5T－8

WWWにおけるhtmlテキスト単位のデータキャッシングの提案と検討

吉田 順一　相田 仁　齊藤 忠夫

東京大学工学部

1 はじめに

WWWの普及に伴うインターネットの輻輳防止及びデータ転送時間の短縮を目指したネットワーク上のキャッシュの研究が盛んに行われている。本稿では、htmlテキスト構成に即したデータ管理による体感転送速度等、各種性能の向上を目指す観点から、htmlテキスト単位のキャッシュ管理方式を提案・検討する。

2 従来のWWWキャッシング技術

2.1 研究動向と成果

WWWにおけるデータキャッシングの研究は様々な所で非常に活発に行われている。その重点は、レスポンスの高速化及びストアされたデータのより有効な活用法に置かれている。米国のHarvestプロジェクトで開発されたHarvest Cacheでは、プログラミングの工夫による高速化と、キャッシュサーバ間の階層構造を用いた連携の仕組みの提案がなされた[1]。

また、データリプレースメントについては、LRUではあまり良好な性能が得られない事が報告されており、データサイズによる優先制御や、転送時間による優先制御等の様々な手法が試されている[2]。

更に、ブラウジングの速度向上の手法として、「先読み」の研究も幾つか行われており、キャッシュサーバでの先読みによる転送速度の向上も試されている[3]。

2.2 従来手法の限界

従来のキャッシング技術では、個々のデータを互いに何の関連も持たない独立したデータとして扱われており、クライアント側でのデータの取り扱いの考慮がなされていなかった。その結果、一つのページを構成するデータ群の一部しかキャッシュ上にない場合が発生し、利用者にとって、キャッシュサーバを使う意義があまり感じられない事になる。

また、先読みについても、トラヒック量の削減という本来のキャッシュの役割に完全に逆行し、更に現状のインターネットトラヒックに余裕があまり無い事から、実際には高速化手法としては殆んど活用されていない。

A proposal of WWW caching strategy based on html-text
Jun'ichi Yoshida, Hitoshi Aida, Tadao Saito
Faculty of Engineering, The University of Tokyo

3 htmlテキスト構文解析を用いたキャッシング技術の提案

3.1 概要

図1に示す様に、WWWでは、各データがハイパーリンク(以後、リンクと表記)によって関連づけされている。htmlテキストは、リンク提供やWWWブラウザの出力(以後、ページと表記)デザインを決定する等の重要な役割を果たす。ページはhtmlテキストをベースに、画像(インラインや背景)等の付随データで構成され、ユーザーに対し直接各種サービスを提供する。

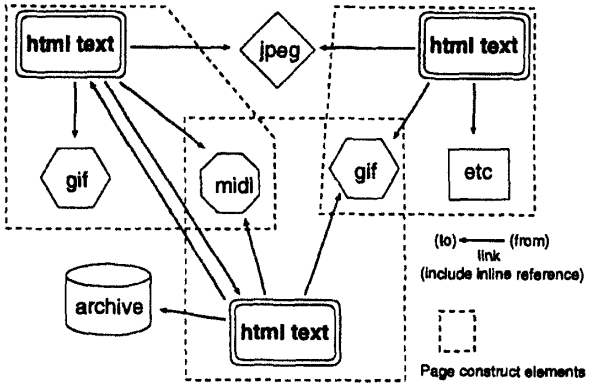

図1: WWW上の各データの相互関係

ここで、WWWサービスの基本単位である「ページ」を形成するhtmlテキストの構成を分析して利用するキャッシュ管理手法を提案する。

WWWキャッシュサーバに単純な構文解析機能を搭載する。そして、クライアントからの要求に応じてキャッシュサーバがHTTPサーバから受信したデータがhtmlテキストであった場合には構文解析を行い、ページの部品(インラインイメージ等)やhtmlテキスト自身(表示文字数等)に関する情報を抽出する。これをページ構成部品個々のMIMEヘッダ等の情報等と統合して使用する事により、一つの「ページ」を構成するデータ群を一括してキャッシュ上で管理を行う事を考案した。

つまり、図1における破線で囲まれたデータ群一つ一つをキャッシュの中でのデータの管理単位としてリプレースメント及び連携サーバへの問い合わせを行うという事になる。

本方式の利点の一つとして、ヒットしたページについては、確実に高速なレスポンスを保証することを可能にする点を挙げる事ができる。

その他にも、次節に示す様なページ単位での転送時間及び転送データ量を考慮したリプレースメント対象選択や、ページの部分的な削減方法等の実装が可能となる。後者については、移動体通信等の狭帯域通信環境下におけるWWWサービスを快適にする為の手法としての応用も期待できる。

また、この他の利点として連携サーバ間の通信回数の削減による高速化が追求可能になるという点を挙げる事ができる。

3.2 リプレースメント対象ページの選定

htmlテキストから複数のデータが呼び出されると、その回数だけサーバに接続してデータを転送してもらう分だけ転送にかかる時間が長くなる。HTTP1.1では一回の接続で複数のデータ転送を行う仕様を備えているが、通信速度が遅い場所に対するアクセスである場合には、やはり転送時間が長くなるので、出来るだけキャッシュ上に残すような仕様にする事が望まれる。

各クライアントから合計 N_{ref} 回参照されたページが n 個の構成要素で構成されているとすると、各構成要素のサイズを $s_i(1 \leq i \leq n)$、転送にかかった時間を t_i、とした場合に、リプレースメント評価関数 Rep の例としては

$$Rep = N_{ref} \times \sum_{i=1}^{n} \frac{t_i^{\alpha}}{s_i^{\beta}} \qquad (1)$$

といったものが考えられる。(α 及び β は定数)。評価関数 Rep の値が小さいページから削除する事で、転送時間が長く構成データ数が多いページを優先的にキャッシュに残す事が可能になる。この評価関数では、サイズが大きいデータが削除され易くなるが、これは転送時間の長期化や輻輳の原因になる恐れがある。

そこで、定数 β の値を1以下に、定数 α の値を1以上に $(0 < \beta < 1 < \alpha)$ 設定すると、データサイズが大きい場合でも、評価関数に占める転送時間の項が大きくなり、キャッシュ上に残存しやすくなると考えられる。また、β を負の値にする事により、ファイルサイズの大きいデータを優先して保存する事が可能となる。

3.3 ページの部分的削除アルゴリズム

前節で述べた手法に従い、選定されたページを削除する場合に、現行のリプレースメント手法では、削除対象データを個々に完全に削除する事になる。しかし、htmlテキストによる「ページ」構成が複数画面に及ぶ場合には、ページの先頭のみを残して残りを削除する手法(図2)が考えられる。これは、ページをまるごと削除するのではなく、先頭部分やリンクが集中している部位等に相当するデータだけを保存して、他の部分を削除する方式である。クライアントからの該当ページの転送要求が生じた場合に、元データの転送をサーバに要求し、キャッシュ上に残存する物との同一性が確認された場合には、キャッシュに残存するデータをまずクライアントに転送し、後に消去された部位の転送を行う。これにより、クライアントに対するレスポンス速度の向上が期待出来る。

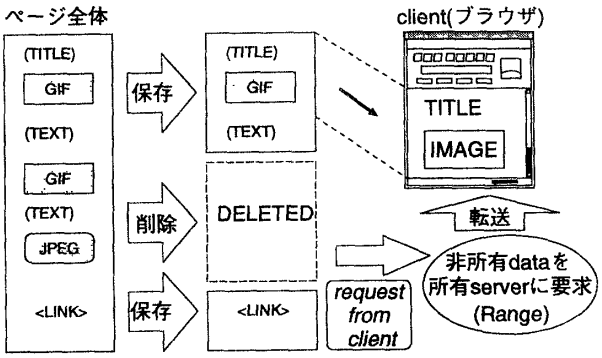

図 2: 「ページ」部分削除の仕組み

実際には、ページを構成するhtmlテキストに含まれる文字数(行数)及び個々のインライン画像の大きさ及び配置情報を基に、該当するページがブラウザ上で何画面分の大きさになるのかを推定しておき、最初の一画面分に相当する場所の配置される画像等についてのみ保存し、残りの部位を削除するという手法が考えられる。また、保存する場所をページの先頭ではなく、リンクが集中している所にするという手法も考えられる。HTTP1.1では、Rangeヘッダフィールドにより、削除した部位のみの転送を要求する事が可能となる。

4 今後の方針

今後の研究の方針としては、今回提案・検討した方式をキャッシュサーバに実装して既存のキャッシュサーバとの性能の違いを多角的に評価する。また、キャッシュサーバ間の連携動作に関する通信負荷の評価を行う予定である。

参考文献

[1] Anawat Chankhunthod,et al.,"A Hierarchical Internet Object Cache", ftp://ftp.cs.colorado.edu/pub/cs/techreports/schwartz/HarvestCache.ps.Z

[2] Roland P.Wooster and Marc Abrams,"Proxy Caching That Estimates Page Load Delays",6th International World Wide Web Conference,April 7-11,1997,http://www6.nttlabs.com/HyperNews/get/PAPER250.html

[3] Masaaki NABESHIMA,"The Japan Cache Project:An Experiment on Domain Cache",6th International World Wide Web Conference,April 7-11,1997,http://www6.nttlabs.com/HyperNews/get/PAPER21.html