

# トラヒック適応型ネットワーク資源最適配置法の提案\*

5 T-6

藤浦 豊徳†

内藤 昭三‡

NTT ソフトウェア研究所

1. まえがき近年のインターネットブームに伴い、インターネットを利用するデータは急激に増加している。そのため、インターネットのトラヒックが集中するインターネットエクスチェンジにおいては、そのトラヒックを処理するために、高速な専用線を用意したり、高速なルータを用いることが求められている。しかし、これらは非常にコストがかかるので、際限無く実施することは困難である。そこで、ここでは、コストを削減するために、トラヒック量を削減することを考え、その方法について提案する。

また、アプリケーションとして、World Wide Web (WWW) の利用が急速に広がっている。そのため、トラヒック全体に占める WWW のトラヒックの割合が非常に増加している。これは、実時間性が妨げられている原因の一つになっていると考えられる。そのため、このトラヒック量を削減する方法について検討する。また、ユーザの増加が、WWW サーバの負荷の増大の原因の一つになっている。そこで、WWW サーバの負荷を減少させる方法についても検討する。

## 2. 提案方式

2.1. サーバキャッシュ法の提案  
トラヒックが集中する場所において、トラヒック量を減少させるための方法として、ここでは、ユーザとサーバの間にトラヒックを束ねるものを設置し、ユーザのトラヒックが直接ボトルネック（図1）となっているネットワークを通過しないようにすることを考える。このときトラヒックを束ねるものとしてここではサーバ代理サーバを用いることを考える。

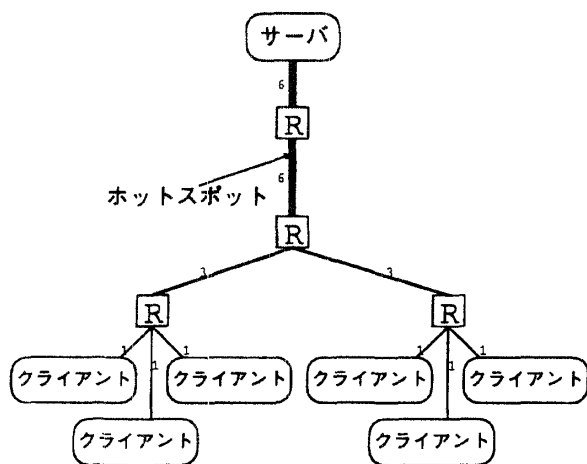


図1: ボトルネック

サーバ代理サーバはサーバに対する要求を、サーバに代わって処理する装置である。クライアントからの要求は、蓄積されたデータをもと全てサーバ代理サーバに行われる。サー

バ代理サーバは、従来のミラーサーバと異なり、要求のうち、頻度の少ないものは、サーバのデータを蓄積するかわりに、サーバ代理サーバがクライアントに替わってサーバをアクセスする。つまり、クライアントからのアクセス要求のうちの一部だけをサーバに要求するため、トラヒック量を削減することができる。また、データアクセス時の応答時間も短くなる。

2.2. サーバ選択法  
また、サーバの負荷を減少させるための方法として、同じ内容の情報を複数のサーバにおいておき、ユーザがネットワーク的に近いサーバの一つを選択し利用する方式 [5] や、DNS のラウンドロビン機能を用いて、ネットワーク距離を無視して選択する方式 [6] が用いられている。ここで、前者の場合には、ユーザが明示的にサーバを選択しなければならないため、使いやすさの点で問題がある。また、後者の場合はクライアント—サーバ間のネットワーク距離を無視しているため、ネットワークの負荷を増加させる欠点がある。そこで、ここでは、サーバの選択をクライアント代理サーバで実現する方式について提案する。この方式は、サーバ代理サーバとユーザの間にクライアント代理サーバを設置し、クライアント代理サーバが、最も近いサーバ代理サーバを自動的に選択し、利用するというものである。このようにすることによって、ユーザは代理サーバの所在地を知らなくても、自動的に最も近いサーバ代理サーバを使うことができる。

2.3. 投機的キャッシュ法  
従来の研究においては、WWW のデータのキャッシュを行う場合、一時記憶内のデータの削除方式として LRU 方式（図2左）が用いられてきた。今回は、LRU 法のようなオンラインアルゴリズムではなく、過去のデータのアクセス履歴をもとにした、投機的キャッシュ法を提案する（図2右）。投機的キャッシュ法は、従来のアルゴリズムのように、キャッシュの対象が時間とともに変化していく方式とは異なり、一定の単位時間毎（例えば一日）にキャッシュの対象を変化させていく。このようにすることによって、同じデータが何度もサーバから取り出されることを防ぐことができる。LRU 法の場合、キャッシュの内容がいっぱいになった場合、直近のアクセスが最も古いものを削除することにより、キャッシュの効率を向上しようとしている。

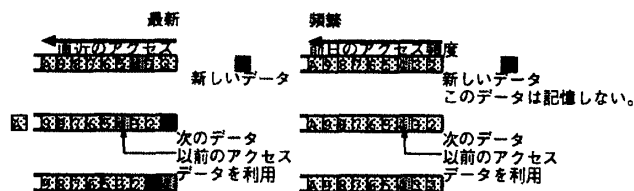


図2: LRU(左)と投機的キャッシュ法(右)

今回の提案方式の場合には、キャッシュの内容がいっぱいにならないように、あらかじめ、キャッシュすべきデータを一単位時間前のアクセス履歴をもとにして決定しておく。例えば、昨日のデータのうち、頻繁にアクセスのあったものについては本日のキャッシュの対象と決め、一日そのデータを保持しておくことにする。

\*A Traffic Adaptive Method for Network Resources Allocation

†Toyonori FUJIURA, NTT Software Laboratories.

‡Shozo NAITO, NTT Software Laboratories.

2.4. 内容の更新法従来の場合、サーバのデータが書きかわった場合についても、代理サーバはそれを検出できず、ユーザに古いデータを提供し続けていた。そこで、本研究では、サーバのデータが書きかわった事をサーバ代理サーバに通知するシステムについて提案する(図3)。

これは、サーバにおいて、ファイルを常に監視する。もし、書き換え等の変化が起こった場合は、マルチキャストを用いて内容の変化したURLをサーバ代理サーバに通知する。サーバ代理サーバはそのURLが必要であれば、サーバからその内容を得る。このようにすることにより、従来のキャッシュのデメリットであった古い情報を提供してしまう可能性があることを防ぐことができる。

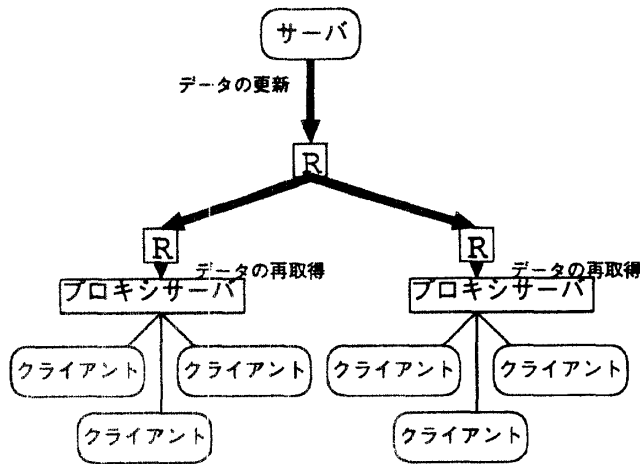


図3:内容の更新法

3. 性能評価今回は、提案方式のうち投機的キャッシュ方式の有効性を示す実験を行った。

3.1. 評価環境アルゴリズムの評価を行うために、NTT Home Page (www.ntt.co.jp) のアクセス履歴を用いた。

3.2. キャッシュ置換アルゴリズムここでは、キャッシュ置換アルゴリズムとして前述の投機的キャッシュ方式と従来からCPUキャッシュ等で用いられているアルゴリズムであるLRUを使用した場合との比較を行い、提案アルゴリズムの有効性を示す。

はじめに、代理サーバにおけるキャッシングの方式として、ファイルのバイト数にかかわらず、アクセス頻度が高いものを代理サーバにおくことが望ましいことが、実験から明らかになった。しかし、現実として、アクセス頻度を求めた日と、サーバが動いている日を同一にすることは困難である。なぜなら、サーバが動いている時点でアクセス頻度は確定しないからである。そこでここでは、アクセス頻度を求める日をサーバが動いている日の前日、前前日、七日前とした。その時の、トラヒック削減の効果を従来の方式と比較し評価する。また、アクセス頻度を一日前のデータ全てを元にして求める方式と、一日前のデータのうち、一時間単位で同じ時刻のもの一時間分を元にして求める方式の二方式について検討した。

3.3. 前提サーバを動かしている日に対して、アクセス頻度を学習する日を、その前日、二日前、一週間前とした。一日は0:00-24:00の24時間とした。また、キャッシュは空から開始した。サーバを動かしている日に転送された全データ量は、約3.2GBytes/dayである。これは、平均約37KBytes/secに相当する。また、ディスク上に存在するファイルの総量は、リンクが張られていないものを含めて約420MBytes、一日のうち、実際にアクセスのあったファイルのサイズの総量は約27MBytesである。

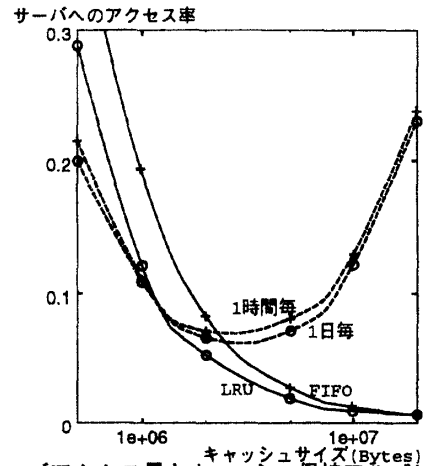


図4:サーバアクセス量とキャッシュ保持アルゴリズムの関係(従来方式との比較)

4. 結果図4は、投機的キャッシュ方式において、キャッシュするデータの選定を、前日24時間のアクセス履歴をもとに行う場合と、前日の同一時間帯一時間のアクセス履歴を元に行う場合について、示したものである。この図より、前日24時間のアクセス履歴を元にした方が良いことが分かる。

このことから、キャッシュサイズが小さい場合は、あらかじめキャッシュすべきデータを決めておいたほうがリアルタイム方式より効率が良いことが分かる。また、一時間毎の履歴を用いるより、一日毎の履歴を用いた方が良い事が分かる。さらに、決める時は、前日のアクセス履歴をもとにして決めるのが良い事も他のデータから明らかになっている。

5. むすび本方式では、キャッシュのページングによるメモリ管理の手間を省くことができるため、キャッシュ容量が少ない場合について、本方式が有効であることがわかった。

現在、キャッシュ機能を持ったクライアント代理サーバとしては、squid[1]など多くの方法が提案、実装されている。また、ドメインキャッシュ[2] (米国内に、jpドメインのサーバの内容をキャッシュする代理サーバを設置する方法)が提案され、実装されている。さらに、投機的キャッシュと類似の方法として、Staticキャッシュ[3]の考え方も提案されている。

今後の課題としては、これらの関連研究も検討しながら、投機的キャッシュ方式と、LRU方式を混合させる方式への拡張、性能評価や、アクセス確率が、統計情報からの推測値とどの程度離れても有効に機能するかを評価すること(感度分析)、また、それに基づく、キャッシュセットの更新時期の決定方法を明らかにしていくことなどが考えられる。

#### 参考文献

- [1] "Squid Internet Object Cache". <http://squid.nlanr.net/Squid/>.
- [2] Masaaki NABESHIMA. "Japan Cache Project: An experiment on Domain Cache". Sixth International World Wide Web Conference, April 1997. <http://proceedings.www6conf.org/HyperNews/get/PAPER21.html>.
- [3] Alex Rousskov, Valery Soloviev, and Igor Tatari-nov. "Static Caching". NLANR Web Cache Workshop, June 1997. <http://www.nlanr.net/Cache/Workshop97/Papers/Rousskov/rousskov.html>.
- [4] 藤浦豊徳, 内藤昭三. "アクセス履歴を用いたWWWサーバキャッシュ法の提案と評価". 信学会総合大会, March 1997.
- [5] <http://www.asahi.com/> など.
- [6] <http://www.ntt.co.jp/>.