刻とキャッシュ有効期限の差であり、低い方が良い。図 5 では、キャッシュ更新機構がある場合はキャッシュが有効期限通りに更新され、キャッシュ鮮度の劣化は一定で抑えられていることが確認できる。一方キャッシュ更新機構無しの場合には、常に最新のコンテンツを取得しているが、応答時間の分だけ遅れるため、キャッシュ更新機構有りよりも鮮度が低くなっている。以上の結果から、キャッシュ更新機構は、サーバが高負荷において、平均応答時間を改善し、スループットを大幅に上昇させることから有効に動作することが確認できた。

#### 4. 今後の課題

今回の実験は簡易な CGI を動的コンテンツとして利用した実験だったため、ブログや DB を利用するサイトなど、様々なタイプの動的コンテンツを用いて評価実験を行うことが必要である。また、キャッシュ更新機構は、サーバが休止中、もしくは通信障害により更新命令を受け取ることができなかったサーバは、キャッシュの更新を行えず、古いキャッシュが残ってしまう。解決する手段として、管理サーバで更新されたコンテンツの情報を保持し、サーバが復帰した際にまとめて更新命令を送る手法を検討している。

#### 参考文献

- [1] Akihiko Horiuchi, "Prototyping and Evaluation of Virtual Cache Server Management Function for Distributed Web System", Proc. 2015 Int. Conf. on Computational Science and Computational Intelligence pp. 324-329, Dec. 2015.
- [2] 溝渕久哲, "分散Webシステムのための優先アクセス機構の実装と評価" 香川大学, 修士論文 2016

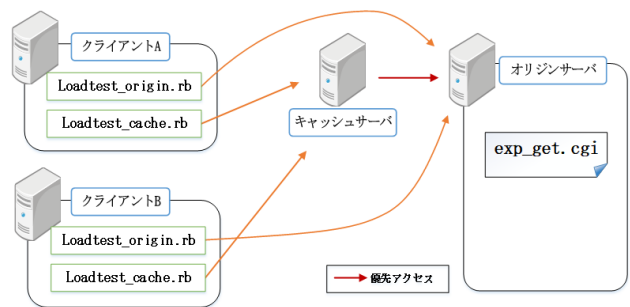


図2 実験環境

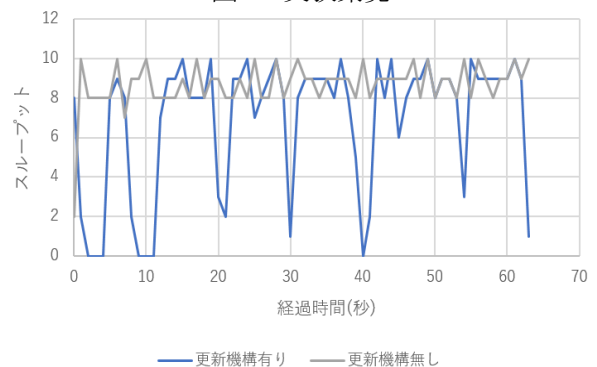


図3 スループット比較

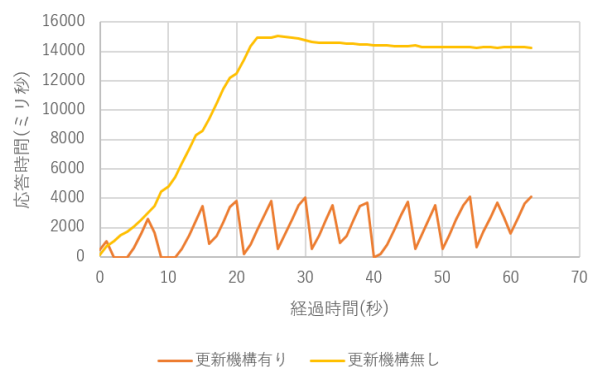


図4 応答時間比較

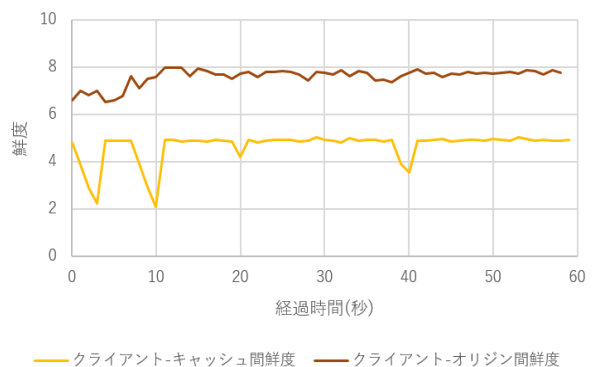


図5 キャッシュ鮮度比較