広域分散配置Webサーバにおける最適サーバ探索システムの検討

荻野司[1]，松田和宏[2]，須藤一顕[2]，針山欣之[2]，向阪正彦[2]，殖栗俊明[2]

Webサーバの負荷を分散させるために、サーバクラスターを構成しアクセスを分散させる方法や、地理的、ネットワーク的に分散したミラーサーバを配置することが一般的に行われている。しかし、時々刻々変化するサーバ、ネットワーク状態に応じて、クライアントを最適なサーバに導くことは、種々の提案がなされているものの、決定的な解決方法が見いだされていない。

本稿では、広域分散配置されたWebサーバ群において、動的に変化するサーバ、ネットワーク状態を計測する手段の提案を、また、その計測手段を用いて真に最適なサーバを検出する新たな方式の提案を、さらに、アクセスクライアントを検出した最適なサーバに導くための最適サーバ探索システムの提案をする。

本方式では、経路情報（BGP：Border Gateway Protocol）のAS path(Autonomous System)をネットワークの論理的な距離計測手段判断子として用いる。また、各種サーバ、ネットワーク情報計測ツールを用いた結果と併せて、最適なWebサーバを決定するものである。本稿では、日米各々に実証実験用Webサーバサイトを構築、実際のインターネット上においてプロトタイプシステムを実装し、性能評価を実施した結果についても併せて報告する。

Study of an Efficient Server Selection Method for Widely Distributed Web Server Network

Tsukasa Ogino[1], Kazuhiro Matsuda[2], Kazuaki Sudo[2], Yoshiyuki Hariyama[2], Masahiko Kousaka[2], Toshiaki Ueguri[2]

In order to disperse the load on a Web server, generally the server cluster is configured to distribute access requests, or mirror servers are distributed geographically or situated on different networks. However, although there are several proposals for leading clients to the most efficient server according to the constantly changing server and network condition, as yet no definitive solution has been proposed.

In this document, we propose a measurement method for dynamically changing server and network environment, a new selecting method to find the most efficient server based on the measurement method, and an efficient server selection system for leading the access clients to the most efficient server among the distributed Web server network.

Under this method, we use AS (Autonomous System) path routing information of BGP (Border Gateway Protocol) as the factors for evaluating the logical distance of the network. We also try to determine the most efficient Web server by using various server/network information measurement tools. Experimental Web server sites have been set up in both Japan and the USA, and a prototype system was implemented on the Internet and its performance was evaluated; results of the experiments and evaluation are included in this report.

1. Introduction

インターネットが急速に普及するなかで、手軽に情報を配信・閲覧する道具として、WorldWideWeb（以降Webと称する）を用いた情報配信システムが急速に拡大している。最近では、シドニーオリンピックの公式Webサーバの如く、全世界をアクセス可能対象としたWebサーバサイトも出現し、その重要性は一段と高まってきている。一方、このような全世界的に情報配信を行うWebサーバサイトは、全世界から集中してアクセスをされるために、多量なアクセスに対して十分な準備をする必要がある。このような対策として、Webサーバの負荷分散が重要な検討課題として着目されている。Webサーバの負荷を分散させるためには、サイト内に複数のWebサーバでサーバクラスターを構成する事により、集中するアクセスを複数のWebサーバに分散させる方法や、地理的、ネットワーク的に異なる他サイトに、同様のコンテンツを持つミラーサーバを配置することによって、地域的、ネットワーク的にアクセスを分散させるといった方法が一般的に行われており、

[1] インターネット総合研究所
[2] FastNet,Inc. インターネット事業部

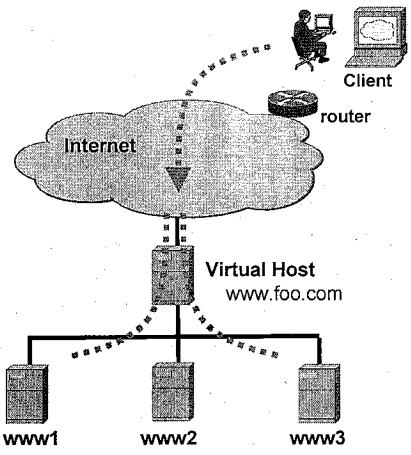

図1 バーチャルホスト方式

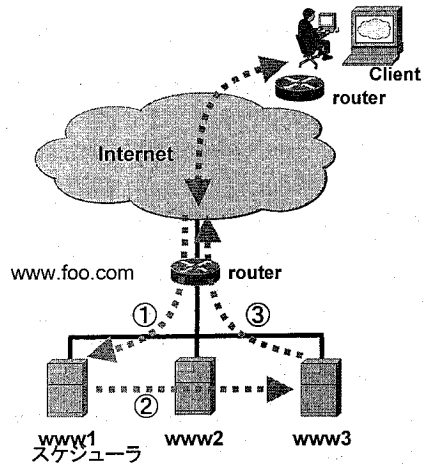

図2 ＴＣＰコネクションホップ方式

アクセスするクライアントに対して、最適なサーバに導くための、種々の提案がなされている。

しかし、後述のような理由で、このような一般的な手法では、集中するアクセス負荷を的確に分散する事は非常に困難である。それは、以下の理由からである。Web サーバへのアクセスは、地域的、ネットワーク的に均一ではない。また、アクセス時間においてもアクセス分布は変化する。例えば、Web サーバへのアクセスは、地域的、ネットワーク的に均一ではなく、むしろかなり偏ったアクセス分布を示す。全世界的な広域の Web サーバがそれで、その地域のローカル時間にアクセス分布は大きく影響される。アクセスされる時間においても、ビジネスタイムと深夜とで、そのアクセス分布は大きく異なる。

本論文では、広域分散配置された Web サーバ群において、動的に変化するサーバ、ネットワーク状態を計測する手段の提案を、また、その計測手段を用いて真に最適なサーバを検出する新たな方式の提案を、さらに、アクセスクライアントを検出した最適なサーバに導くための最適サーバ探索システムの提案をする。本方式では、経路情報 (BGP : Border Gateway Protocol) の AS path (Autonomous System) をネットワークの論理的な距離計測手段判断子として用いる。また、各種サーバ、ネットワーク情報計測ツールを用いた結果と併せて、最適な Web サーバを決定するものである。本稿では、日米各々に実証実験用 Web サーバサイトを構築、実際のインターネット上においてプロトタイプシステムをインプリメント、性能評価を実施した結果についても併せて報告する。

本論文では、2 章で、従来の負荷分散方式について、概観する。3 章では、本研究で提案する最適サーバ探索システムについて述べ、4 章で使用するパラメータの性能評価、及び、本提案のプロトタイプシステムのテスト結果について述べる。5 章で本提案方式について考察をし、6 章で今後の課題とまとめを述べる。

2. 従来の負荷分散方式

現在、いくつかの手法が提案されている。その代表的な 2 手法について、図 1, 図 2 をもとに概説する。

2.1 バーチャルホスト方式

この方式は、図 1 のようなバーチャルホストを利用する方式である。これは、集中するアクセス負荷を複数のサーバで均一に分散させる事を目的としている。この例では、アクセスクライアントが、www.foo.com なる URL (アクセスアドレス) にアクセスした場合を考えている。www.foo.com なるアクセスがあると、まずバーチャルホストにアクセスされる。バーチャルホストでは、そのアクセスを配下の各 Web サーバの負荷状況から、適切にアクセスの分散を図る。実際には、www1, www2, www3 のいずれかでクライアントからのアクセスに対して所定の情報を配信するのだが、バーチャルホストで一度アクセスを受ける事によって

適切なアクセスの分散を図っている。

しかし、図 1 で述べたバーチャルホストを利用した場合には、下記のような問題点が考えられる。

1) 専用のバーチャルホストの役目をする装置が別途必要である。

2) 専用のバーチャルホストですべてのアクセスを一時受け付けるために、多くのアクセスがあるとこのバーチャルホストの処理能力がボトルネックになる。

つまり、実質バーチャルホストの処理能力がこの Web サーバサイトの処理能力になってしまう。

3) バーチャルホスト方式は、あくまで自サーバサイト内のアクセスについて受付け、自サーバサイト内の Web サーバのアクセスを平準化するため、分散配置された Web サーバサイトには、各々バーチャルホストが必要になる。そして、各々に所定の異なる URL を設定する必要があった。そのため、アクセスする側が、分散配置されている Web サーバサイトを自ら選択する必要がある。

4)アクセスする側が Web サーバサイトを選択するために、各々の Web サーバサイトへのアクセスを平準化する事ができない。

2.2 TCP コネクションホップ方式

次に、TCP コネクションホップによる転送方式について図2を用いて説明する。クライアントからアクセスがあると、まず、www1にアクセスされる。www1では、スケジューラが動作しており、このスケジューラで最適な Web サーバ www3 を選択し、クライアントからのアクセスに応答する。つまり、www1 が常にアクセス要求に応え、最適 Web サーバ (www1、www2、www3) が応答する。この方式では、スケジュール機能においてアクセスの分散を図っている。

しかし、図2で述べた TCP コネクションホップ転送方式を利用した場合には、下記のような問題点が考えられる。

1)常にスケジューラが常駐している Web サーバにアクセスが集中するため、万一、スケジューラが常駐している Web サーバに障害が発生した場合には、このサーバサイト全体が機能できなくなる。

2)分散配置された Web サーバサイトへのアクセスを最適に制御する事ができない。この方式は、スケジューラが管理している Web サーバサイト (自ネットワーク) 内でのアクセスの平準化を目的としている方式であるため、前述のバーチャルサイト方式と同様に、各々の Web サーバサイトに所定の異なる URL を設定する必要がある。そのため、アクセスする側が、分散配置されている Web サーバサイトを自ら選択する必要がある。

3) 前述のバーチャルサイト方式と同様に、アクセスする側が Web サーバサイトを選択するために、各々Web サーバサイトへのアクセスを平準化する事ができない。

3. 最適サーバ探索システム

3.1 最適サーバ決定方式の概要

広域に分散された Web サーバ群から、アクセスクライアントにとっての最適な Web サーバを決定するにあたり、本提案の方式では、以下の事項を目標として定義した。

1) 同一サイト内において負荷を均一に分散する。
2) 複数に分散配置されたサイト間においても負荷を均一に分散する。
3) Web サーバに別途過度の負荷を要求しない。
4) クライアントからのレスポンスには、高速で対応をする。

上記目的を達成するため、本方式では、1.ネットワーク情報を収集するモジュール、2.1 のモジュールで収集したデータを受け取り分析し、その結果から総合的に最適サーバを決定するモジュール、3.2 のモジュールの結果に基づき最適サーバへ誘導するモジュールからなるシステムを構築した。以下にネットワーク情報を収集するモジュールの特徴を列挙する。

1) アクセスクライアントと Web サーバとのネットワーク的距離を経路情報より取得。具体的には、BGP の AS Path 情報により論理的なネットワーク距離を収集、各々の Web サーバとの距離を比較し、最短の path を持つ Web サーバを求める。
2) 各々の Web サーバから、クライアントまでのネットワーク状態を計測。具体的には、RTT、パケットロス、スループット、ルータホップ数を測定する。
3) サイト内ネットワーク情報計測。サイト内の送受信ネットワークトラフィック、パケットエラー数、コリジョン発生数、GW ルータトラフィック、GW ルータ廃棄パケット数を測定する。
4) 各々Web サーバ状態を計測する。TCP コネクション確立数、ディスク負荷、CPU アイドル状態、ロードアベレージなどのサーバの負荷を測定する。

本方式では、以上の計測結果に基づいて最適な Web サーバを決定する。

3.2 プロトタイプシステム構成

本システムは次の4つのシステムで構成されている (図3)。

- NS-Agent
- Network Status Server (NS Server)
- Network Status Probe (NS-P)
- Route Server

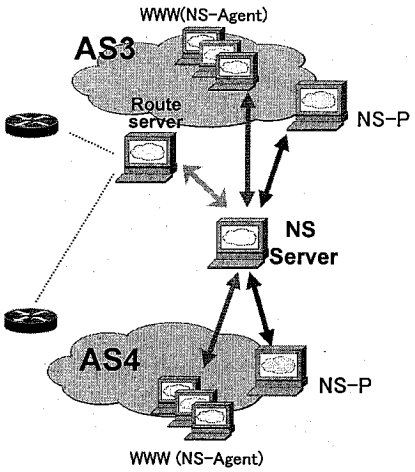

図3 システム構成

NS-Agent は、最適サーバへ誘導するモジュールである。NS Server は、最適サーバを決定するモジュールである。また、NS-P 及び Route Server は、ネットワーク情報を計測するモジュールである。

3.3 システム機能

(1) NS-Agent

NS-Agent は、Web サーバの Apache に「モジュール (mod_nss)」として実装している。モジュール[3]は、Apache の機能を自由に変更できるように提供された仕組みである。開発した NS-Agent は C 言語で約 1,400 行からなる。NS-Agent が起動されるタイミングは、Apache が、クライアント (Web ブラウザ) からの HTTP リクエストを受信したときである。 つまり NS-Agent は Apache が HTTP リクエストを受信する毎に Apache 内部でコールされ、受信したリクエストに対して必要な処理を実行する。コールされた NS-Agent は、次の 2 つの処理を実行する。

- 最適サーバを知るため NS サーバへ最適サーバ情報を問い合わせる
- HTTP レスポンスでクライアントへ最適サーバを知らせると同時に最適サーバへリダイレクトさせる指示を出す

具体的には、クライアント (Web ブラウザ) からの HTTP リクエストが Apache に届くと Apache は NS-Agent をコールする。NS-Agent は、クライアントの IP アドレスを Apache 内部で取得し NS サーバへ最適サーバ情報コマンドを発行する。このときパラメータにクライアントの IP アドレスをセットする。NS-Agent から最適サーバ情報コマンドを受信した NS サーバは、分散配置されたサイトの中からクライアントに最も近いサイトを決定し、サイト内にあるサーバの IP アドレスを NS-Agent へ返す。サーバの IP アドレスを受信した NS-Agent は、HTTP レスポンスをクライアントに返す。このとき最も近いサイトを指定するサーバの IP アドレスをレスポンスにセットし、HTTP レスポンスコードには 302 をセットする。レスポンスコード 302 は、Moved Temporarily[4] の意味で、このコードを受信したクライアントの Web ブラウザは、指定された ＩＰアドレスのサーバにリダイレクトする。

(2) Network Status Server

Network Status Server (以下 NS サーバ) は、本システム全体の中枢的な位置にあるシステムであり次の５つの機能を持っている。

- ネットワーク情報取得機能
- 最適サイト決定機能
- 最適サーバ決定機能
- データ集約機能
- サイト管理機能

ネットワーク情報取得機能では、NS-P が探査したネットワーク情報を収集する。最適サイト決定機能では、RS やネットワーク情報取得機能で NS-P から取得した計測データから最適サイトを決定する。最適サーバ決定機能では、各サイトの NS-P マスターから現在クライアントのアクセスを受けさせるのに最適な状態にあるサーバを取得しかつ決定する。ここでの現在最適な状態にあるサーバとは負荷の最も軽いサーバをいう。データ集約機能では、クライアントの情報を集約し管理している。ここで管理している情報は、クライアントの IP アドレス、クライアント・サーバ間のネットワーク探査結果である。サイト管理機能では、各サイトの負荷状態を把握している。

NS サーバ は C 言語で約 5600 行からなり、マルチスレッドで動作する。OS は現在のところ FreeBSD である。

(3) Network Status Probe

NS-P はサイト内にあるすべてのサーバ上に実装され、そのうちの一つがマスターとして動作する。残りの NS-P はスレーブとして動作する。スレーブは、自己のサーバ上の負荷を随時測定している。マスターはネットワーク情報探査機能などクライアント・サイト間の測定やサイト内ネットワーク情報探査機能の GW ルータトラフィックの測定を行う。またマスターはサイト内の全スレーブからスレーブが測定した各サーバ

上の負荷を収集し、サイト全体の負荷を集中管理する。(図4)

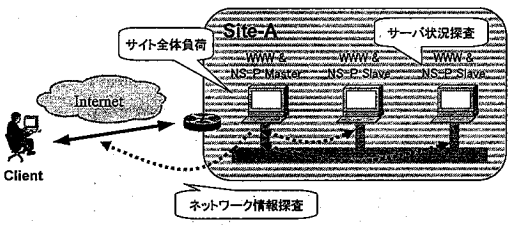

図4 NS-P構成

Network Status Probe (以下 NS-P) は、次の3つの探査機能を持つ。

・ネットワーク情報探査機能
・サイト内ネットワーク情報探査機能
・サーバ状況探査機能

ネットワーク情報探査機能では、クライアント<->サイト間のRTT(Round Trip Time)、パケットロス、スループット、ルータホップ数を測定する。サイト内ネットワーク情報探査機能では、サイト内の送受信ネットワークトラフィック、パケットエラー数、コリジョン発生数、GWルータトラフィック、GWルータ廃棄パケット数を測定し、サイト内サーバ稼動状態の監視も行う。サーバ状況探査機能では、TCP コネクション確立数、ディスク負荷、CPU アイドル状態、ロードアベレージなどのサーバの負荷を測定する。

探査機能の実装は、UNIX コマンドと同等な機能を直接組み込んだ。探査機能のそれぞれの計測項目とコマンドの対応は表1-表2のとおりである。

表1 ネットワーク情報探査機能一覧

計測項目	測定単位	同等コマンド
RTT	ms	ping
パケットロス	%	ping
スループット	bit/s	(独自)
ルータホップ数	段数	traceroute
ASパス長	段数	(RS)

表2 サイト内ネットワーク情報探査機能一覧

計測項目	測定単位	同等コマンド
送受信トラフィック	packet/s	netstat n
パケットエラー数	packet/s	netstat n
コリジョン発生数	回数	netstat n
ルータトラフィック	bit/s	snmp
ルータ廃棄パケット数	packet/s	snmp

表3 サーバ状況探査機能一覧

計測項目	測定単位	同等コマンド
TCP コネクション数	本数	netstat
ディスク負荷	転送回数/s	iostat n
CPU アイドル値	%	iostat n
ロードアベレージ	プロセス数	uptime

スループットについて

Telnet のような小刻みにデータが発生する通信では帯域の影響をあまり受けることがない。そのため、今回は、遅延性能が重要になり、スループットは無視できるとした。逆に、Web などのバルクデータの場合には、帯域の影響を大きく受けるためスループットのほうが重要になる。ネットワークのスループットを計測するツールとして、ttcp[1]や netperf[2]などのコマンドがあるが、これらは計測するネットワークの送信側と受信側両方の拠点でプロセスを起動する必要があり、今回のような不特定なクライアントとサイト間の計測をリアルタイムに行うことは不可能である。そこで、スループットを計測する機能は独自に作成、組み込むことを行った。プロトタイプシステムでのスループット計測機能は次の条件で作成した。

・サイトからの機能だけで計測できること
・リアルタイムにすばやく計測できること

具体的な実装は次のとおりである。クライアントに向けて 64バイトの ICMPエコー要求を2回連続して発行し、そのあと 1024バイトの ICMP エコー要求を1回発行する。2回目の64バイトの測定結果 (RTT1) と最後に発行した 1024 バイトの測定結果 (RTT2) の差分からスループットを算出する。式(1)に算出式を示す。

スループット
$$= ((1024 - 32) \cdot 8) / ((RTT2 - RTT1) / 2) \quad (1)$$

1回目の 64 バイトのデータは、クライアントプログラムをロードさせる (オーバヘッド) 目的で行うため、データとしては無視する。以降、本論文でいうスループットはプロトタイプに組み込んだスループットの測定結果を指す。

NS-P はC言語で約7300行からなり、マルチスレッドで動作する。OS は現在のところ FreeBSD である。

(4) Route Server

Route Server は zebra[5] を本システム用に変更して使用した。zebra は各種経路制御プロトコルをサポートしているが、今回はBGP プロトコル処理部に対してだけ変更を加え、コンフィグレーションおよびコマンドの追加を行った。具体的には、コンフィグレーションでは各サイトの Gate way(GW)ルータアドレスを登録できるようにし、追加したコマンドではクライアントの IP アドレスを受信できるようにした。このコマンドを受信すると、zebra はコマンドにセットされたクライアントの IP アドレスから、クライアントと各サイトの GW ルータ間の AS パス長を算出し、各サイトの GW

ルータと AS パス長のリストをコマンド発行元へレスポンスする。

3.4 アルゴリズム

プロトタイプでは最適サーバを2ステップのアルゴリズムで決定している。最初のステップは収集したネットワーク情報から最適サイト決定する。次に最適サイト内で最も負荷の少ないサーバを探査し最適サーバに決定する。最適サイト決定アルゴリズムはNSサーバに実装し、最適サーバ決定のアルゴリズムは、NS-Pマスターに実装している。NSサーバは最適サイト決定後、最適サイトの NS-P マスターに最適サーバを問い合わせる。最適サイト決定のアルゴリズムと最適サーバ決定のアルゴリズムについて以降に詳細を述べる。

3.4.1 最適サイト決定

最適サイト決定には、クライアントのアクセスが初回なのかそうでないかによって次の2つの方法がある。

(1) ASパス長で決定する（初回アクセスのとき）
(2) 収集してあるネットワーク情報から決定する（再アクセスのとき）

(1) ASパス長で決定

NSサーバは、ASパス長を取得するため、クライアントのIPアドレスをセットしたASパス長取得コマンド（ASInfoCOM）を発行し、レスポンス（ASInfoRSP）としてASパス長のリストを得る。ここで得たリストから最適サイトを決定する。レスポンスの内容は具体的にはサイトのルータアドレスと、クライアントまでのASパス長がリストになっている。このリストの中からAS パス長が最小値になるサイトを最適サイトに決定する。例えば、

```
RTList : address 202.228.128.217  AS Length 1  <- SiteA
RTList : address 216.98.110.62    AS Length 3  <- SiteB
```

の場合、 AS Length が小さい方の SiteA が最適サイトになる。

(2) ネットワーク情報から決定

ここでは、ネットワーク状態値とサイト状態値を使って最適サイトを算出する。まず、集約テーブルに確保されているネットワーク情報（RTT,ルータホップ数等）からネットワーク状態値を算出する。次に、サイト内ネットワーク情報（送受信ネットワークトライック、コリジョン発生数等）からサイト状態値を算出する。算出されたネットワーク状態値と、サイト状態値を加算した結果がサイトの最適サイト判定値である。

そして、それぞれのサイトの最適サイト判定値を比較し、最適サイトを算出する。最適サイト決定の流れを図5に示す。

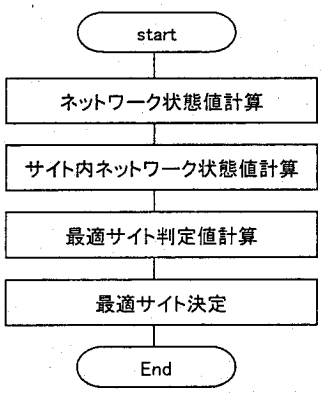

図5 最適サイト決定の流れ

具体的な計算について以下に述べる。ネットワーク状態値計算は、現在のネットワーク状態の各パラメータにデータの重みを加味して求める。計算式は式(1)のようになる。

ネットワーク状態値
= ASL・A + RTT・B + RN・C + PL・D + TP・E　(1)

各記号の意味は、次の測定データおよび係数である。

- ASL：クライアントまでのASパス長
- RTT：クライアントまでのRTT計測値
- RN ： クライアントまでのルータホップ数
- PL ：パケットロス率
- TP ：クライアントまでのスループット
- A-E：重み係数

プロトタイプシステムでは、ネットワーク状態の各パラメータの値は計測しているが、計算に使用するパラメータはまずASパス長と ルータホップ数の2つを有効にした。重み係数は0.5づつである。その他のパラメータの係数はネットワーク情報の有効性を調べるための予備実験をして得られた結果から考察することにした。

サイト内ネットワーク状態値は、サイト内のネットワーク状態の各パラメータからデータの重みを加味して計算する。計算は式 (2) のようになる。

サイト内ネットワーク状態値
= CS・F + PS・G + ES・H + RTR・I + RTE・J　(2)

各記号の意味は、次の測定データおよび係数である。

- CS ：サイト内で発生したコリジョン数
- PS ：サイト内で発生したパケット数
- ES ：サイト内で発生したパケットエラー数

・RTR：GW ルータトラフィック
・RTE：GW ルータ廃棄パケット数
・F-J：重み係数

プロトタイプシステムでは、計算に使用するパラメータはコリジョン数、パケット数、GW ルータトラフィックを有効にした。コリジョン数とパケット数からコリジョン発生率を求め、GW ルータトラフィックから帯域占有率を求め、求めた 2 つの数値に重み係数を 0.5 づつ与えた。

3.4.2 最適サーバ決定

サーバ負荷をリアルアイムに計測し、負荷の少ないサーバを最適サーバに決定する。計測した負荷からサーバ評価値を算出し、各サーバの評価値を比較する。サーバ評価値は、サーバ負荷の各パラメータからデータの重みを加味して計算する。計算は式 (3) のようになる。

サーバ評価値
= LINK・K + IO・L + IDLE・M + CPU・N (3)

各記号の意味は、次の測定データおよび係数である。
・LINK：TCP コネクション確立数
・IO　：ディスク負荷
・IDEL：CPU アイドル状態
・CPU　：ロードアベレージ
・K-N：重み係数

プロトタイプシステムでは、TCP コネクション確立数、CPU アイドル状態、ロードアベレージを有効にした。重み係数は、TCP コネクション確立数が 0.2、CPU アイドル状態が 0.3、ロードアベレージが 0.5 である。

4. 性能評価

本章では、作成したプロトタイプシステムを使って実施した予備実験の結果、及び、システムテストの結果について述べる。予備実験では、本システムで計測したネットワーク情報がどの程度有効なのかを調査するために実施した。

4.1 ネットワーク情報の評価

(1) 予備実験の目的および概要

実際のインターネット上のサイトを使ったテストによって、「最適サイト決定」のための「ネットワーク状態値」がどの程度有効性（的中率）を持っているのか調査することを目的とした。本実験では、データ転送時間をネットワーク的な距離として考える。つまり、データ転送時間が短いほど、ネットワーク的には近いと考える。ここでいうデータ転送時間は、クライアントが Web アクセスしたとき Web サーバからクライアントへのデータ転送に要した時間で、クライアント<->Web サーバ（サイト）間のコネクション接続時間を計測して求める。このとき同時にクライアント<->Web サーバ間のネットワーク状態を計測しデータ転送時間とネットワーク状態の相関を調べる。

(2) 計測方法

クライアント<->サイト間におけるデータ転送時間は、tcpdump コマンドで計測し、ネットワーク状態は作成した NS サーバ、NS-P を使って計測する。

実験の手順（図 6）は、1) Web ページのトップページに画像（イメージファイル）を 2 つ埋め込む。画像サイズは 2,525 バイトである。2) この画像の実体を、片方は US サイトにおき、もう片方は JPN サイトにおく（トップページではリンクする）。3) クライアントがトップページにアクセスすると、画像ファイルはリンクになっているため、クライアントのブラウザは US サイトと JPN サイトへ画像ファイルを取得にいく。4) ここでクライアントと 2 つのサイト間のコネクション接続時間（図 7）を計測する。その結果コネクション接続時間の短い方が、データ転送時間が短かくネットワーク的に近いということになる[6]。

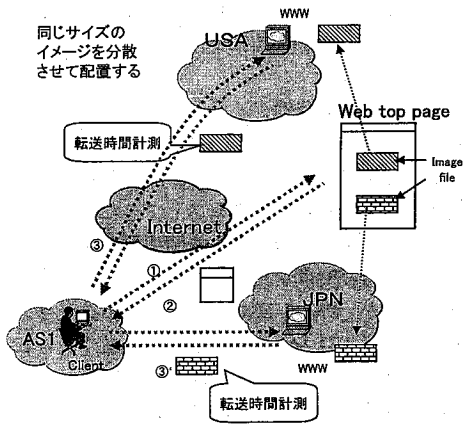

図 6　同じサイズのイメージを分散させて配置する

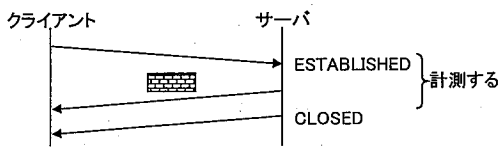

図 7　コネクション接続時間を計測する

今回、NSS、NS-P によって計測（図 8）したネットワーク状態は、クライアントまでの RTT、スループット、ルータのホップ数、AS パス長である。

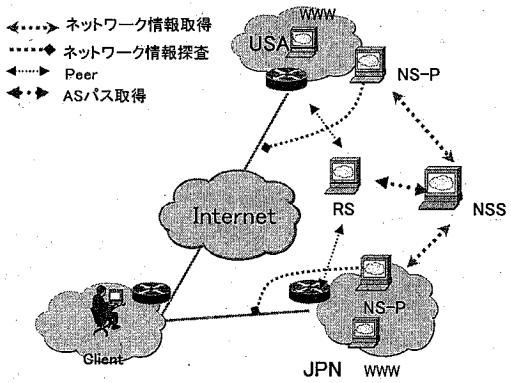

図 8　ネットワーク情報探査

コネクション接続時間の計測は、JPN サイト、USA サイトのそれぞれの Web サーバ上で tcpdump コマンドを実行し、Web アクセスに関するログを収集して行う。

今回の予備実験にあたって留意した点は、

・広範囲な地域からアクセスされる Web サーバサイトを選択
・日本、米国ともに同一のマシン仕様
・埋め込む 2 つの画像サイズは同じ大きさ
・時間帯、曜日のネットワークパターンを調査できるように任意の 1 週間を連続してデータを取得
・各サーバの内部 LAN における影響が無いことである

(3) データ評価の条件

NSS で取得したネットワーク情報の各データとコネクションの接続時間とを比較し、NSS の判断が正しいか判定する。判定は、アクセスしてきたクライアント毎に JPN と USA のネットワーク情報測定結果およびコネクション接続時間を比較する。具体的には、ネットワーク情報が JPN ＜ USA のとき、コネクション接続時間が JPN＜ USA なら判断は正しい（的中）とし、逆にコネクション接続時間が JPN≦USA になっていた場合、判断は間違いとする。ただし TP(スループット)に関しては値が大きいほど転送時間は短くなるため判定基準は逆になる。詳細は、次のとおりである。クライアント<->サイト（USA, JPN）間のネットワーク状態計測値（NET-STAT）とデータ転送時間（DATA-T）を比較し、

条件 1) NET-STAT(USA) ＞ NET-STAT(JPN)
　　DATA-T(USA) ＞ DATA-T(JPN) ・・的中
　　DATA-T(USA) ≦ DATA-T(JPN) ・・不正解

条件 2) NET-STAT(USA) ＜ NET-STAT(JPN)
　　DATA-T(USA) ＜ DATA-T(JPN) ・・的中
　　DATA-T(USA) ≧ DATA-T(JPN) ・・不正解

条件 3) NET-STAT(USA) ＝ NET-STAT(JPN)
　　DATA-T(USA) ＝ DATA-T(JPN) ・・的中
　　DATA-T(USA) ≠ DATA-T(JPN) ・・不正解

ネットワーク計測値：　AS, RT, RTT, TP（TP は判断が逆）と判断する。

有効データに関しては以下の条件を取り入れた。

・コネクションの接続時間
　マイナスの場合と 30 秒以上の場合は除く。
・AS パス
　等しい場合は除く。

また、2 回目の実験に関して 2 つの AS パスの長さのうち短い AS パスの長さ毎に正解率を集計した。

・RT（ルータホップ数）
　両サイトからクライアントまで到達した場合のみ有効にする。
・RTT
　両サイトからクライアントまで到達した場合のみ有効にする。
・TP（スループット）
　両サイトからクライアントまで到達した場合のみ有効にする。

Apache の httpd.conf ファイルを下記のように設定した。

KeepAlive　　Off

実験データの比較は小数点第 3 位を切り捨てて行う。

実験は 2 回行った。1 回目の実験では、日本サイトを横浜に置いた。2 回目の実験では、日本サイトを大手町に置いた。Web サーバのマシンスペックを表 4 に示す。

表 4　Web サーバのマシンスペック

1 回目）

	JPN	USA
OS	FreeBSD 2.2.7	FreeBSD 2.2.7
CPU	Pentium Celeron 333MHz	Pentium II 333MHz
MEM	160MB	256MB
HD	4GB	4GB

2 回目）

	JPN, USA
OS	FreeBSD 3.2
CPU	Pentium III 450MHz
MEM	256MB
HD	12GB

(4) 結果

結果は以下のようになった。

表5 ネットワーク情報の正解率

1回目）5日間：データ総計 21008件

	平均正解率	サンプル件数
AS	53.4%	14605
RT	48.9%	5727
RTT	71.1%	6743
TP	67.8%	6028

2回目）8日間：データ総計 39474件

	平均正解率	サンプル件数
AS	52.2%	27365
RT	48.9%	10250
RTT	71.5%	12666
TP	66.2%	9958

また、2回目の実験のなかで、ASパスの長さ毎の正解率を出すと以下のグラフのようになった。

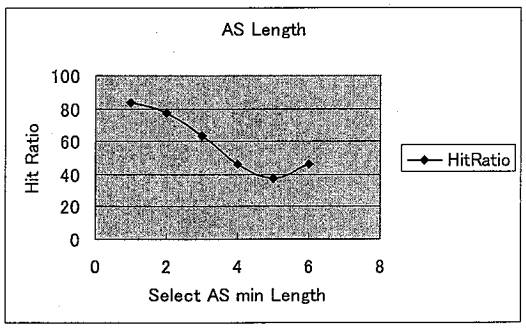

図9. ASパスの長さを考慮した正解率

以上より、経路情報（ASパス長）の長さを考慮しないで最短経路を割り出す方法では、約 50%の割合で最適な経路を選択。一方、RTT では、約70%の割合で最適な経路を選択した。

一方、ASパスの長さを考慮して、最適としたサイトからのASパスの長さが3以下の場合には、63%以上と有効な結果を得た。

この結果から、うまく両者のデータを利用すれば、最適な経路を短時間で取得できる。

4.2 システムテスト

今回作成したプロトタイプでは、ネットワーク状態値とサイト状態値の算出において利用するパラメータは AS パス長、ルータホップ数だけ有効にした。現時点でのパラメータの計算によって得られた最適サーバへのリダイレクト動作を確認できた。

5. 考察

5.1 パラメータについて

今回の予備実験ではネットワーク情報のパラメータ4つ（クライアントまでのRTT、スループット、ルータのホップ数、AS パス長）について有効性を検証した。2回実施した実験において、1回目、2回目の平均正解率をパラメータ毎に比較すると、ほぼ同じ（2%以内）結果であることからデータの信頼性は高いと考える。ただ、ASパスの長さは離散値であり実際に両サイトからの AS パスの長さが等しいというケースが多いということも分かった。一番低い正解率であったのは、ルータのホップ数であった。一般的にルータのホップ数が多いとルータ内でのオーバヘッドが影響し、ネットワーク的な距離は遠くなると考えられるが、今回の結果では、現在のインターネット（WAN）上において、ネットワーク（=回線）そのものの混雑による遅延の方がルータ内のオーバヘッドより上回っているため、ルータ内のオーバヘッドはそれほど大きく影響しないと推測できる。

なお、予備実験で検討したスループットについては67%程度の正解率となりRTTの結果を下回っている。スループットは、RTT より帯域の影響を受ける度合いが大きいという理由から、RTT の結果を上回ると推測していた。しかし、測定に使用したパケットサイズが小さい、また、発行のタイミングがよくないなどの理由で、正解率が上がらなかったと考えられる。今後、スループット計測方法においての検討が必要である。

ASパスによる最適サイトの判断は、非常にオーバヘッドの少ない方法であるため、ASパスの長さが短い場合には有効な判断子であるといえる。逆にASパスの長さが一定以上の場合はRTT等のほかの判断子と併用すべきであるということが分かった。

5.2 アルゴリズムについて

最適サーバを決定するための手順としてプロトタイプでは2ステップのアルゴリズムで決定した。最初のステップで最適サイトを決定し、次のステップで最適サーバを決定した。また最適サイトを決定するさいは、クライアントが初回アクセスかそうでないかで2つの方法で実施した。その結果、当初の想定通りの動作を確認でき、アクセスクライアントに対しては高速に応答する事ができた。今後は、計測パラメータをさらに吟味し、より精度の高い決定ができるように、システムの調整が必要である。

5.3 システムについて

最適サーバへの振り分けを HTTP リダイレクトによって実現した。HTTP リダイレクトはオーバヘッドが大きく、負荷がそれほど高くないシステムでは有効に機能する。しかし、負荷の高いシステムではオーバヘッドがボトルネックになる可能性がある。

6. 今後の課題とまとめ

本論文では、広域に分散配置された Web サーバにおいて、最適サーバを探索するためのシステムについて検討した。実際のインターネット上にサイトを構築(日本と米国に計測用の Web サーバサイトを構築) し、サーバとクライアント間のデータ転送時間および各ネットワーク状態値を計測し、これらの相関から最適な Web サーバ探索の手法を検討した。

ネットワーク状態の計測には、今回作成したプロトタイプシステム (NSS, NS-P) を使った。また、パラメータの一部を用いて、妥当性のあるサーバを算出、決定した最適サーバへリダイレクトすることの動作をシステム的に確認できた。

今後の課題としては、

- 予備実験で調査できなかったサイト内ネットワーク情報など、残りのパラメータの有効性について調査
- 調査した結果からネットワーク情報とサイト内ネットワーク情報の重みのバランス (パラメータの調整) を再度検討
- AS パスの長さの中で、さらにプリペンドの情報を省くことによる調査。

などがある。

また、今後はこれら残りのパラメータの調査を進めるとともに、パラメータの精度を上げ実用性を高めていく予定である。

謝辞　プロトタイプシステムの開発にあたり RS (zebra) の改良に協力していただいた (株) デジタル・マジック・ラボの石黒邦宏氏に感謝いたします。

参 考 文 献

1) ttcp - Enger, R., Reynolds, J., FYI on a Network Management Tool Catalog, RFC1470(1993)

2) netperf - http://www.netperf.org/netperf/NetperfPage.html

3) Apache modules - http://modules.apache.org/

4) Berners-Lee, T., Fielding, R. and Frystyk, H., Hypertext Transfer Protocol — HTTP/1.0, RFC1945(1996).

5) GNU Zebra - http://www.zebra.org/

6) Yutaka Nakamura, Ken-ichi Chinen, Suguru Yamaguchi, Hideki Sunahara:
"Proposal of WWW Server Behavior Observation Method by Packet Monitoring", In Proceedings of the Internet Conference 1998, December 1998.

7) W.Richard Stevens, TCP/IP Illustrated, Volume1:The Protocols. Addison-Wesley Publishing Company 1995

8) W.Richard Stevens, UNIX Network Programming, Prentice Hall, 1990

9) Bassam Halabi, Internet Routing Archtectures, Cisco Press/New Riders, 1997