

# インターネットにおける ドメインキャッシュサーバの導入

高品 智一      水野 忠則

静岡大学

インターネットを使用した広域な情報提供サービスの増大により、1つのサーバに対して多くのクライアントが同時にアクセスするようになり、1つのクライアントに対する応答時間が長くなる傾向にある。

また、Proxyサーバなどではキャッシュにあるデータが常に最新であるとは限らないといった利用者側の問題や、データアクセスに対する課金ができないなどの情報提供者側の問題がある。

本論文では、クライアントがサーバからとってきた情報を記憶するドメインキャッシュサーバ(DCS)を提案する。このサーバを導入したシステムにおいてクライアント-サーバ間の通信プロトコル(HTTP)を部分的に拡張した。これによって、サーバやインターネット全体の負荷を減らし、サービスを向上させることを目的としている。

## Introduce of Domain Cache Server in Internet

Tomokazu Takashina    Tadanori Mizuno

Faculty of Information, Shizuoka University

3-5-1, Johoku, Hamamatsu, 432 Japan

An amount of Internet DB access has been increasing. Because one server is concurrently accessed by many clients in such Internet DB access, response time by client request is needed long time. There are many problems, for example, cached data may not necessarily be latest in Proxy server system, and how to account for DB access by Internet Provider.

This paper describes the Domain Cache Server(DCS) system that keeps information which client gets from server. DCS reduces workload of server and whole Internet.

## 1 はじめに

近年、インターネットを使用した広域な情報提供サービスが行われてきている。それは、点在するサーバに情報を持たせ、多数存在するクライアントからアクセスできるようにしたものである。しかし、1つのサーバに対して多くのクライアントが同時にアクセスするため、1つのクライアントに対する応答時間が長くなる傾向にある。また、サーバが故障したときには、そのサーバが持つ情報を得ることができなくなる場合がある。

我々はインターネットを実際に使用してみて、ネットワークからデータを取り出すために多大な時間がかかるといった利用者側の不便さ、そしてProxyサーバなどのキャッシュにあるデータに対する課金ができないなどの情報提供者側の問題を実感した。本研究では、クライアントがサーバからとってきた情報を記憶するドメインキャッシュサーバ(DCS)を提案する。これによって、サーバの負担を減らし、さらにはインターネット全体の負荷を減らし、サービスを向上させることを目的としている。また、サーバが故障した場合、DCSに蓄積された情報によってサービスを継続し、信頼性を高めることも目的の1つである。

## 2 背景

1994年5月に米国のゴア副大統領により提案された「スーパー情報ハイウェイ構想」[1]、その後、日本の郵政省により提案された「光ファイバ化計画」と、時代はいまマルチメディア[2]に向かって大きく変わろうとしている。

ネットワークの世界とて例外ではない。情報を狭い範囲ではなく、世界的視野で取り扱うようになってきており、ネットワークも世界レベルに及ぶものを必要としてきている。またそれとともに、マルチメディア化の傾向により情報の中心は、テキスト型のものから、より人間の五感に訴える映像・音声・動画といった型のものへ移っている。

そこで、現在注目を浴びているのがインター

ネット[3, 4, 5, 6]である。広範囲という点では世界で最も巨大なネットワークである。またインターネットが注目され始めた要因となったのが、WWWのブラウザであるMosaic[7, 8]の登場である。

このMosaicとは、インターネットを検索するツールであり、画像を中心としたメディアとマウスの使用によるユーザライクなインターフェースで、今まで研究機関等の閉じた世界で使用されてきたインターネットを一般の人々に広く知らしめることとなった。

しかし、Mosaicの登場といった情報の多角化により、ネットワーク間のトラフィックは日々増大している。そこで、このトラフィックを減少させる手段を考えなければならなくなった。

## 3 従来研究

本研究のモデルとなるインターネットの情報提供サービスの処理は、クライアント-サーバ方式をとっている。即ち、クライアントとなるユーザがサーバを選択し、サーバの中の欲しい情報を引き出す方式である。

ここで、毎回クライアントがサーバにアクセスするたびにデータを送ってはいりソース資源の利用効率が悪い。クライアント(もしくはクライアントが属するドメインなど)でそのデータを保存しておくこと(キャッシング)が考えられる。

インターネット上の広域情報サービスで利用されている情報のキャッシュ方式について、Proxyサーバ方式[9]とWWFS方式[10]の2つについて特徴を述べる。

### 3.1 Proxyサーバ

Proxyサーバ(図1)は、セキュリティ上の理由からクライアントが自由に外のネットワークにアクセスできない場合に、代理のサーバの役目を果たすものである。このProxyサーバを使うことによって、IPリーチャブルでないクライアントからもWWWを使うことができるようになる。また、マッキントッシュのクライアントのようにクライアント側で漢字変換をサポート

トしていない場合、Proxy サーバ側で Shift-JIS に変換することで、マッキントッシュ上で日本語が読めるようになる。

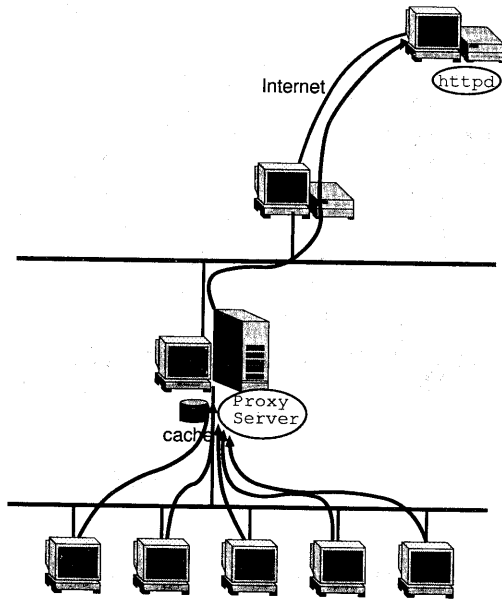


図 1: Proxy サーバの構成

さらに、Proxy サーバにはキャッシュの概念が存在する。Proxy サーバに一度インターネットから転送されたリソースを保存しておき、同じリソースへのリクエストがキャッシュにあるときは、インターネットへ同じものを取りに行くようなことはせず、キャッシュに保存しておいたリソースをクライアントに渡す。通常、Proxy サーバとクライアントは、イーサネットなどの LAN で接続されているので高速な表示ができる。

キャッシュの保存期間や、キャッシュのサイズは設定が可能である。ユーザがたくさんいるのなら、大容量のキャッシュを設定する必要があるし、キャッシュの保存期間が長すぎると、データが古くなる危険性がある。

問題点としては、キャッシュ情報の管理と、クライアントやサーバとの通信により Proxy サーバを常に通過するため、Proxy サーバに負荷が集中するといった点である。

### 3.2 WWFS

WWFS の特徴は広域ネットワーク環境における情報共有を支援する分散ファイルシステムを実現していることである。そのシステムの基本構成 (図 2) では、既存の protocols との整合性をとるために、クライアントからサーバへのファイルアクセスの中間に csd (cluster server daemon) と呼ぶサーバを置き、クライアント (LAN) 側と広域ネットワーク側にある情報提供サーバ群とのアクセス・プロトコルの変換を行う。

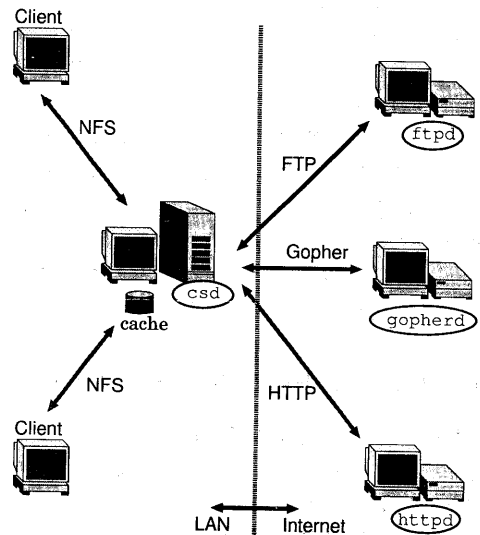


図 2: WWFS の構成

広域ネットワークの遅延には、csd における情報のキャッシング技術を用いて対応している。csd では、サーバにアクセスして得た情報をクライアントに渡して処理を終了するのではなく、いったんキャッシュに保管する。そして、同じ情報に再度アクセスがあったときは、サーバにアクセスせずにキャッシュに保管されているデータをクライアント側に渡す。ある情報への最初のアクセスには時間がかかるが、それ以降のアクセスはごく短時間で処理できる。さらに、キャッシュの導入によりアクセスの最適化も図れる。

csd でキャッシュをもつので、情報の同期の

問題が生じる。しかし、WWFS では疎な情報共有のモデルを採用しているため、同期についてはそれほど神経質にならなくてもよい。つまり、同じファイルに再度アクセスがあった場合はいったんサーバにアクセスし、情報が更新されていないかどうかをチェックするだけでよい。いかなるシステムであっても、情報の更新がおこなわれたかどうかは比較的短時間でチェックできるので、これはさほど大きなオーバーヘッドにはならない。更新されていないならばそのままキャッシュ内のデータを使えばよいし、更新されていたら、その時点であらためて情報をキャッシュに転送すればよい。

もう1つの問題は、キャッシュの管理である。キャッシュの大きさは有限である。WWFS では、一定期間以上アクセスされていない情報は、キャッシュから捨ててしまう。これは、メモリ管理などにおける LRU アルゴリズムと同様の方法である。このキャッシュ管理手法が有効なのは、ファイルアクセスのパターンにローカルティ、つまり特定のファイルへのアクセスが多いという偏りがみられる場合である。

#### 4 ドメインキャッシュサーバの導入

Proxy サーバと WWFS の両システムにはキャッシュの概念が存在する。しかし、どちらの方式においても

“1 台の端末に負荷が集中する”

といった問題点が存在する。そこで我々は、

- 負荷を分散する
- データの一貫性を考慮する
- 応答時間を短くする

という3点に重点を置き、これらを満たす方式として DCS システムを検討した。

図3に我々が提案する DCS システムの構成を示す。クライアントとサーバ間は FTP 等の通常のプロトコルを拡張したもので通信を行う。クライアントが取得した情報を DCS が記憶したり提供するために、クライアントと DCS との間に新しい通信プロトコルを設けている。

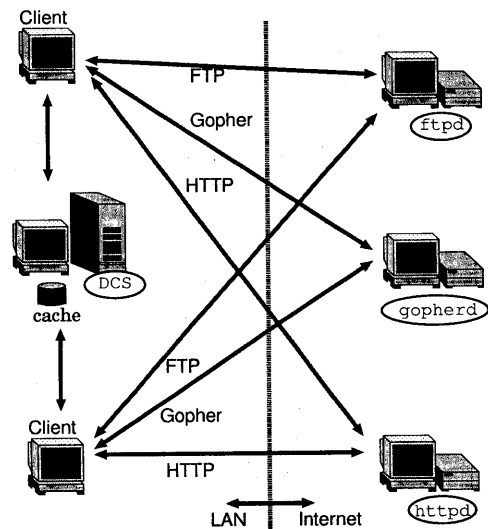


図3: DCS システムの構成

次に、DCS システムの処理の流れを述べる。まず、クライアントは通常どおりデータを送ってくれるようにサーバに要求に送る。サーバ側はクライアントから送られてきたメッセージをもとに、必要なファイルを捜す。その際、もしそのファイルが DCS に存在するならば、クライアント側にそこからデータを取ってくるようにする。これはサーバ側が、どのファイルの複製がどこの DCS に存在するかといった情報を持つことによって実現する。そして、存在しないならば、サーバがクライアントにそのデータを送り、クライアントは都合のいいときに(一貫性を考慮して)DCS にそのデータを送る。

以上によって、同じドメイン内のクライアントから同じサーバに同じデータを取りに行く場合には、そのサーバへのアクセスは1回ですむことになる。即ち、クライアントとサーバの間のネットワークの負荷を減らしたり応答時間を短縮することが可能になる。

DCS システムが Proxy サーバや WWFS と違うのは、前者は ftpd や httpd や gopherd 等のサーバとは別に新たに DCS というキャッシュのみを扱うサーバを導入したことである。従来

方式はクライアントとサーバとの中継を行う端末を使ってキャッシュを実現しており、その端末に負荷が集中する可能性があるといった問題点が存在する。DCS システムは中継を行う端末が存在しないため、負荷が集中するといった問題を避けることができる。

また、Proxy サーバではデータが最新かどうか分からないが、DCS システムでは、毎回サーバにアクセスするため確実に最新のデータを取得できると同時に、課金などの問題にも対応することができる。

## 5 ドメインキャッシュサーバシステムの実現

### 5.1 DCS システムの実装

現在、前の章で述べたシステムが有効であることを証明するために、DCS システムのプロトタイプを作成した。これは WWW をターゲットにして、このシステムに合うようにクライアント (Mosaic) 側とサーバ (NCSA httpd) 側を変更し、さらに DCS を作成している。図 4 にプログラム構成を示す。

主な変更点は、

- クライアントがデータを要求するときに http では “GET” を用いるが、拡張したプロトコルでは “DGET” や “RGET” を用いる。これによって新しい httpd と従来のものとの互換性を保つ。
- サーバにドメインごとのファイル参照履歴を管理する機能を取り付ける。これによって DCS のキャッシュの状態を管理する。
- DCS を新たに導入したため、クライアントと DCS との間の通信を行うプロトコルを作成する。

といった点である。

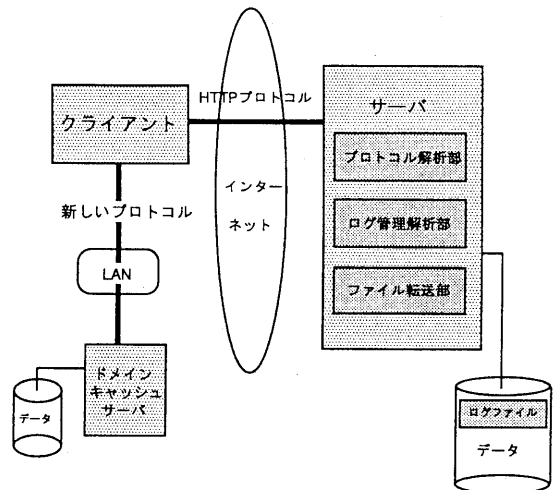


図 4: DCS システムプログラム構成

### 5.2 有効性評価シミュレーション

現在、DCS システムのプロトタイプ作成とは別に、このシステムが前述の Proxy サーバを用いたシステムと比較して有効であるかを評価するシミュレーションを作成している。

前提条件は、ネットワークは複数の LAN が専用回線で接続されている形であり、そこには情報獲得要求とそれに対する情報のみが流れるものとしている。主な入力パラメータとしては、各サーバが持つ情報のサイズと、それを要求するクライアントの要求頻度がある。評価するものは、各クライアントにおける情報獲得要求に対する応答時間と、各ネットワークのトラフィック量である。

次にシミュレーションの構成 (図 5) について述べる。本シミュレーションは以下のオブジェクトによって構成されている。

- scheduler  
時間の管理を行う。シミュレーション全体の時間を管理したり、job の発生待ちや cpu などの処理待ちを取り扱う。

- ws  
job によって発生した内容を cpu などの資源を利用して処理を行う。
- job  
ws(サーバ) の data を取得するジョブをどのくらいの時間間隔でどの ws(クライアント) に発生させるかを記述する。
- network  
どの ws が同じ回線を使用しているかを記述してあり、ws 間のデータの転送を処理する。
- cpu  
ws の cpu 資源を利用するとき使用する。
- disk  
ws の disk 資源を利用するとき使用する。
- data  
データのサイズや種類が記述されていて、job によって呼ばれクライアント、Proxy サーバ、DCS に複製が置かれる。

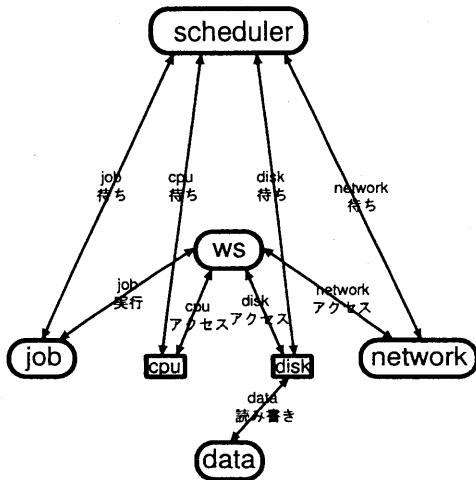


図 5: 有効性評価シミュレーションの構成

## 6 おわりに

本論文で DCS システムを導入した。このシステムでは Proxy サーバや WWFS などで問題となるキャッシュサーバへの負荷の集中を解決するために、クライアントとサーバ間にキャッシュサーバを配置するのではなく、独立させて負荷を分散している。また、クライアントは毎回サーバにアクセスするため、データが最新でありセキュリティや課金などにも対応している。

今後の課題としては、実装およびシミュレーションに基づくシステムの有効性について検討し、さらに、それらを世の中に出ているシステムに導入してもらえるようにしていきたい。

## 参考文献

- [1] 佐伯 純一: 情報スーパーハイウェイに乗り遅れるな, ミオシン出版 (1994.8)
- [2] 西垣 通: マルチメディア, 岩波新書 (1994.6)
- [3] 村井 純: インターネットユーザーズガイド, オーム社 (1994.5)
- [4] C.Malamud 著, 後藤、村上、野島 訳: インターネット縦横無尽, 共立出版 (株)(1994.3)
- [5] 村井、吉村監修, WIDE Project 編: インターネット参加の手引, 共立出版 (株)(1994.7)
- [6] 吉田茂樹: これがインターネットの世界だ, Internet Magazine, No1, pp.42-55, Impress (1994,10)
- [7] Vetter, Spell, Ward: Mosaic and the World-Wide Web, COMPUTER, Vol.27, No10, pp.49- 57(1994,9)
- [8] 石川一也: WWW 大航海記, Internet Magazine, No2, pp.28-59, Impress(1994.12)
- [9] 吉村 伸: httpd のインストール, UNIX MAGAZINE, 株式会社アスキー, pp67-74(1994.7)
- [10] 山口 英: WWFS(前編), UNIX MAGAZINE, 株式会社アスキー, pp34-42(1994.11)