分散 Web システムのためのキャッシュ管理機構の提案

09T277 村上 拓哉 （最所研究室）

本研究では、分散 Web システムにおけるキャッシュ管理機構を提案し、キャッシュ更新アクセスのオーバヘッドについて評価する。

1 はじめに

Web サーバの高負荷問題を解決する方法の一つとして、キャッシュサーバを用いた手法がある。このキャッシュサーバを用いる手法には、Web ページへのアクセスが少ないときにキャッシュサーバが遊ぶという問題がある。この問題は、クラウドを用いてキャッシュサーバを構築し、キャッシュサーバ数を負荷量に応じて動的に増減することで解決できる。当研究室ではこの手法に基づいた分散 Web システムの開発を行っている [1]。キャッシュサーバ数が変動するため、Web サーバへのキャッシュ更新のアクセス量が一定しない。そのため、キャッシュサーバ数が極端に多くなる場合、キャッシュ更新に時間がかかり、利用者への応答も遅くなる。本研究ではキャッシュサーバ間でのトラヒックに注目し、動的コンテンツを考慮したキャッシュ機構を提案する。

2 動的コンテンツ

動的コンテンツとは、クライアントからのリクエストに応じて、部分的または全体的にコンテンツを生成する Web コンテンツのことである。クライアントからのリクエストごとに、生成される内容が変化することが多い。具体例としては検索サイトや掲示板などがある。文献 [2] では、動的コンテンツをコンテンツ変化の度合いとコンテンツ内容の鮮度で分類している。本研究では、これに経過許容時間を導入し、鮮度について再定義して動的コンテンツを分類する。ここでは、キャッシュ更新の評価に用いた許容経過時間についてのみ述べる。

許容経過時間とは、クライアントに提供する情報の劣化をどの程度許容するかという時間である。例えばオークションなどの最新情報を得る必要のある即時性をもつコンテンツは短時間で情報が著しく劣化するので経過許容時間が短く、ブログや掲示板といった最新情報でなくても充分な情報を得られるコンテンツは経過許容時間が長い。経過許容時間が短いコンテンツは、キャッシュ更新が頻繁に行われる。そのため、更新アクセスが殺到し、Web サーバが過負荷になる可能性もあるが、キャッシュを作ることで負荷が減る可能性もあり、キャッシュの可否判断が難しい。

3 キャッシュ更新の概要

キャッシュの更新モデルとして検証モデルと期限モデルを考えた。検証モデルでは、キャッシュされたコンテンツに変更がないか確認する。変更があれば、キャッシュの更新を行う。このとき、変更がなければ余分な通信が発生する。期限モデルでは、設定された期限まで Web サーバに問い合わせることなく、キャッシュを利用する。

一方キャッシュの更新タイミングとしてプリフェッチとオンデマンドの 2 通りがあり、モデルと組み合わせると以下について考慮する必要がある。期限モデルにおけるオンデマンドは、キャッシュの有効期限が切れても更新せず、ユーザのアクセスによってキャッシュの更新を開始する。検証モデルにおいてプリフェッチは使用できない (使用する意味がない)。これは検証モデルが、アクセスするたびにコンテンツが最新かどうか確認するモデルなのに対しプリフェッチが、決められた期間ごとに Web サーバにアクセスし、事前にキャッシュを更新するものだからである。

4 キャッシュ管理機構の概要

本研究で提案するキャッシュ機構を組み込む分散 Web システムは、Web サーバ、キャッシュサーバ、管理サーバ、リクエスト振り分け機構で構成される。本研究ではキャッシュサーバ上のキャッシュ管理機構を対象としている。キャッシュ管理機構を、以下の目的を満たすよう設計する。

- 動的なコンテンツをキャッシングする
- 可能な限り最新情報をユーザに提供する
- Web サーバとの通信を飽和させない

この目的を満たすため、キャッシュの更新タイミングをキャッシュサーバ数やキャッシュ自身の情報に応じて、最適になるよう変更、設定ができるようにする。Web サーバの負荷も影響することが考えられるが、当研究室ではキャッシュ更新のためのリソースを確保できる Web サーバの開発 [3] を行っており、これを用いることを前提としているので Web サーバの負荷を考慮しない。

5　キャッシュ更新シミュレータ

シミュレータを用いてキャッシュの経過許容時間や Web サーバ上でのキャッシュ更新にかかる時間がどのように影響するか調べる。シミュレータは、キャッシュサーバと Web サーバで構成されていおり、 期限モデルかつプリフェッチの場合のシミュレーションを行う。Web サーバでは、リクエストをキューに入れて到着順に処理している。キャッシュサーバは、キャッシュの有効期限が切れると更新リクエストを Web サーバに送信する。この状況をシミュレートするために、それぞれのコンテンツに対してプロセスを割当、指定された有効期限を過ぎるとそのプロセスがキャッシュの更新を行う。キャッシュ更新プロセスは、キャッシュの有効期限が切れるまで待機し、更新を繰り返す。なお、生成されたキャッシュをすべて保持するだけの容量をキャッシュサーバが持っているとする。

6　シミュレーション

経過許容時間を 100ms に固定し、Web サーバでの処理時間をパラメータとしてシミュレートする。Web サーバ上のキューでの滞留およびキャッシュ書込み時間をオーバーヘッドと呼ぶ。これにサーバでの処理時間を加えたものを応答時間となる。処理時間を 20ms から 160ms まで 20ms づつ増加させ、それぞれを順に cnt1,cnt2,...,cnt8 と呼ぶ。この実験では、cnt1〜cnt8 から一つ選び、それを 2 台のキャッシュサーバが、キャッシュするコンテンツとして扱う。そして、2 台のキャッシュサーバが 1 台の Web サーバにアクセスする。

オーバヘッドの推移を図 1 に示す。cnt1〜cnt8 は、経過許容時間と処理時間が同じ cnt5 を中心に経過許容時間よりも処理時間の短いグループと長いグループに分かれた。処理時間の短いグループ (cnt1〜5) は、ばらつきや誤差はあるものの、処理時間の違いにかかわらずオーバヘッドは 5ms 付近が平均となっており、大きな違いはみられない。処理時間の長いグループ (cnt6〜8) では、オーバヘッドが長くなった。これは、キャッシュの更新間隔以上に処理に時間がかかっているため、キャッシュの更新中に別のキャッシュの更新リクエストが発生し、処理待ちの状態が発生するためである。このような理由からオーバヘッドのデータ傾向は cnt5 を境に変化した。そこで、オーバヘッドの平均値が明確になるよう平均値をグラフ化したものを図 2 に示す。処理時間の短いグループは、図 2 においても 5ms 付近となり、大きな差がないことがわかる。しかし、処理時間の長いグループは平均時間が大きく増加している。これを見ると、処理時間と許容経過時間の同じ cnt5 から線形的に増加していることがわかる。オーバヘッドを抑えるためには、処理時間を変えることができないので経過許容時間を変える必要がある。

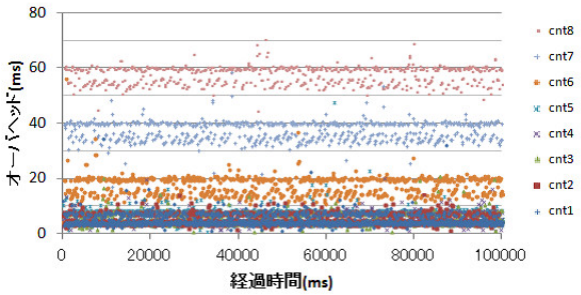

図 1: 経過時間とオーバヘッド

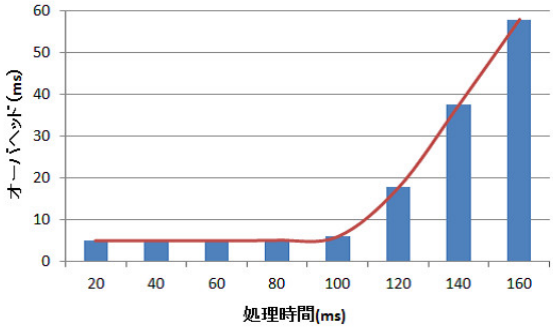

図 2: コンテンツごとの平均オーバヘッド

7　まとめ

本研究では、動的コンテンツを考慮したキャッシュ管理機構について提案し、機構における単数コンテンツによるキャッシュサーバと Web サーバ間の経過許容時間と処理時間によるオーバヘッドを調査した。課題としては、プリフェッチとオンデマンドによって与えるコンテンツの鮮度の平均値および、実際の Web サーバを想定し複数のコンテンツが混在している場合について評価が必要がある。

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論文集, 17-14, p.360, 2012.

[2] 河田光司, 最所圭三, “動的コンテンツのための動的ミラーリング機構の設計,” 平成 20 年度 電気関係学会四国支部連合大会論文集, 16-44, p.356, 2008.

[3] 山田茂和, 最所圭三, “NAP-Web の時間予測も関する評価と優先アクセス機構の設計,” 平成 24 年度 香川大学修士論文集, 2013.