

多様な選択ポリシーを利用可能なサーバ選択機構

下川 俊彦[†] 吉田 紀彦^{††} 牛島 和夫[†]

Server Selection Mechanism with Pluggable Selection Policies

Toshihiko SHIMOKAWA[†], Norihiko YOSHIDA^{††}, and Kazuo USHIJIMA[†]

あらまし インターネット上の多くのサービス提供者は、サービスの混雑を避けるために複数のサーバを用いてサービスを提供している。このような環境で、クライアントがどのサーバを選ぶかという問題が生じてきている。そこで、最適なサーバを自動的に選択するサーバ選択機構が必要となった。本論文では、まずサーバ選択機構への要求事項をまとめ、それらを満たす新たなサーバ選択機構を提案する。本機構は DNS (Domain Name System) を基盤とすることで幅広い適用性をもちつつ、多様なサーバ選択ポリシーを組み込み可能としたものである。これらのサーバ選択ポリシーは、ラウンドトリップタイムなどのネットワーク状態を用いた判断を行うことも可能である。動的に選択ポリシーの切替を可能とすることで、様々な状況に対応可能となる。我々は、プロトタイプシステムを用いて、広域に分散配置したサーバ群を対象に評価実験を行い、本サーバ選択機構の有効性を示した。

キーワード インターネット、サーバ選択、リクエストナビゲーション、選択ポリシー、広域分散システム

1. ま え が き

近年インターネットでは電子商取引や音楽配信をはじめとする様々なサービスが提供されるようになってきた。一方、利用者数も爆発的に増加している。この結果、サーバ計算機やそのネットワークの混雑という問題が表面化してきている。サーバ計算機やネットワークが混雑することで、サービスの反応時間が悪化し、サービスが停止する場合さえある。

この問題を解決するためには、多数のアクセスを処理することが可能な規模適応性のあるサーバを構築する必要がある。単一の計算機の性能には限界があり、また、その性能を拡張することも容易ではない。そこで、複数の計算機を用いて単一のサービスを構築することが多い。更に、それらを局所的に配置しては、ネットワークの混雑が解消しない。そこで、広域分散的に計算機を配置することが有効である。この場合、広域

分散配置した複数のサーバに対して、利用者からのアクセスをどのようにして振り分けるかが、規模適応性を実現するために重要である。見方を変えると、各利用者（あるいは各アクセス）にどうやってサーバを選択させるか、という問題である。

これを解決するために、何らかのサーバ選択機構が必要である。サーバ選択機構はリクエストナビゲーション機構とも呼ばれる。しかし、既存のサーバ選択機構は利用者に過度の負担を強いたり、柔軟性に欠ける。そこで、本研究では柔軟性の高いサーバ選択機構を開発し、規模適応性の高いサーバ構築を可能にすることを目指している。我々の提案方式は DNS (Domain Name System) を基盤として用いる。本方式は、多様なサーバ選択ポリシーを組み込むことができ、それらを動的に使い分けることが可能である。また我々の提案方式は、サービス提供者とは独立にサーバ選択が可能という特徴ももっている。

本論文は、以下、2. でサーバ選択機構への要求をまとめ、既存のサーバ選択機構を概観する。3. で我々が開発している多様な選択ポリシーを利用できるサーバ選択機構について述べる。4. で本機構のプロトタイプシステムに関して説明し、5. では本プロトタイプシステムを用いた実験とその評価について述べる。最後に

[†]九州大学大学院システム情報科学研究院, 福岡市

Graduate School of Information Science and Electrical Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka-shi, 812-8581 Japan

^{††}長崎大学工学部, 長崎市

Faculty of Engineering, Nagasaki University, 1-14 Bunkyo, Nagasaki-shi, 852-8521 Japan

6. で本論文のまとめを行う。

2. サーバ選択機構

本研究では、広域分散配置した計算機に対するサーバ選択機構が満たすことが望ましい性質について検討し、6 点にまとめた。本章でそれらについて述べ、更に既存のサーバ選択機構がそれらの性質を満たしているかを議論する。

2.1 サーバ選択機構への要求

本研究でまとめた、サーバ選択機構が満たすことが望ましい性質を以下に述べる。

2.1.1 規模適応性

1. で述べたとおり、サーバ選択機構は規模適応性のあるサーバ構築のための要素技術の一つである。よって、本機構が規模適応性の障害となつてはならない。

2.1.2 利用者透明性

規模適応性のあるサーバ構築の目的の一つは、利用者の利便性を増すことである。よって、サーバ選択機構を取り入れることによって利用者の利便性が減少してはいけない。そのためには、利用者に対してサーバ選択機構の存在を意識させないことが重要である。

2.1.3 サービス非依存性

構築するサーバが対象とするサービスには様々なものがある。代表的なものとしては、WWW、マルチメディアのストリーミングがある。したがって、サーバ選択機構はサービスに対して非依存であることが望ましい。

2.1.4 実装非依存性

サービスには複数の実装があることが一般的である。例えば、WWW サーバとしてはフリーソフトウェアの apache や商用ソフトウェアの Microsoft Internet Information Server など様々な実装が利用されている。サーバ選択機構の利用に当たって、実装の選択範囲を狭めるべきではない。したがって、サーバ選択機構は実装に対しても非依存である方が望ましい。

2.1.5 サービス提供者非依存性

サーバ選択を行うのはサービス提供者のみとは限らない。例えば、ある組織のネットワーク管理者が、その組織のネットワーク構成（対外接続線の種類など）に応じてサーバ選択を行うこともある。よって、サービス提供者とは独立してサーバ選択機構が利用できた方が、より適用可能範囲が広がる。

2.1.6 選択ポリシーの柔軟性

サーバ選択の判断基準はサービスの内容など様々な

要因に依存する。本論文ではこの判断基準のことを選択ポリシーと呼ぶ。選択ポリシーは画一的なものではなく、サービス提供者あるいは利用者によって異なる。よってサーバ選択機構は、様々な選択ポリシーを取り扱うことが可能であり、更にそれらを柔軟に切り換えられることが望ましい。

2.2 既存のサーバ選択機構及び関連研究

本節では、既存のサーバ選択機構について述べる。比較としてサーバ選択機構を用いないサーバ選択についても触れる。更に、これらが前節でまとめた性質を満たしているかを検討する。本節の最後で、関連研究として局所的に配置した計算機を対象としたサーバ選択機構について述べる。

現在、広く用いられているものの一つが DNS ラウンドロビンである。DNS とはインターネットでの名前解決に広く利用されているシステムである。最近の DNS サーバが備えるラウンドロビン選択機能を用いてサーバ選択が可能である。この機能は、あるホスト名に対して複数の IP アドレスを登録しておく、それらを順に（ラウンドロビンで）返答する。この結果、すべてのサーバが平均的に選択されることになる。

DNS は分散的に動作するため、この機能がボトルネックになる可能性は低い。よって、規模適応性に優れている。利用者は本方式が提供されていないにもかかわらず DNS を利用する。よって利用者透明性にも問題がない。現在のインターネット上のほぼすべてのサービスは DNS を利用する。よってサービス非依存性もよい。また、インターネット上のソフトウェアのほぼすべてが DNS を利用する。よって実装非依存性にも優れている。

ただし、選択ポリシーについては欠点がある。ラウンドロビンによる単純な順選択しか利用できない。利用者からはサーバはランダムに選択されているように見える。例えば、複数化されたサーバの一つが利用者の近傍にあったとしても、そのサーバが選択されるとは限らない。この結果、利用者はサーバが複数化された恩恵を十分に受けることができない。また、一般的には DNS への IP アドレスの登録はサービス提供者が行うので、サービス提供者に依存する。

重み付けラウンドロビン方式 [1] は、DNS ラウンドロビンに選択ポリシーを導入する試みである。この手法では、サーバごとに処理能力に応じた数の IP アドレスを割り当て、擬似的にサーバの処理能力を反映できるようにしている。この方法は、運用方法の変更の

みで利用できるという利点がある。しかし、基本的にはラウンドロビン選択であるため、選択ポリシーの柔軟性はほとんど改善されない。

SWEB [2] も、DNS ラウンドロビンの改良である。本方式ではサーバ選択は 2 段階で行われる。まず、DNS サーバがラウンドロビンでサーバ選択を行う。次に、前段階で選択されたサーバが他のサーバの状態を判断して、最終的なサーバを選択しリクエストを転送する。本方式は、固定的な選択ポリシーの導入には成功している。しかし、HTTP でしか利用することができずサービス非依存性に欠ける。また、選択ポリシーの柔軟性も不十分である。

Cluster DNS [3] も DNS サーバにサーバ選択機構を組み込んだものである。しかし、選択ポリシーの柔軟性に欠け、またサービス提供者と独立にサーバ選択を行う場合を考慮していないため、サービス提供者非依存性にも欠ける。

Smart Clients [4] では、専用のクライアントを用いて、クライアント側でサーバ選択を行う。本方式は Java applet を用いて、サーバの情報をクライアントに伝える。この方式はサーバ・クライアント側双方の実装を変更する必要があり、実装非依存性に欠ける。

重近らの方法 [5] では、複数化したサーバ群に同一 IP アドレスを割り当てておき、ネットワークの経路制御によるサーバ選択を行う。インターネット上のすべてのパケットは、経路制御機構の影響を受けるので適用可能範囲は非常に広い。しかし、選択ポリシーと経路制御のためのポリシーとを分離することができないため、選択ポリシーの柔軟性に欠ける。

DistributedDirector [6] は広域負荷分散を行うための商用システムである。商用サービスでは広く使われている。本方式を用いるためには、周辺のルータにも同じシステムを組み込む必要があり、実装依存性に欠ける。

AKAMAI [7] は、世界中に広域分散的に計算機を配置した商用のコンテンツ配信システムである。広域分散的なシステムを構築することで規模適応性に優れている。サービス選択という側面から見ると、その基盤技術としては DNS を用いている。選択ポリシーについては、AKAMAI から提供されるものを使う必要があり、その柔軟性に欠ける。また、サービス提供者非依存性にも欠ける。

比較としてサーバ選択機構を用いない場合について述べる。この場合、利用者自らの判断でサーバを選択

する。本論文ではこの手法を利用者選択と呼ぶ。この手法を用いる場合、利用者に対して、複数化したサーバのホスト名すべてを周知しておく。例えば、新聞社などのニュースサイトでは複数のサーバで同一コンテンツを公開している場合がある。また、商用ソフト及びフリーソフトの配付サイトでは、ミラーサーバを用意しておき、ダウンロード元を利用者に選択させることがある。この手法は、明らかに利用者透明性がない。利用者は自らのホストが最適かを判断しなければならない。これは利用者に対して、本来不要であるはずの負担を強いることになり、著しく利便性に欠ける。また、サービス提供者としてもサーバの増減が困難である。増減するたびにそのことを利用者全員に周知しなければならない。したがって規模適応性にも問題がある。本方式の最大のメリットは、サービス提供者の初期導入時の手間の少なさだと考える。単に複数のホストを提供し、それを周知するだけでよい。しかし、利用者透明性と規模適応性に重大な欠陥がある。

最後に、関連研究として局所的に配置した計算機を対象としたサーバ選択機構について述べる。ACEDirector [8]、BIG-IP [9]、LSNAT [10] は NAT 技術を応用しネットワーク層やデータリンク層で処理の分散を行う。これらと本研究が対象とする広域的なサーバ選択機構とは補完関係にある。これらを用いて高性能化したクラスタを、更に広域分散配置することで、より規模適応性に優れたサーバの構築が可能となる。

3. 多様な選択ポリシーを利用可能なサーバ選択機構

前章で述べたとおり、既存のサーバ選択機構は、サーバ選択機構への要求事項のいずれかを満たすことができない。そこで我々はすべての要求を満たすサーバ選択機構 TENBIN (Tenbin is Experimental Nameserver for Balanced INternet) を提案してきている [11]。

TENBIN は DNS を基盤として用いる。選択ポリシーを柔軟に切り換えることができる仕組みを導入することで、DNS がもつ選択ポリシーの柔軟性に関する問題を解決した。また、問合せを受けた際に、DNS データベースに登録されている IP アドレスとは異なる IP アドレスを独自に回答する機能を付加したことで、サービス提供者非依存性を解決した。我々は以下の方針で TENBIN を設計した。

- (1) 複数の選択ポリシーを組み込み可能

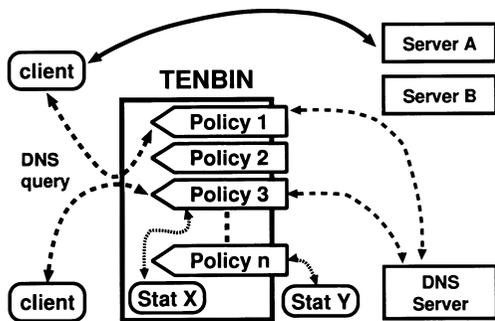


図 1 TENBIN の概念図
Fig.1 Concepts of TENBIN.

(2) 選択ポリシーを切り換えることが可能

(3) 既存のシステムに容易に組み込み可能

この方針を模式的に示したものを図 1 に示す。本章ではこれらについて説明する。

3.1 複数の選択ポリシー

TENBIN では、各選択ポリシーをプログラムとして記述する。この結果、多様な選択ポリシーを実現することが可能となる。選択ポリシーを記述したプログラムのことを、ポリシーモジュールと呼ぶ(図 1 中の Policy 1~n)。ポリシーモジュールを動的に組み込み可能な構造とし、複数の選択ポリシーの利用を容易にした。選択ポリシーの例としては以下のようなものがある。

(1) クライアント・サーバ間のラウンドトリップタイムが最短のものを選択

(2) クライアント・サーバ間のスループットが最大のものを選択

(3) クライアント・サーバ間の経路が最短のものを選択

(4) サーバの負荷が最も低いものを選択

3.2 ネットワーク情報収集

選択ポリシーの実現のために様々な情報を収集することが必要な場合がある。そのような情報を収集するために、各ポリシーモジュールが外部のモジュールと通信することを可能とした。本外部モジュールのことをネットワーク情報収集モジュールと呼ぶ(図 1 中の Stat X,Y)。図 1 中の Stat X が示すように、ネットワーク情報収集モジュールは TENBIN 中に組み込むことも可能である。

3.3 変更可能な選択ポリシー

TENBIN では、組み込んだ多数の選択ポリシーを

使い分けることが可能である。このために TENBIN への問合せごとに選択ポリシーを変更可能とした。

TENBIN はクライアントに対して DNS サーバとして機能する。このため、TENBIN への問合せは DNS サーバへの問合せと同一形式である。DNS サーバへの問合せには以下の情報が含まれる。

- 問合せホスト名
- 問合せ型
- 問合せクラス

問合せホスト名は FQDN (Fully Qualified Domain Name) である。問合せ型としては IPv4 アドレス問合せ (A 型), IPv6 アドレス問合せ (AAAA 型), メール配送先問合せ (MX 型) などがある。問合せクラスとしてはインターネットクラスなどがある。TENBIN では、この三つの情報の組合せすべてに対して選択ポリシーを使い分け可能である。例えば、同じ FQDN に対するインターネットクラスの IPv4 アドレス問合せと IPv6 アドレス問合せで異なる選択ポリシーを使うことができる。

3.4 既存のシステムへの組み込み

既存の DNS サーバは、問合せに答える機能と DNS のデータベースを管理する機能を併せもっている。TENBIN ではこれらのうち問合せに答える機能のみを提供する。ただし TENBIN は、問合せに答える際に選択ポリシーに応じた判断を行う。TENBIN は、DNS サーバの proxy サーバのように動作する。既存の DNS サーバ群は変更する必要がない。

利用者側で動作させた TENBIN によるサーバ選択を用いるには、クライアントが利用する DNS サーバを、既存のものから TENBIN に変更するだけで良い。近年のクライアントは、利用する DNS サーバを DHCP や PPP によって自動的に設定する場合が多い。したがって、多くの場合この変更は利用者から透明に行うことが可能である。サービス提供者側で TENBIN を動作させる場合、DNS の仕組み上、利用者がその存在を意識することは一切ない。

以上で述べたことから、TENBIN は既存システムへ容易に組み込むことが可能である。

4. プロトタイプシステムの実装

我々は TENBIN のプロトタイプシステムを実装している。本プロトタイプシステムのことを Tenbin と表記する。3. で述べた設計方針を実現するために、Tenbin は以下の構成になっている。

- (1) ポリシーモジュール群
- (2) ネットワーク情報収集モジュール群
- (3) 選択ポリシーデータベース
- (4) 問合せ応答部

ポリシーモジュールは独立したスクリプトとして記述する。記述言語としては、既存のスクリプト言語である Ruby を採用した。既存の言語を用いることで新たな言語を設計する必要がなくなった。4.1 でポリシーモジュールの例について述べる。

Tenbin に組み込むポリシーモジュールは、選択ポリシーデータベースに登録する。ポリシーモジュールの記述言語として採用した Ruby はオブジェクト指向言語であり、Tenbin では各ポリシーモジュールは各々が独立したクラスをもつ。選択ポリシーデータベースは、問合せホスト名、問合せクラス、問合せ型を文字列に変換後結合したものをキーとし、ポリシーモジュールのインスタンスを値とする単純なハッシュ表として構成した。ハッシュ表を用いたのは性能の確保と実装の容易さを重視したためである。本データベースとポリシーモジュール群を用いることにより、複数の選択ポリシーを組込み可能とし、またそれらを柔軟に切り換えることができる。

問合せ応答部は、リクエストを受け取るごとに新たなスレッドを起動し、選択ポリシーデータベースを用いて利用するポリシーモジュールを決定する。決定したポリシーモジュールに問合せを行い、サーバを選択する。この結果を問合せ元ホストに送り返す。本問合せ応答部が DNS サーバとして振る舞い、質問処理プロセスとして機能する。

4.1 ポリシーモジュールの例

ポリシーモジュールの例として具体的には次のような選択ポリシーを実現するものが考えられる。そこでこれらについて実装を行った。

- (1) 問合せ転送
- (2) 静的設定選択
- (3) ネットワーク情報選択

(1) は既存の DNS サーバに問合せを転送し、その回答をもって選択結果とする。現在は、本ポリシーをデフォルトの選択ポリシーとして用いている。

(2) は、Tenbin の管理者がポリシーモジュール内に選択結果を静的に記述しておき、その内容をもって選択結果とする。

(3) は 3.2 で述べた、ネットワーク情報収集モジュールからの情報を用いるものである。本モジュール

に関する詳細は次節で説明する。

4.2 ネットワーク情報選択

1. で述べたとおり、TENBIN の特徴の一つとしてサービス提供者と非依存にサーバ選択が行えることがある。そこでクライアント側から調査することが容易なネットワーク情報を用いるポリシーモジュールを実装した。具体的には、ネットワーク情報としてはクライアント・サーバ間のラウンドトリップタイムを用いた。

本ネットワーク情報収集モジュールはラウンドトリップタイムを計測するために、ICMP ECHO パケットを用いている。定期的に、すべての選択候補に対して ICMP ECHO REQUEST パケットを送信する。戻ってきた ICMP ECHO REPLY パケットからラウンドトリップタイムを計測し、これを一定回数繰り返す。平均値を計算しその候補へのラウンドトリップタイムとする。

すべての選択候補を得る方法としては二つの方法を実装した。DNS から情報を取得する方法と、手動で候補を指定する方法である。

(1) DNS ラウンドロビンを提供している場合、DNS サーバに問い合わせることですべての選択候補の IP アドレス群が得られる。この IP アドレス群を選択候補として用いる。

(2) 自動的に取得が不可能な場合には、手動でリストを作成する。利用者選択が提供されている場合の、複数化したホスト名の一覧がこれにあたる。一般的には「複数化したすべてのホスト名」を自動的に取得する方法はない。このため、Tenbin の管理者は「複数化したすべてのホスト名」のリストを維持する必要がある。この作業は、従来すべての利用者が必要だった作業である。それが管理者に集約されることになる。よって、この方法は利用者にとっては有効である。サーバ選択の対象となるサービスを限定すると、Tenbin の管理者が維持管理する必要があるリストの数を限定できる。このような場合には本方法は実用的である。

本ポリシーモジュールは、クライアントからの問合せのたびに、ネットワーク情報収集モジュールと通信し、最近の計測で最もラウンドトリップタイムが短い選択候補を知る。その結果を選択結果とする。

5. 実験と評価

プロトタイプシステム Tenbin を数箇所のネットワーク上で動かし評価を行った。その結果を示し検討を加える。

5.1 組み込みの容易性

Tenbin を、我々の研究室のネットワーク上で動かした。ネットワークの設定を動的に行うホストから利用するために、研究室内の DHCP サーバ並びに PPP サーバの設定を変更し、DNS サーバとして Tenbin を指示するようにした。ネットワークの設定を静的に行うホストに対しては研究室内の計算機管理者に依頼し、それらのホストが利用する DNS サーバを変更した。この結果、一般利用者はネットワークに Tenbin が組み込まれたことを意識せずに Tenbin のサービスを受けることができるようになった。以上によって、組み込みの容易性を実証的に確認した。

5.2 オーバヘッド

クライアントと DNS サーバの間に Tenbin が入ることによるオーバヘッドを、5.1 で導入した Tenbin のログを解析することで調査した。

Tenbin が受け付けた問合せのうち、問合せ転送ポリシーが選ばれた場合について、Tenbin の実行処理時間を計測した。その結果、Tenbin によるオーバヘッドは平均 9.1 ミリ秒であった。

このオーバヘッドと、実サービスを受ける時間を比較した。実サービスを受ける時間の例として、九州大学で全学向けに提供している web cache サーバの 2000 年 6 月 1 か月間のログを解析した。1500 万件のアクセスがありその平均時間は 3000 ミリ秒であった。この両者を比較すると、実サービスを受ける前に 9.1 ミリ秒のオーバヘッドが加わったとしても、これは無視できる範囲である。

5.3 サービス提供者に依存しないサーバ選択

本実験では、サービス提供者と非依存に行うサーバ選択の効果を調査した。そのために、選択ポリシーとしては 4.2 で述べたラウンドトリップタイム選択ポリシーを用いた。本実験では、ラウンドトリップタイム選択の対象として RingServer プロジェクト [12] の FTP サーバである ftp.ring.gr.jp. を用いた。このサーバを用いた理由は以下のとおりである。

(1) 本 FTP サーバは、20 台以上のサーバを、すべて異なるネットワークに広域分散配置している。これは広域分散配置されたサーバ群としては国内最大級のものである。

(2) 本 FTP サーバは、現在は、利用者選択と DNS ラウンドロビンによるサーバ選択しか提供しておらず、より良いサーバ選択機構を必要としている。

上記のうち (2) について説明する。本 FTP サーバ

は、利用者選択のために、すべてのサーバに異なった名前を付けている。例えば、etl.ring.gr.jp. や, asahi-net.ring.gr.jp. である。これらの一覧は WWW ページを通して周知されており、サーバの増減も本ページを通じて周知される。また DNS ラウンドロビンのために、ftp.ring.gr.jp. というホスト名に対して、これらすべてのサーバがもつ IP アドレスを登録してある。サーバの増減は ftp.ring.gr.jp. に登録する IP アドレスを変更することによって周知されるが、利用者はこのことを意識する必要はない。

本実験では ftp.ring.gr.jp. に登録されているすべての IP アドレスを選択候補として用いた。4.2 で述べた (1) の場合に相当する。

なお、実際の選択候補は IP アドレス群である。しかし、本論文においては可読性を高めるために、実験結果や考察においてはこれら IP アドレスを直接用いずに、これらの IP アドレスがもつそれぞれ異なったホスト名 (etl.ring.gr.jp. や asahi-net.ring.gr.jp.) を用いて議論を進める。

本実験では、5.1 で導入した Tenbin を用いて、2000 年 7 月 1 日より 2000 年 7 月 7 日までの間、約 30 分ごとに全選択候補に対するラウンドトリップタイムを計測し、サーバ選択を行わせた。

図 2 に、全選択候補あてラウンドトリップタイムの平均値 (average), so-net.ring.gr.jp. あてのラウンドトリップタイム (so-net), 選択されたホストのラウンドトリップタイム (selected) の変化を示す。また、それぞれの値の中間値は、average : 73.08 ミリ秒, so-net : 47.73 ミリ秒, selected : 46.17 ミリ秒であった。

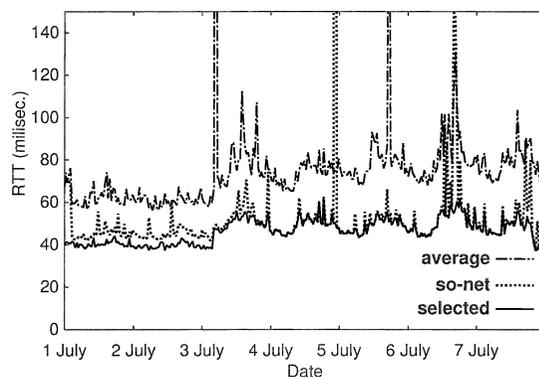


図 2 ラウンドトリップタイムの変化
Fig. 2 Change of round trip times.

本実験において、全選択候補あてラウンドトリップタイムの平均値は、2.2 で述べたラウンドロビン選択におけるラウンドトリップタイムの期待値である。

so-net.ring.gr.jp. は、本実験期間において、最も多くの回数にわたってラウンドトリップタイムが最小だったホストである。この結果は 2.2 で述べた利用者選択で最も良い選択を行えた場合の近似値と考えることができる。利用者選択においては、選択するサーバを頻繁に変更することは煩雑である。よってほとんどの利用者はある特定のものを常に選択する機会が多いと考えてよい。つまり、実測の結果最も多くの回数ラウンドトリップタイムが最小であったホストを選択していた場合が最善となる。

ラウンドトリップタイム選択モジュールは、各計測においてラウンドトリップタイムが最小であったホストを選択する。以下、本章ではこのホストのことを選択されたホストと呼ぶ。前段落の説明から明らかとなり、選択されたホストは多くの時間においては、so-net.ring.gr.jp. であった。図 2 からこのことが読み取れる。

図 2 及びこれらの値の中間値の比較からわかるように、ラウンドトリップタイム選択モジュールは、静的に so-net.ring.gr.jp. を選択した場合やラウンドロビン選択より良い結果を得ている。利用者選択やラウンドロビン選択のように単純な方式ではなく、ネットワーク情報を用いた選択ポリシーを用いることで、より良いサーバ選択が実現できることが明らかになった。また、選択候補が自動的に取得できる場合には、良いサービス提供者非依存性が得られることが確認できた。

5.4 多地点での実験

Tenbin を多地点の異なったネットワーク上で走らせ、その結果の違いを調査した。本実験では、5.3 と同様の計測を、国内 4 箇所で行った。ただし、実験期間は、1999 年 10 月 1 日より 1999 年 10 月 31 日までである。計測地によりばらつきがあるが、この間に約 880 回の選択を実行した。

九州大学 (表 1)、長崎大学 (表 2)、WIDE プロジェクト福岡 NOC (表 3) での結果をそれぞれの表に示す。これらの表は、選択された結果の上位 3 位までを示す。

九州大学 (表 1) と長崎大学 (表 2) は似た傾向を示していることがわかる。九州大学と長崎大学は、ネットワークポロジータ的には同一 AS (Autonomous System) に属し、非常に近いところに位置する。こ

表 1 九州大学で選択された割合
Table 1 Selected rate at Kyushu Univ.

ホスト名	割合 (%)
nacsis.ring.gr.jp.	51.1
ocn.ring.gr.jp.	20.7
etl.ring.gr.jp.	7.68

表 2 長崎大学で選択された割合
Table 2 Selected rate at Nagasaki Univ.

ホスト名	割合 (%)
nacsis.ring.gr.jp.	88.2
ocn.ring.gr.jp.	10.0
shibaura-it.ring.gr.jp.	0.53

表 3 WIDE プロジェクト福岡 NOC で選択された割合
Table 3 Selected rate at WIDE Project Fukuoka NOC.

ホスト名	割合 (%)
shibaura-it.ring.gr.jp.	20.0
asahi-net.ring.gr.jp.	16.4
omp.ring.gr.jp.	10.6
ocn.ring.gr.jp.	10.6

こで AS とは広域での動的経路制御に用いられている BGP (Border Gateway Protocol) における経路制御の単位である。選択結果の類似性は、このネットワークポロジータの類似性によるものと考えられる。

地理的には、九州大学 (表 1) と WIDE プロジェクト福岡 NOC (表 3) は同一キャンパス内に位置する。しかし、ネットワークポロジータ的には異なる AS に属する。この両者の選択割合は大きく異なっている。この差異はネットワークポロジータの差異によるものと考えられる。

これらの結果から、Tenbin は適切なネットワーク情報を用いた選択ポリシーを用いることで、各々が配置されたネットワークの状態を反映したサーバ選択を行えることが確認できた。

また、WIDE プロジェクト福岡 NOC (表 3) は他の二つと比べると、その選択結果にはっきりした傾向がない。これは Tenbin が、ネットワークポロジータの変化に自動的に追従したためであると考えている。

1999 年 10 月 14 日に、WIDE プロジェクトバックホーン網と国内で広く使われている AS 間接続地点の一つである NSPIX3 との接続関係が変更されたことがわかっている。すなわち、この時点でネットワークポロジータが変化したわけである。そこで、それ以前のデータについてまとめたものを表 4 に示す。これは他と同様、選択結果に一定の傾向を見ることができ

表 4 WIDE プロジェクト福岡 NOC で選択された割合 (1999 年 10 月 14 日 10 時以前)

Table 4 Selected rate at WIDE Project Fukuoka NOC. (before Oct. 14 1999 10:00)

ホスト名	割合 (%)
omp.ring.gr.jp.	58.3
ip-kyoto.ring.gr.jp.	22.1
htcn.ring.gr.jp.	11.0

この結果より、ネットワーク情報を用いた選択ポリシーは、ネットワークの状態（ここではネットワークポロジ）の変化を反映して、自動的にサーバ選択の結果を変化させることを確認することができた。

6. む す び

本論文では、サーバ選択システムへの要求事項を明らかにし、それを満たす柔軟性の高いサーバ選択機構である TENBIN を提案した。本機構のプロトタイプシステムを実装し評価を行い、要求事項を満たしていることを示した。

今後の課題としては、更に様々な選択ポリシーについて検討する必要がある。併せて各種のネットワーク情報の収集方法についても検討しなければならない。また、サービス提供者側でのサーバ選択についての実験・評価を行い、適用性の広さを確認する。

更に、様々な場合に適した選択ポリシーがどのようなものであるのかを検討する。

謝辞 本研究は、日本学術振興会の未来開拓学術研究推進事業（プロジェクト番号：JSPS-RFTF96P00603）による。

文 献

- [1] 馬場始三, 山口 英, “DNS を用いた広域負荷分散の実装,” 情処学研報, 98-DSM-9, pp.37-42, May 1998.
- [2] D. Andresen, T. Yang, V. Holmedahl, and O. Ibarra, “SWEB: Towards a Scalable WWW Server on MultiComputers,” Proc. 10th International Parallel Processing Symposium, pp.850-856, Hawaii, April 1996.
- [3] V. Cardellini, M. Colajanni, and P.S. Yu, “DNS Dispatching Algorithms with State Estimators for Scalable Web-server Clusters,” World Wide Web Journal, Baltzer Science, vol.2, no.3, pp.101-113, July 1999.
- [4] C. Yoshikawa, B. Chun, P. Eastham, A. Vahdat, T. Anderson, and D. Culler, “Using Smart Clients to Build Scalable Services,” USENIX '97, 13 pages, Anaheim, U.S.A., Jan. 1997.
- [5] 重近範行, 中村 修, 笹川信義, 村井 純, “長野オリンピックのネットワークと情報提供システム,” 情報処理, vol.39, no.2, pp.86-90, Feb. 1998.
- [6] CISCO Systems, Inc., DistributedDirector, <http://www.cisco.com/warp/public/cc/pd/cxsr/dd/>

- [7] Akamai Technologies, Inc., <http://www.akamai.com/>
- [8] Alteon Websystems, <http://www.alteonwebsystems.com/products/acedirector/index.asp>
- [9] F5 Networks, Inc., <http://www.f5.com/f5products/bigip/index.html>
- [10] X. Gan, T. Schroeder, S. Goddard, and B. Ramamurthy, “LSMAC vs. LSNTAT: Scalable Cluster-based Web Servers,” Cluster Comp.: J. Networks, Software Tools and Apps., pp.175-185, 2000.
- [11] T. Shimokawa, N. Yoshida, and K. Ushijima, “Flexible server selection using DNS,” International Workshop on Internet 2000, pp.A-76-A-81, Taipei, Taiwan, ROC, April 2000.
- [12] RingServer プロジェクト, <http://www.ring.gr.jp/> (平成 12 年 9 月 7 日受付, 13 年 1 月 22 日再受付)



下川 俊彦

1990 九大・工・情報卒。1992 同大学院工学研究科情報工学専攻修士課程了。同年(株)東芝入社。1997 九大大学院システム情報科学研究科情報工学専攻助手。2000 同大学院システム情報科学研究科情報工学部門助手。博士(情報科学)。広域分散協調処理、インターネットなどの研究に従事。情報処理学会会員。



吉田 紀彦

1981 東大大学院工学系研究科計数工学専攻修士課程了。(株)三菱総合研究所, 東京大学工学部, 九州大学工学部, 同大学院システム情報科学研究科を経て, 現在, 長崎大学工学部教授。1993~1994 スタンフォード大学客員研究員。工博。プログラミング方法論, 並列・分散・協調処理などの研究に従事。International Journal of Cooperative Information Systems 編集委員, IEEE-CS Asia-Pacific Activity 委員。情報処理学会, 日本ソフトウェア科学会, ACM, IEEE 各会員。



牛島 和夫 (正員)

1961 東大・工・応用物理(数理工学コース)卒。1963 同大学院修士課程了。同年九州大学中央計数施設勤務。1977 九州大学工学部情報工学科教授(計算機ソフトウェア講座担当), 1996 同大学院システム情報科学研究科(現・研究院)教授, 現在に至る。1990 年 4 月から 1994 年 3 月まで九州大学大型計算機センター長, 1996 から九州大学大学院システム情報科学研究科長(現・研究院長)を兼務。工博。情報処理学会, ソフトウェア科学会, ACM, IEEE-CS 各会員。