

## WWW キャッシュにおける隣接サーバ情報の設定支援機構

坂本 岳史† 井上 博之† 山口 英† 尾家 祐二‡

†奈良先端科学技術大学院大学

‡九州工業大学 / 奈良先端科学技術大学院大学

### 概要

World Wide Web におけるネットワークの過度の利用や、アクセスの集中によるサーバへの過負荷、それに伴う利用者から見た応答時間の遅延の問題に対処するために、分散 WWW キャッシュシステムが導入されている。現状の方式では、参照の対象となるキャッシュサーバ群の選択等の設定は各キャッシュサーバの管理者が経験をもとに手動で行っている。また、ネットワークや他組織のキャッシュサーバの状態の変化に動的に対応できず、最適な設定を保つことは困難である。本稿では、ヒントサーバを導入することにより、分散環境の設定・管理の自動化・簡略化を支援し、またキャッシュサーバ間の問合わせパケットの削減と、アクセス応答時間の短縮を図る方式を提案する。

## Automatic Configuration Mechanism of the Information of Neighbor Servers for WWW Cache System

Takeshi Sakamoto† Hiroyuki Inoue† Suguru Yamaguchi† Yuji Oie‡

†Nara Institute of Science and Technology

‡Kyushu Institute of Technology / Nara Institute of Science and Technology

### Abstract

The distributed cache system for WWW infrastructure has been introduced in order to reduce the network traffic and the server load, which will, in turn, contribute to the improvement in the response time for the users. In this kind of systems, administrators of cache servers have to determine by themselves how and which neighboring cache servers should be incorporated, based upon their knowledge. Therefore, it is very difficult for them to update the related management in response to change of the state of network and/or the neighboring cache servers, in a dynamic and timely manner. For this reason, in this paper, we propose the *hint server* which will automatically update the information associated with neighboring cache servers in a way to reduce both the number of packets exchanged among cache servers and the response time for the users.

# 1 はじめに

現在では World Wide Web (WWW) に代表される広域ネットワーク上での情報サービスは急速に発展を遂げている。その結果、ネットワークの過度の利用や、アクセスの集中によるサーバへの過負荷、それらに伴う応答時間の遅延が深刻な問題となっている。

これらの問題を解決するために、WWW キャッシングプロキシサーバ (キャッシュサーバ) が広く利用されている。キャッシュサーバはプロキシサーバにキャッシュ機能を持たせたもので、クライアントが要求するオブジェクトをキャッシュ内に保有している場合にはそのキャッシュから必要なオブジェクトを取得し、クライアントに転送する。必要なオブジェクトがキャッシュにない場合には、オブジェクト本来の WWW サーバ (オリジンサーバ) からオブジェクトを取得し、クライアントに転送するとともに自身のキャッシュにも格納する。これにより同一オブジェクトの再利用が実現され、ネットワークの利用の削減と平均応答時間の短縮につながる。

また各組織が持っているキャッシュサーバを連携させ、全体として1つの大きなキャッシュサーバとみなす分散キャッシュシステムが提案されている。このシステムでは、クライアントから要求を受けたキャッシュサーバが必要なオブジェクトを保存していない場合、他のキャッシュサーバにそのオブジェクトの有無を問合わせ、もしあればそのキャッシュサーバからオブジェクトを取得してクライアントに転送することにより、全体としてキャッシュの効率を向上させることができる。この方式では各組織のキャッシュサーバの管理者は、ネットワーク構成情報や他組織のキャッシュサーバの位置情報をもとに、参照の対象となるキャッシュサーバの選択等を手動で静的に設定しているのが現状である。また、ネットワークの状態や他組織のキャッシュサーバの変化に動的に対応できないため、常に管理者に負担がかかるだけでなく、最適な設定を行うことが困難という欠点がある。

本稿では、ネットワークや各組織のキャッシュサーバの状態、また各キャッシュの内容等を把握するヒントサーバを置くことにより、分散環境の設定・管理を自動簡略化し、またアクセス応答時間の短縮を行う手法を提案する。クライアントからの要求を受けたキャッシュサーバは、ヒントサーバに必要なオ

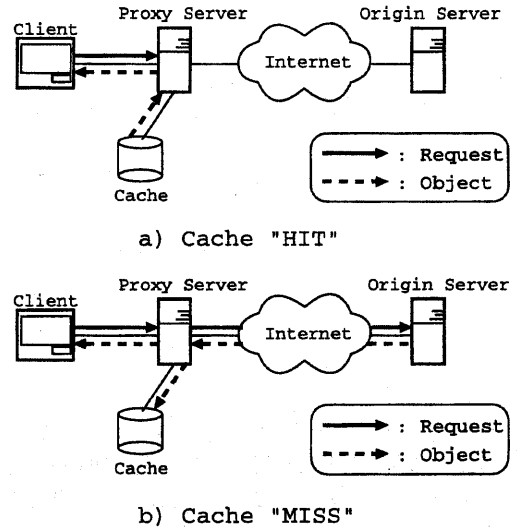


図 1: caching proxy server

ブジェクトのありかを尋ねる。ヒントサーバは自身を持つデータベースから必要なオブジェクトを保持しているキャッシュサーバを見出し、要求を出してきたキャッシュサーバにヒントとして与える。ヒントサーバはネットワークやキャッシュサーバの状態の変化に対応することにより、各組織のキャッシュサーバの設定の自動化が実現できる。またキャッシュサーバは通信相手をヒントサーバに特定できるので、通信における待ち時間を最適化することが可能となり、結果としてクライアントからみた応答時間を短縮することができる。

## 2 背景

### 2.1 キャッシュサーバ

多くの組織では組織内のセキュリティの向上を図るためにファイアウォールを利用している場合が多い。ファイアウォールを越えてインターネットにアクセスするために、データを中継するプロキシサーバが設置されている。このプロキシサーバにキャッシュ機能を持たせることにより (図 1)、クライアントから見た応答時間の短縮を図ることができる。クライアントの要求するオブジェクトがプロキシサーバのキャッシュ内に存在する場合、サーバはキャッシュからオブジェクトをクライアントに返送する。

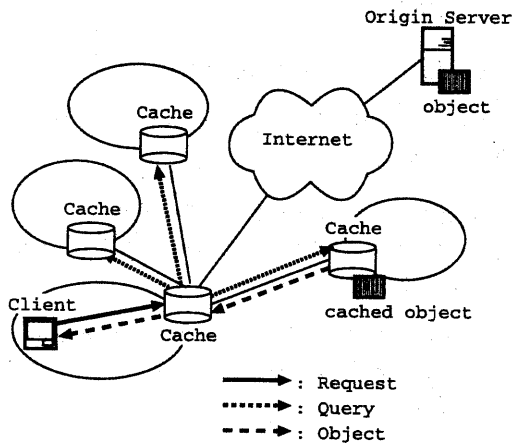


図 2: Distributed Cache System

キャッシュ内に必要なオブジェクトがない場合は、オリジンサーバからオブジェクトを取得し、クライアントに中継するとともにキャッシュ内に保存しておく。このように同一内容の情報の再利用を実現することによって、応答時間の長い WWW サーバへのアクセスを省略することができ、トラフィックの削減とクライアントから見た応答時間の短縮が可能となる。

キャッシュサーバの代表的なシステムとして CERN-httpd [1] や Harvest [2]、またその他多くの商用システムが存在する。

## 2.2 分散キャッシュシステム

Harvest や Squid [3]、あるいは R.Malpani [4] のシステムでは、複数のキャッシュを階層的あるいはメッシュ状に配置し、全体として1つの大きなキャッシュサーバを構成することができる(図2)。

この方式では、クライアントからの要求を受け取ったキャッシュサーバが要求されたオブジェクトを持っていないとき、直接オリジンサーバからオブジェクトを取得するのではなく、他のキャッシュサーバにオブジェクトの有無を問い合わせる。問い合わせを受けたキャッシュサーバはオブジェクトを持っているかどうかを返答する。問い合わせ側のキャッシュサーバはもっとも早く「ヒット(オブジェクトを持っている)」を返してきたキャッシュサーバからオブジェクトを取得する。全てのキャッシュサーバが「ミス(オ

ブジェクトを持っていない)」を返してきた、あるいは待ち時間内に応答がない場合にはオリジンサーバからオブジェクトを取得する。

## 2.3 現状システムの問題点

分散キャッシュシステムでは、クライアントから要求を受けたキャッシュサーバが必要なオブジェクトを保持していない場合、あらかじめ設定されている他の全てのキャッシュサーバに問い合わせを行い返事を待つ。問い合わせに対し設定時間内にヒットの返事を返すキャッシュサーバが1つもなければタイムアウトし、オリジンサーバからオブジェクトを取得する。現状では他のキャッシュサーバを参照するための設定は、各キャッシュサーバの管理者が経験をもとに手動で行っている。しかしながら、ネットワークや他のキャッシュサーバの状態の変化に動的に対応しようとするとき常にキャッシュサーバの管理者に負担がかかるうえ、最適な状態を保っておくことは困難である。

## 3 ヒントサーバ

### 3.1 ヒントサーバによる解決

先に述べた問題点を解決するためにヒントサーバを導入する。ヒントサーバは各キャッシュサーバが持っているオブジェクトを把握することにより、キャッシュサーバからの要求に対し必要なオブジェクトを持っているキャッシュサーバを選択して応答する。各キャッシュサーバはヒントサーバの存在のみを知っていればよく、他組織のキャッシュサーバに関する設定をする必要がなくなり、またネットワークや他のキャッシュサーバの状態の変化を意識することが不要となることから、他のキャッシュサーバを参照するための設定を自動化・簡略化することができる。

### 3.2 ヒントサーバの概要

ヒントサーバは各キャッシュサーバのキャッシュの内容を把握している。ヒントサーバはキャッシュサーバからの質問に対して適切なキャッシュサーバを選択し応答する。応答の内容は、オブジェクトを持っていると思われるキャッシュサーバのアドレス、あ

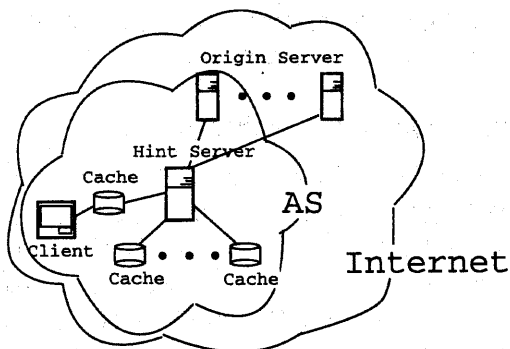


図 3: Hint Server

るいはどのキャッシュサーバもオブジェクトを持っていないという情報のいずれかである。

ヒントサーバとキャッシュサーバの間の通信が高速に行われる必要があること、異なるポリシーで運用されている他組織のキャッシュサーバの設定を変更するのは困難であることを考慮し、今回ではヒントサーバは AS (Autonomous System) 内の数十個程度のキャッシュサーバに対してサービスを行うものとする (図 3)。

### 3.3 ヒントサーバに対応するキャッシュサーバ

キャッシュサーバはヒントサーバの応答をもとに、オブジェクトを持っているキャッシュサーバ、あるいはオリジンサーバにオブジェクトを取りに行く。キャッシュサーバは自分のキャッシュの内容を更新した場合、そのことをヒントサーバに通知する。これにより、各キャッシュの内容とヒントサーバが持つデータベースとの整合性を保つことができるようになる。

ヒントサーバからの応答がない場合は、キャッシュサーバは適当な待ち時間を設定し、応答がタイムアウトするまで待つ。キャッシュサーバは応答がタイムアウトしたときはオリジンサーバにアクセスする。この待ち時間の設定は、過去のヒントサーバからの応答時間の履歴などから適切な値を設定することができる。

## 4 設計

### 4.1 ヒントサーバの持つ情報

ヒントサーバはデータベースに次のような情報を持っている。

- 各キャッシュサーバが持っているオブジェクトの URL のリスト (キャッシュに書き込まれた時刻等も持つ)。
- そのオブジェクトに関する情報 (最終更新時刻、ファイルのサイズ、キャッシュ内の有効期限等)。
- サービスを行うキャッシュサーバのアドレスのリスト (キャッシュサーバが利用可能かどうかという情報も含む)。

### 4.2 ヒントサーバ～キャッシュサーバ間の処理手順

ヒントサーバとキャッシュサーバとの間のプロトコルは ICP (Internet Cache Protocol; RFC2186) [5] を拡張して用いる。具体的には新規の ICP Opcode を追加し、オブジェクトの更新時刻等の情報を載せるようにする。

ヒントサーバとキャッシュサーバとの間の処理手順を以下に示す。

1. クライアントから要求を受けたキャッシュサーバは、欲しいオブジェクトの URL を載せた ICP 問い合わせパケットをヒントサーバに送る。
2. ヒントサーバは自身のデータベースからどのキャッシュサーバにオブジェクトが存在するかを調べる。オブジェクトが存在する場合はそのキャッシュサーバのアドレス、キャッシュの更新時刻、オブジェクトの更新時刻、オブジェクトのサイズなどを ICP 応答パケットに載せて返送する。複数のキャッシュサーバに同一のオブジェクトが存在する場合には、適切なキャッシュサーバを 1 つ選択する。この方法については、たとえば更新時刻の情報に基づく方法など種々考えられるが、この点については今後検討を行う。オブジェクトが存在しない場合は「ミス」を ICP 応答パケットに載せて返送する。

3. キャッシュサーバはヒントサーバの応答の内容に従って、オブジェクトを持っているキャッシュサーバ、あるいはオリジンサーバからオブジェクトを取得する。
4. キャッシュサーバは取得したオブジェクトをクライアントに転送し、キャッシュの内容を更新した場合にはそのことをヒントサーバに通知する。
5. ヒントサーバはキャッシュサーバからキャッシュの内容の変更通知を受け取ると、データベースを更新する。

### 4.3 キャッシュサーバの改造

ICP の追加 Opcode に対応するため、現在インターネットで広く利用されている代表的なキャッシュプログラムである Squid をもとに改造を行う。ヒントサーバへの問い合わせの処理、ヒントサーバからのヒントを解釈し他のキャッシュサーバまたはオリジンサーバからオブジェクトを取得する処理、ヒントサーバにキャッシュの内容の変更を通知する処理を追加する。

## 5 期待される効果

### 5.1 設定の自動化による効果

ヒントサーバを設置することにより、キャッシュサーバの通信相手はヒントサーバのみとなる。各キャッシュサーバは、ヒントサーバに関する情報を知っているだけよく、他のキャッシュサーバに関する情報を持つ必要がなくなり、設定を簡略化することができる。

本学の属する AS である WIDE [6] を例に見ると、バックボーンの大規模なネットワークの変更が過去 1 年間に 5 回も起こっている。WIDE 内のキャッシュサーバの変更は 3 週間に 1 回程度の割合で起っている。さらに局所的なネットワークやキャッシュサーバの状態の変化はより頻繁に発生している。ヒントサーバを導入すればこれらの変化に自動的に対応できるため、管理者への負担を減らすことができる。

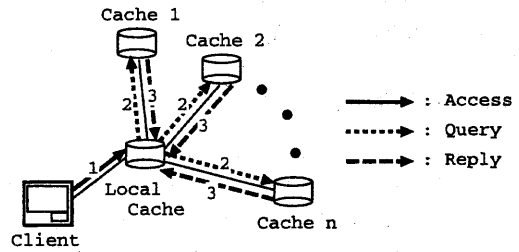


図 4: 従来法における ICP パケット

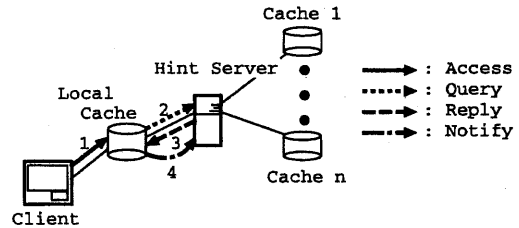


図 5: ヒントサーバをおいたときの ICP パケット

### 5.2 ICP 問い合わせパケットの削減

クライアントからの要求の回数を  $A$ 、ローカルのキャッシュのヒット率を  $h$ 、問い合わせの対象となるキャッシュサーバの個数を  $n$  とする。

このとき、ローカルのキャッシュにヒットせずに他のキャッシュサーバあるいはヒントサーバに問い合わせる回数の期待値は  $A \times (1 - h)$  で表される。これを  $\alpha$  とすると、問い合わせ・応答パケットの個数の期待値は以下ようになる。

- 従来法 (図 4)

- unicast で問い合わせる場合  
(問い合わせ  $n$ 、応答  $n$ )  
 $A \times (1 - h) \times (n + n) = 2n\alpha$

- multicast で問い合わせる場合  
(問い合わせ 1、応答  $n$ )  
 $A \times (1 - h) \times (1 + n) = (n + 1)\alpha$

- 今回の方式 (図 5)

- $A \times (1 - h) \times 3 = 3\alpha$   
(ヒントサーバへの通知パケットを含む)

これらの式より今回の方式では、問合わせ・応答パケットの総数はおよそ  $1/n$  に減っている。

### 5.3 応答時間の短縮

従来の分散キャッシュシステムでは、どのキャッシュサーバも必要なオブジェクトを持っていないということを確認するには、全てのキャッシュサーバからのミスの応答を受けるか、あるいは応答がタイムアウトするまで待たなければならない。この応答待ち時間の値として一般には1~2秒が多く用いられている。今回の方式では、キャッシュサーバはヒントサーバのみと通信を行えばよいから、応答待ち時間を最適化することができる。

WIDEのバックボーンから末端のネットワークまでの応答時間は約100m秒以下に収まっており、ヒントサーバだけを相手にすることで応答待ち時間をかなり小さくすることが可能である。

## 6 今後の課題と予定

ヒントサーバは、各キャッシュサーバからキャッシュ内容の変更通知を受けとることにより、ヒントサーバのデータベースとキャッシュの内容の整合性を取る。しかしながら、変更通知が届かないなどの障害が起これば、この整合性が取れなくなるとヒントサーバはキャッシュサーバに誤った情報を渡してしまうことになるが、これがどの程度ヒット率や応答時間に影響を及ぼすかを確認する必要がある。

今後は、それぞれのキャッシュサーバの間のネットワークの状態や、ルーティング情報、その他の情報をもとに、ヒントサーバが最適と思われるキャッシュサーバを選択した場合の評価を行う。

ヒントサーバは全てのキャッシュサーバと高速に通信を行う必要があるため、ヒントサーバを配置する場所を考慮しなければならない。今回はAS内での運用を仮定したが、ヒントサーバをIX (Internet eXchange point) 等のAS間の接続点に置き、運用範囲を複数のAS間に拡張した場合の評価を行う。

## 7 まとめ

本稿では、分散WWWキャッシュシステムの設定の自動化の支援機構を実現するために、ヒントサーバを設置したヒントサーバ付き分散キャッシュ方式

を提案した。ネットワークやキャッシュサーバの状態、また各キャッシュの内容等を把握するヒントサーバを導入することにより、従来の分散WWWキャッシュシステムでは困難であった分散環境の設定・管理の自動化を支援する。各キャッシュサーバはヒントサーバのみと通信を行うことにより、ICP問合わせ・応答パケットの削減、応答待ち時間の短縮が可能であると期待される。

## 謝辞

本研究を行うにあたり、多大なご協力を頂いた奈良先端科学技術大学院大学情報ネットワーク講座の諸氏、および貴重な助言を頂いたWIDE Projectのメンバに感謝します。

## 参考文献

- [1] Ali Luotonen, Henrik Frystyk Nielsen and Tim Berners-Lee: "CERN httpd" (1995). (<http://www.w3.org/Daemon/Status.html>).
- [2] A. Chankhunthod, P.B. Dangiz, C. Neerdaels, M.F. Schwartz and K.J. Worrell: "A Hierarchical Internet Object Cache", Technical Report 95-611, Computer Science Department, University of Southern California, Los Angeles, California (1995). (<ftp://ftp.cs.colorado.edu/pub/cs/techreports/schwartz/HarvestCache.ps.Z>).
- [3] D. Wessls: "Squid Internet Object Cache". (<http://squid.nlanr.net/>).
- [4] R. Malpani, J. Lorch and D. Berger: "Making World Wide Web Caching Servers Cooperate", *Fourth International WWW Conference* (1995). (<http://http.w3.org/pub/Conferences/WWW4/Papers/59/>).
- [5] D. Wessls and K. Claffy: "RFC2186, Internet Cache Protocol (ICP), version2". (<http://ds.internic.net/rfc/rfc2186.txt>).
- [6] WIDE Project: "WIDE (Widely Integrated Distributed Environments)". (<http://www.wide.ad.jp/>).