

アプリケーション層での動的な WWW アクセス経路選択 — 移動エージェント支援によるモデルの提案 —

富川裕樹[†] 高井昌彰^{††} 山本 強^{††}

[†]北海道大学大学院工学研究科, ^{††}北海道大学大型計算機センター

WWW で公開されているコンテンツは膨大であり、同じ Cache Proxy を使用するユーザ間でアクセスの重ならない、キャッシュの効果の少ないコンテンツも多いと考えられる。このようなコンテンツへのアクセスにおいては、Cache Proxy を介したアクセスは不適切である。ここでユーザ個人の効率的なアクセスという視点に立つて考えると、できるだけ通信時の遅延時間の小さい経路を使用してコンテンツが読み出せることが重要である。

本研究では、ネットワーク上に複数存在する Proxy Server を移動エージェント支援により選択可能とするモデルを提案し、ある程度のアクセス経路選択の自由をユーザ個人に与え、ユーザにとって効率的なアクセスを実現することを目指す。

WWW Access Path Selection on Application Layer — A Model by Using Mobile Agents —

Yuki Tomikawa [†], Yoshiaki Takai ^{††}
and Tsuyoshi Yamamoto^{††}

[†] Graduate School of Engineering, Hokkaido University

^{††} Computing Center, Hokkaido University

A lot of contents on the WWW often have few cache effect because all clients sharing the cache proxy wouldn't request the same contents. From each client point of view, it's not necessary good to access via the cache proxy which is defined and given beforehand. The most important thing for every client is that he/she can always reach the specific contents with minimum delay.

In this paper, we propose a personal path selection model for WWW access by using mobile agents on multi-proxy environment. The objective of our model is to provide each client with efficient access by limited flexibility for network path selection on the application layer.

1 はじめに

WWW は情報発信するに当たっての敷居が低く、誰もが容易に広域への情報発信が可能であることから、ここ数年のインターネットの急速な普及の原動力の一つとなっている。WWW の利用増により通信量が非常に増加したインターネットにおいて、効率的なアクセスを実現するための手段の一つとして Proxy Server が用いられている。Proxy Server とはクライアントとサーバの間で交わされる、種々のプロトコル (HTTP, FTP, gopher 等) に基づく通信を中継するアプリケーションである。サーバから転送されるデータをキャッシングし、それをクライアントに提供する機能を持つ Proxy Server は Cache Proxy と呼ばれる。

Cache Proxy は、そこを経由してアクセスするユーザが同一の Web Server の同一のコンテンツにアクセスする頻度が高い場合、あるいは更新頻度の低いコンテンツに対してはキャッシュから読み出せる可能性が高くなり、ネットワークのトラフィック低減効果が高い。また、アクセス先の Web Server よりもネットワーク的に近い Cache Proxy から読み出すことにより、短時間でコンテンツにアクセスできるため、ユーザにとってもメリットがある。

しかし、WWW で公開されているコンテンツは膨大であり、同じ Cache Proxy を使用するユーザ間でアクセスの重ならないコンテンツも多く、この場合はキャッシュの効果は非常に小さくなってしまふ。このようなコンテンツへのアクセスにおいては、Cache Proxy を介したアクセスは必要ではなく、アクセス先に無関係に多くのユーザが同一の Cache Proxy を集中して使うことにより、その Cache Proxy 自体がボトルネックとなる要因を増加させていると考えられる。

ここでユーザ個人の視点に立って見れば、効率的なアクセスとは、できるだけ通信時の遅延時間の小さい経路を使用して求めるコンテンツを読み出せることに他ならない。しかし現在の所、ユーザがアクセス先ごとに経路を選択する手段は提供されていない。

そこで、Proxy Server のプロトコルの中継器としての機能に着目する。ネットワーク上に複数の Proxy Server が存在し、それらをユーザがアクセス先に応じて選択して使用することが可能であれば、回線が細かったり、トラフィックが多くなって遅くなっている経路を避けてアクセスする、というある程度の経路選択の自由度をユーザに与えることが可能となる。

しかし単に Proxy Server が複数存在するだけでは、どの Proxy Server を使用すればアクセスが効率化されるかの判断のための情報は、ユーザ側の機構だけでは十分に得ることはできない。

本論文では、ネットワーク上に Proxy Server が複数存在する環境において、移動エージェント [1, 2] の支援により経路選択の指標となる情報の収集を行うモデルを提案し、ある程度のアクセス経路選択の自由をユーザに与え、ユーザにとって効率的なアクセスを実現することを目指す。

2 アプリケーション層での経路選択

TCP/IP プロトコル・アーキテクチャ [3] を、ネットワークアクセス層、インターネット層、トランスポート層、アプリケーション層の 4 層で表した場合、データグラムレベルでの経路制御は、インターネット層で行われている。今回提案するモデルでは、データグラムレベルのような下位層での経路制御をユーザに許可するわけではなく、アプリケーション層で可能な範囲内で、ある程度ユーザに経路の選択の自由を与えようという考えに基づいている。アプリケーション層での経路選択を実現するため、広域ネットワーク上に Proxy Server が複数存在する環境を考える。

2.1 Proxy Server が複数存在する環境

ここで公共的にアクセス可能な、プロトコルの中継器としての Proxy Server が広域ネットワーク上に複数存在する環境を考える。

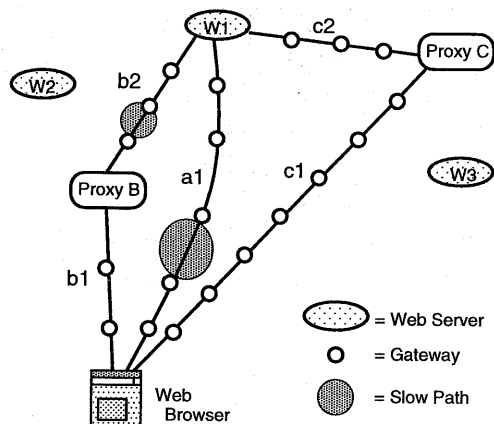


図 1: Proxy Server が複数存在する環境

現在の一般的な Web ブラウザでは、プロトコルごとに固定的な一つの Proxy Server しか指定できない。このため、もしユーザがアクセス先ごとに使用する Proxy Server を切り替えようとするなら、その都度手動で設定を書き換える必要がある。また、どの Proxy を用いれば効率的であるかの判断に必要な情報を与える機構は現状では提供されていないため、効率的な Proxy Server を選択するのは困難である。

ある Web Server へのアクセスにおいて、最も効率的なアクセスが可能となるのは、ユーザの Web ブラウザ-Proxy Server 間と Proxy Server-Web Server 間の通信の遅延時間の和が最小になるような Proxy Server を選択した場合であると考えられる。

図 1 を用いて、Proxy Server が複数存在する環境でのアクセス経路選択について考える。

図 1 において、Web Server W1 へアクセスする経路は、Proxy Server を経由しない、Proxy B 経由、Proxy C 経由の 3 通りがある。通信時の遅延時間は、Web ブラウザと W1 まで直接、Proxy B まで、Proxy C までそれぞれの間を $a1$, $b1$, $c1$ とする。また W1 と Proxy B, Proxy C それぞれの間の遅延時間を $b2$, $c2$ とする。

3 通りの経路それぞれの遅延時間の和は $(a1)$, $(b1 + b2)$, $(c1 + c2)$ である。ここで図 1 のように、W1 - Proxy B 間の一部、および Web

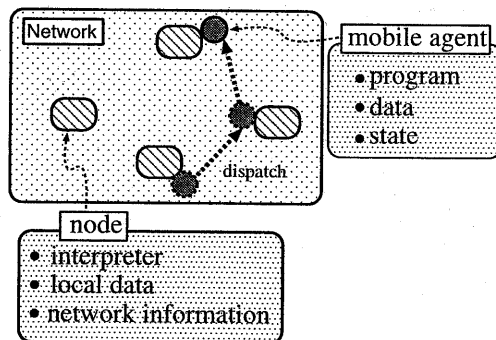


図 2: 移動エージェントの概要

Browser - W1 間の直接の経路の一部に回線の帯域が狭い、あるいは通信が混雑しているなどで非常に遅延が大きいパスがある場合には、途中で経由するゲートウェイの数が多いにもかかわらず、Proxy C を使用してアクセスするのが 3 通りの経路の中で最も効率が良い、という状況も有り得る。

しかし、 $a1$, $b1$, $c1$ はユーザ側から計測可能であるが、 $b2$, $c2$ についてはユーザ側からでは計測は困難であるため、どの Proxy Server が最も効率的かを判断するための情報が不足している。そこで、Proxy Server とアクセス先の Web Server 間のネットワーク状況についての情報を得るための手段として、移動エージェントを利用することを考える。

3 移動エージェント

従来の分散システムでは、多くの場合ノード上のプロセスに知性があり、プロセス間のデータのやり取り（メッセージパッシング）により処理が行われていた。これに対し、自律的オブジェクトパラダイム (autonomous objects paradigm[4]) に基づく分散システムでは、メッセージは単なるデータではなく移動可能なプログラムであると考えられる。また、知性の主体はメッセージにあり、ノードにはメッセージを処理するインタプリタが存在すると考える。移動可能プログラムで構成されるエージェントは移動エージェントと呼ばれる。移動エージェント

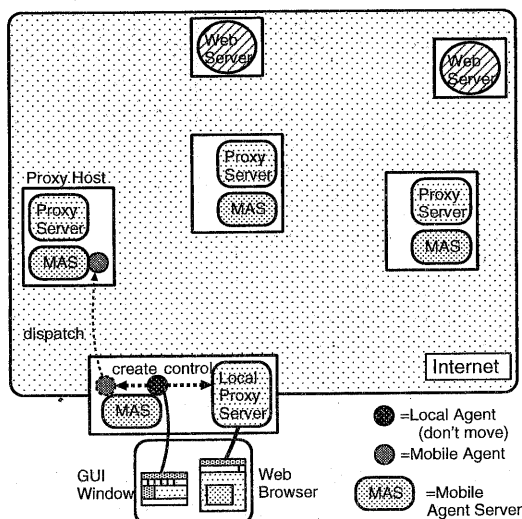


図 3: モデルの概要

の概要を図 2 に示す。

移動エージェントは自身の持つプログラム、データおよび状態に基づき、ネットワーク上を自律的な判断を行いながら動的に移動し処理を行う。移動先のノードにおいては、そのノードのみが持つ固有のデータや、そのノード特有の情報（そのノードを取り巻くネットワークの状況等）を利用した処理が可能である。

4 移動エージェント支援による経路選択モデル

移動エージェントの支援によりアプリケーション層での WWW のアクセス経路選択を行うモデルを提案する。

4.1 モデルの概要

提案するモデルの概要を図 3 に示す。移動エージェントのサーバと、キャッシュ機能を持たない Proxy Server を一組にして設置した計算機を Proxy Host と呼ぶこととする。広域ネットワーク上に Proxy Host を分散配置する。ユーザが Web ブラウザを実行している計算機、あるいはネットワーク的によく近い

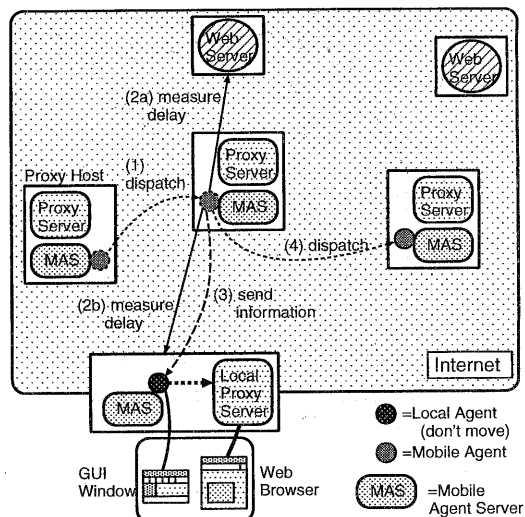


図 4: 移動エージェントの動作

他の計算機上に移動エージェントのサーバを置き、そこにローカルエージェント（移動しないエージェント）を配置する。また、ユーザの Web ブラウザから Proxy Server として指定する Local Proxy Server を起動しておく。Local Proxy Server はローカルエージェントから制御可能な機構を持つ独自のものである。ローカルエージェントは、ユーザと対話するための GUI ウィンドウを表示する。

4.2 実行時の動作

このモデルに基づくシステムを利用して Web Server にアクセスしようとする時、ローカルエージェントの GUI ウィンドウからユーザがアクセス先の URL アドレスを指定すると、ローカルエージェントは移動エージェントを生成し、利用可能な Proxy Host のリストを渡す。

移動エージェントはリストに基づき Proxy Host の移動エージェントサーバを巡回する。移動エージェントの動作の例を図 4 に示す。巡回中の移動エージェントの動作は以下の通りである。

(1) 次の移動エージェントサーバへ移動

(2) 遅延時間の計測

(2a) 現在地とアクセス先 Web Server 間

(2b) 現在地と Local Proxy Server 間

(3) 計測した情報をローカルエージェントへ通知

(4) リストにまだ巡回していない Proxy Host があれば、(1) へ戻る。全て巡回し終った場合は、ローカルエージェントへ終了を通知して自身を消滅させる

Local Agent は移動エージェントからの情報に基づき、どの Proxy Server を使用するかを決定し、Local Proxy Server へ通知する。Local Proxy Server は指定された Proxy Server 経由で目的の Web Server へアクセスし、ユーザの Web ブラウザへ表示する。

以前アクセスした Web Server へ再度アクセスする際、Local Proxy Server は以前使用した Proxy Server を記憶しておき、自動的にその Proxy Server を使用する。これにより、Local Proxy Server はアクセス先ごとに効率のよい Proxy Server を学習していく。ユーザが指示した場合には、移動エージェントを用いた Proxy Server の決定を再び行うことも可能である。

5 実装方針

今回提案したモデルでは、全てのアクセス先に対して効率化が可能なわけではないため、従来の Cache Proxy が有効なアクセス先については Cache Proxy を経由したアクセスを併用する。Local Proxy Server は、ユーザからの指示がない場合には Cache Proxy を介してアクセスを行うこととする。

モデルに使用される Proxy Server と移動エージェントサーバは、すでに開発されて使用されているものがほぼそのまま適用可能であると考えられる。また、Web ブラウザについては Local Proxy Server をブラウザが使用する Proxy Server として指定するだけで利用可能であるので、Web ブラウザ自体には一切手を加える必要はない。

現在、提案モデルのその他の構成要素であるローカルエージェント、移動エージェント、および Local Proxy Server を実装中である。移動エージェントはネットワーク上で移動先の計算機上でプログラムを実行するため、セキュリティ上の問題を十分考慮した実装が必要となる。また、広域に配置する Proxy Server については、不正な匿名アクセスにより悪用されないようにするための機構を持つものを用いる必要がある。

実装終了後、実際にモデルに基づくシステムを稼働させ、提案モデルの有効範囲と有効度を調べる評価実験を行う予定である。

6 おわりに

本論文では、ネットワーク上に Proxy Server が複数存在する環境を想定し、移動エージェントの支援により WWW アクセス経路選択を実現するモデルの提案を行った。

このモデルでは、アプリケーション層においてある程度のアクセス経路選択の自由をユーザに与え、ユーザにとって効率的なアクセスを実現することを目指している。

参考文献

- [1] 中島 達夫, 渡部 卓雄: “分散オブジェクト技術とモバイルエージェント”, 信学誌, Vol. 80, No. 4, pp.357-363 (1997).
- [2] 飯田一朗, 西ヶ谷岳: “モバイルエージェントとネットワーク”, 情報処理, Vol. 38, No. 1, pp.17-23 (1997).
- [3] C. Hunt, 村井 純監訳: “UNIX システム管理者のための TCP/IP ネットワーク管理”, オーム社, pp. 10-30 (1994).
- [4] L. F. Bic, M. Fukuda and M. B. Dillencourt: “Distributed Computing Using Autonomous Objects”, *IEEE Computer*, Vol. 29, No. 8, pp.55-61 (1996).