商用WWWサービスのIPv6への現実的な移行手法

河合栄治[†1] 白波瀬章[†2]
塚田清志[†3] 山口英[†4]

IPv6の普及が進むにつれ，商用WWWサービスのIPv6への移行が強く求められている．しかしながら，商用WWWサービスは，その高度なサービスの実現のためシステムが複雑であることが多く，さらにはサービス品質，費用対効果，セキュリティなどに非常に敏感であるため，IPv6用のシステムを別途に用意したり，既存のシステムをIPv4/IPv6デュアルスタック構成にするといった単純な方式では，IPv6への移行は実現が困難である．本研究では，商用WWWサービスのIPv6への現実的な移行を実現するリバースプロキシサーバを設計し，実装した．本システムで主に開発したのは，IPv6–IPv4中継機能，高い処理能力を達成するメモリキャッシュ，IPv6に未対応なサービスを一元的に管理するサービスフィルタである．また，本システムは，既存のIPv4ネットワークセキュリティのフレームワークへの組み込みも容易であるという特長も持つ．

Practical IPv6 Transition of Commercial WWW Services

Eiji Kawai,[†1] Akira Shirahase,[†2] Kiyoshi Tsukada[†3]
and Suguru Yamaguchi[†4]

A growing number of IPv6 users push up the demand for IPv6 transition of commercial WWW service. However, IPv6 transition of such service is often difficult because of the richness of the service and the complexity of the system. To make matters more complicated, commercial service is highly sensitive to service quality, cost performance, and security. Therefore, simple solutions such as whole system duplication for IPv6 or installation of IPv4/IPv6 dual stack to the system cannot get reality. In this study, we have designed and developed a reverse proxy server that enables practical IPv6 transition of the commercial WWW service. The key techniques of the system include IPv6–IPv4 relay mechanisms, an efficient memory-based cache algorithm especially for reverse proxies, and a flexible service filter that allows administrators to manage the service intensively for both IPv4 and IPv6. Our reverse proxy server also provides good conformity to the current IPv4 security framework.

1. はじめに

現在，インターネットは数多くの既存通信機構を統合する次世代通信基盤として認識されるようになり，いわゆるワークステーションやパーソナルコンピュータのようなコンピュータ機器だけでなく，携帯電話やPDAなどの情報端末，冷蔵庫や電子レンジといった家電製品などもインターネットに接続されるようになりつつある．そのため，インターネットに接続される機器の数が急激に増加し，現在主に利用されているインターネットプロトコルであるIPv4において，アドレス枯渇問題をはじめとする様々な問題が表面化してきている．

一方で，これらの問題については早くから指摘されており，アドレス空間を拡張したIPv6が開発されている．IPv6は，その拡張により実用上は無限に近い多数のアドレス割当てを可能にし，さらにはサービス品質保証機構のサポートやセキュリティ機能，モバイル通信機能などとの統合など，高機能なインターネットプロトコルとしても期待されている．また，IPv6はIPv4との共存を考慮した設計がなされている[1)]のも特長の1つである．そのため，小規模ネットワークではIPv6への全面移行が比較的容易であり，実例も報告されている[2)]．しかし，IPv6への移行を，範囲が限定された小規模ネットワークではなく，インターネット全体で実現しようとする場合，移行コストが大きな

†1 科学技術振興事業団さきがけ研究21
PREST, Japan Science and Technology Corporation
†2 エヌ・ティ・ティ・スマートコネクト株式会社
NTT SmartConnect Corp.
†3 株式会社毎日放送
Mainichi Broadcasting System, Inc.
†4 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

問題となる．そのため，IPv6 インターネットへの移行は，ある日突然実行できるようなものではなく[3]，現実の要求項目を満足させつつ段階的に進めていかなければならない．

本研究は，インターネットにおける広範なサービスのうち，商用 WWW サービスに焦点を当て，IPv6 への低コストで段階的な移行を可能にする技術の開発を目的とする．商用 WWW サービスの IPv6 への移行を実現するには，単に WWW サーバを IPv6 化するだけでは不十分であり，WWW サービス特有の現実的な問題を解決する必要がある．WWW は HTTP を通じて情報を配信するだけではなく，その他の多くのインターネットサービスのインタフェースとしての役割も担っており，付随する様々な問題の発生が予想される．具体的には，IPv6 未対応のサービスへのリンクにおけるエラーの発生や，IPv6 ホストの安易な導入によるネットワークのセキュリティ破壊などがあげられる．また，商用 WWW サービスでは，そのほかにも費用対効果やサービス品質などにも配慮が必要となる．

そこで本研究では，商用 WWW サービスの IPv6 への移行を実現するために，リバースプロキシサーバによるサーバサイドキャッシュ技術を用いて，システム変更を最小限におさえつつ既存の IPv4 環境の IPv6 への対応を実現した．本システムにより，従来の IPv4 向けサービスの品質およびセキュリティ保護のフレームワークを維持したまま，同時に IPv6 環境へのサービスも低コストで行うことが可能となる．また，一部の IPv6 未対応なサービスに対しても，リバースプロキシサーバにおいて一元的に管理し，エラーの発生を抑制することができる．

以下，本論文の構成を示す．まず 2 章では，WWW サービスの IPv6 移行に関する問題点について議論する．次に 3 章では，WWW サービスの IPv6 移行という観点から主な IPv6 移行技術を分類し，それぞれについて利点および欠点を考察する．4 章では，本研究で開発したリバースプロキシサーバの設計および実装について述べる．本プロキシサーバは，ベンチマークテストによる性能評価および実際の IPv6 ネットワークにおける実証実験を行った．5 章でそれらの実験について述べ，結果を考察する．最後に 6 章で本研究についてまとめ，今後の課題を述べる．

2. 商用 WWW サービスの IPv6 移行における課題

本研究で対象とする商用 WWW サービスを IPv6 に移行する際には，様々な課題が存在する．本章では，それらを WWW サービス特有の課題と商用サービス特有の課題とに切り分けて考察する．前者により，HTTP による通信を基本とする WWW サービスにおける技術的課題を明らかにする．また，後者により，商用サービス全般の IPv6 への移行において解決策が満足すべき条件を明らかにする．

2.1 WWW サービス特有の課題

WWW サービスを IPv6 へ移行する際に大きな問題となるのは，IPv6 未対応なサービスが混在してしまうことと，IPv6 セキュリティの保護が難しいことである．本節ではそれらについて述べる．

2.1.1 IPv6 未対応サービスの混在

WWW サービスを IPv6 へ移行するのに，単に WWW サーバを IPv6 に対応させても，場合によってはエラーが発生する．たとえば，WWW は CGI をはじめとする各種アプリケーションをサービスする目的でも使われているが，これらのアプリケーションが通信相手などの IP アドレスデータを用いている場合，IPv6 に対応していなければエラーが発生する可能性がある．さらに，WWW サービスの本質は様々なサービスを互いにリンクすることであることから，WWW は HTTP によるデータの転送だけでなく，他の HTTP 以外のプロトコルを用いたサービスのインタフェースとしても用いられている．そのため，IPv6 に未対応なサービスをリンクしている場合，IPv4 によるアクセスができないユーザはエラーに遭遇する可能性がある．この問題を解決するには，エラーが発生しないように IPv6 用のコンテンツを変更する必要がある．すなわち，エラーが発生するリンクを削除もしくは別のコンテンツに切り替えればよい．

2.1.2 IPv6 セキュリティ

現在の IPv6 ネットワークは，実験ネットワークとしてだけではなく商用サービスとしても利用可能になりつつある．その一方で，IPv6 ネットワークにおけるセキュリティについて考慮しているサイトは非常に少ない．現在主流である IPv4 ネットワークにおけるセキュリティは，ファイアウォールなどの装置によって包括的に保護する傾向が強い．そのような環境において，セキュリティ対策を十分とらないまま安易に IPv6 接続が可能なホストを導入すると，そのホストを通じて既存ネットワークのセキュリティが脅かされる恐れがある．

セキュリティ対策が進まない原因の 1 つに，高速通信に耐えうるだけの性能を持った IPv6 ファイアウォール製品がまだないことがあげられる．IPv6 に対応してい

るUnixなどを用いてIPv6ルータを構築し，ファイアウォール機能を設定することも可能ではあるが，大規模ネットワークでは性能に不安が生じる．また，IPv6ではその広いアドレス空間を生かした終端間（End-to-End）通信の広範な実現が期待されており，ファイアウォール利用の是非が議論されている．こうした事情も問題を複雑にしている要因の1つである．

2.2 商用サービス特有の課題

商用サービスをIPv6へ移行する場合，実験サービスにおけるIPv6対応の際には問題とならなかった様々な条件を満足する必要がある．本節では，それらの要件のうち最も重要なものとして，IPv6 WWWサービスのIPv4向けサービスからの隔離およびコストパフォーマンスについて述べる．

2.2.1 IPv6 WWWサービスの隔離

商用サービスにおいては，サービス品質が非常に重要である．特にWWWサービスは，様々な情報を提供するプラットフォームとして現在利用されており，そのサービス品質はサービス提供者（個人および法人）の評価に直接結び付く重要なものと認識されている．そのため，現行のIPv4サービスとは隔離した形でIPv6を導入するのが望ましい．これは，IPv6サービスを論理的だけではなく物理的にも隔離することを意味する．これは，万が一IPv6サービスに障害が発生したとしても，既存のIPv4サービスにはいっさい影響を与えずに問題の分析および復旧ができなければならないことから必要である．

2.2.2 低コスト・高パフォーマンス

商用サービスでは，コストパフォーマンスに対する非常に高い意識が求められる．まずコストについては，サーバコスト，ネットワークコスト，セキュリティコスト，コンテンツ管理コストなどが含まれる．これらのコストが高い手法は現実的ではない．一方で，パフォーマンスについては，従来から様々な技術がその向上のために投入されてきた．キャッシュ技術や負荷分散技術などはその一例である．商用サービスにおける移行を実現するためには，これらの技術との干渉が小さく，整合性の高い手法が望ましい．

3. WWWサービスにおけるIPv6移行のフレームワーク

本章では，WWWサービスのIPv6移行のフレームワークについて考察する．IPv6 WWWサービスを提供する場合，下記の手法が考えられる．

- IPv4/IPv6デュアルスタックサーバ
- IPv6専用サーバ
- IPv6–IPv4リバースプロキシサーバ

本章では，2章における議論をふまえ，それぞれについて利点欠点を考察する．

3.1 IPv4/IPv6デュアルスタックサーバ

本手法は，同一のホストでIPv4およびIPv6の両方をサービスするものである．IPv6をサポートするためには，オペレーティングシステムおよびサーバアプリケーションの両方でIPv6に対応する必要がある．現在，IPv6はLinuxや各種BSD，Solaris，Windows 2000/XPなど多くのオペレーティングシステムでサポートされている．また，最もよく用いられているWWWサーバであるApacheにおいても，IPv6はサポートされている．そのため，これらを用いているサイトでは，サーバホストのカーネルをIPv6対応のものに変更し，IPv6ネットワークに接続すれば，IPv6クライアントからのリクエストを処理することが可能となる．本手法は，これまで用いているサーバホストでIPv4およびIPv6の両方を用いるためサーバコストが低いという利点がある．

このように，本手法はIPv6への移行が原理的には容易であるかのように見えるが，多くの問題を含んでいる．まず，現在IPv6対応が完了していないプラットフォームを利用しているサイトでは，プラットフォームの移行が必要となる．データベースなどのバックグラウンドサーバへの接続を含む複雑なアプリケーションが動作しているようなサイトでは，このプラットフォームの移行が容易でない場合が多い．次に，先にあげたIPv6サービスの隔離という点で問題がある．何らかの問題が発生した場合に，問題を切り分けることができなくなる恐れがある．また，IPv6未対応のサービスにおけるエラーを未然に防ぐためにIPv6用コンテンツを用意しなければならず，結局のところ，仮想ホスト機能などを用いて，論理的にサービスを分離しなければならない．このことはコンテンツ管理コストの高騰を招く恐れがある．ネットワークセキュリティの観点からも，本手法は問題を含んでいる．それまでファイアウォールによって守られていたIPv4ホストが，IPv6経由では自由にアクセス可能になる恐れがある．最後に，負荷分散装置との併用が難しいという問題もある．現在の商用WWWサービスでは，その多大なアクセスを処理するためにレイヤ4/7スイッチなどの負荷分散装置を用いることが多いが，これらがIPv6に対応しておらず，併用が難しい．

3.2 IPv6専用のサーバ

本手法は，IPv6専用のサーバを別途用意し，システムを複製することでIPv6サービスを構築するもの

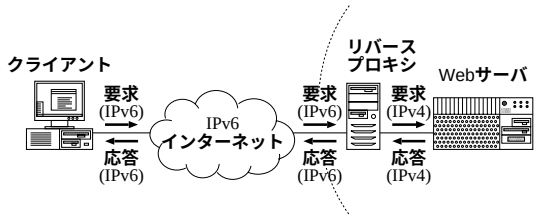

図 1 IPv6 リバースプロキシサーバ
Fig. 1 IPv6 reverse proxy server.

である．本手法の利点は，IPv4 と IPv6 のサービスの分離が可能であることである．一方で，別途サーバを用意しなければならないことから，サーバコストが高いという欠点があげられる．また，デュアルスタックサーバの場合と同様に，IPv6 未対応のサービスにおけるエラーを未然に防ぐために IPv6 用コンテンツを用意しなければならず，その管理コストは高い．

3.3 IPv6–IPv4 リバースプロキシサーバ

IPv6 のリクエストを受信し，IPv4 のリクエストに変換して IPv4 サーバに中継するリバースプロキシサーバ（サーバアクセラレータとも呼ばれる）を別途用意することにより，IPv6 サービスを実現することができる（**図 1**）．リバースプロキシサーバは，クライアントから IPv6 リクエストを受け取り，オリジンサーバに IPv4 によるリクエストの中継を行う．本手法は以下にあげるような様々な利点を有している．

まず，IPv6 専用のサーバを用いる場合と同様，従来の IPv4 サービス用サーバとは別のホストを利用するため，IPv4 サービスと IPv6 サービスの分離が可能である．次に，プロキシサーバにおいてキャッシュ技術を用いることにより単体で高い性能が得られるのも利点の 1 つである．一般的に，リバースプロキシサーバは高いヒット率を達成することが可能であり[4]，既存の IPv4 サービスへの影響を小さくすることができる．これは，高いヒット率により，オリジンサーバへ転送されるリクエストを少なくすることができるからである．この高い単体性能という利点は，IPv6 対応の負荷分散装置がほとんどない現時点では特に重要である．また，従来の IPv4 サービスのフレームワークをそのまま用いることができる点も利点の 1 つである．これにより，IPv4 における負荷分散装置やファイアウォールなどによるセキュリティ保護のフレームワークをそのまま活用した形で IPv6 サービスを組み込むことが可能になる．さらに，リバースプロキシサーバはコンテンツ複製にともなう管理コストを軽減する．これは，リバースプロキシサーバが自律的にオリジンサーバからコンテンツを取得するからである．

4. IPv6 リバースプロキシサーバの設計と実装

本研究では，商用 WWW サービスの IPv6 への移行における諸問題を解決するため，リバースプロキシサーバによる IPv6–IPv4 中継システムを開発した．本章では，その設計および実装について述べる．

4.1 設　　計

本システムの主な技術開発点は，リバースプロキシサーバにおける IPv6–IPv4 中継機能，リバースプロキシサーバにおけるメモリベースのキャッシュに特化したオブジェクト置換アルゴリズム，サービスフィルタによる柔軟かつ集約的な IPv6 用コンテンツ管理機構の 3 点である．本節では，それぞれについて述べる．

4.1.1 リバースプロキシサーバにおける IPv6–IPv4 中継

リバースプロキシサーバにおいて IPv6–IPv4 中継を行うことによって，IPv4 用 WWW サーバを用いた IPv6 環境への WWW サービスの提供を可能にする．クライアントからのアクセスの具体的な処理手順は次のとおりである．クライアントから送られてきた IPv6 によるアクセスに対して，可能な場合はプロキシサーバのキャッシュから直接コンテンツを応答する．また，オリジンサーバへのアクセスが必要な場合には，プロキシサーバは IPv4 によりオリジンサーバからコンテンツをいったん受け取り，キャッシュに格納しながら IPv6 によりクライアントに転送する．この手順により，IPv6 環境からの IPv4 WWW サービスへの透過的なアクセス手段を実現する．本手法は，3.3 節で述べた様々な利点を有している．

4.1.2 高性能メモリキャッシュ

近年，WWW サーバにおいて高い性能を達成するために，メモリベースサーバが開発されている[5]．メモリベースサーバとは，ファイル I/O を削減するために，サーバプロセス用のロックされた（ページアウトしない）メモリ領域を確保し，独自のファイルキャッシュとして用いるものである．従来のファイルシステムにおけるキャッシュ機構を用いることも可能であるが，I/O の対象となるファイルのサイズ分布やディレクトリ構造，アクセス頻度，生成/読み込み/書き込み/消去の分布などの特性が大きく異なるため，効率の良いキャッシングが行えない．本研究では，特にリバースプロキシサーバにおけるメモリキャッシュの効率を高めるオブジェクト置換アルゴリズムを開発した．

まず最初に考慮しなければならないのは，利用可能なメモリ容量および対象となるオブジェクトデータの

総容量である．一般的に，リバースプロキシサーバは，いわゆるユーザサイドプロキシサーバ（フォワードプロキシサーバ）と比較して，対象となるサービスコンテンツが限定され，データ総容量はたかだか数 GB 程度である．このことから，キャッシュ容量が十分に用意されたリバースプロキシサーバでは非常に高いヒット率が達成可能である．一方で，このデータ総容量とサーバに搭載されるメモリ容量とを比較すると，ほとんどの場合メモリ容量の方が小さい．そのうえ，このメモリは，オペレーティングシステムやバッファキャッシュ，他のプロセス，プロキシサーバ自身のバッファなどと共有されるため，オブジェクトデータのキャッシュとして用いることができる領域はさらに小さくなる．

実装が容易であることからオブジェクト置換アルゴリズムとしてよく用いられている LRU（Least Recently Used Algorithm）は，アクセス回数が多いコンテンツは短期間に繰り返しアクセスされるという仮定のもと，キャッシュを管理する．そのため，キャッシュ容量が制限された環境では，アクセス周期が非常に短いごく一部のオブジェクトしかキャッシュヒットせず，多くのオブジェクトがキャッシュされては捨てられるというサイクルに陥る．

一方で，LFU（Least Frequently Used Algorithm）は，各オブジェクトをアクセス回数でソートしておく必要があるため，LRU と比較して実装が複雑であり，用いられることが少ない．しかし，シミュレーションを行った結果，キャッシュ容量が小さい場合でも良好なキャッシュヒット率を示すことが判明した（**図 2**）．このシミュレーションでは，リクエスト分布には Zipf の法則，オブジェクトサイズには対数正規分布を用い[6]，1,300 万オブジェクトに対して 3,300 万リクエストを生成した．ここで完全 LFU とは，置き換えによりキャッシュから削除されたオブジェクトを含むすべてのオブジェクトについてアクセスカウントを保持するアルゴリズムであり，アクセスカウント用の記憶領域のコストを無視しているが，LFU としては最良のヒット率となる．また，オンキャッシュLFU とは，キャッシュされているオブジェクトに関してのみアクセスカウントを保持するものであり，現実的な実装である．LFU をリバースプロキシサーバに用いる場合の問題点は，ワーキングセットの変化に対応できないことである．これは，LFU がアクセス回数のみを考慮するため，一度頻繁にアクセスされると，その後アクセスされなくなってもオブジェクトがキャッシュにとどまるのが原因である．

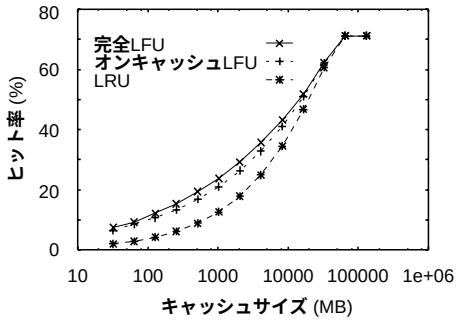

図 2 LRU と LFU のキャッシュシミュレーション結果
Fig. 2 Performance of LRU and LFU.

以上の考察から，本実装では LFU を基礎として，オブジェクトのエージング処理を行うことにより，効率的なキャッシュ管理を実現した．基本的なデータ構造としては，**図 3** に示したように，リンクリストの配列を用いることにより，擬似的な LFU を実現した．オブジェクトのエージングは，ガーベージコレクションを行う際に，各オブジェクトの最終アクセス時刻に，そのオブジェクトのアクセス回数によって線形に増加するオブジェクト保持時間を加えた時刻と，現時刻を比較することによって実現している．C 言語による疑似コードを**図 4** に示す．以後，本アルゴリズムをエージング LFU（A-LFU）と記す．

4.1.3 サービスフィルタ

現在の WWW サービスは，HTTP を用いたクライアントとサーバの間の通信だけではなく，WWW をインタフェースとして用いている HTTP 以外のプロトコルによるサービスを含めたものと考えられている．たとえば，ストリーミングサービスは WWW を

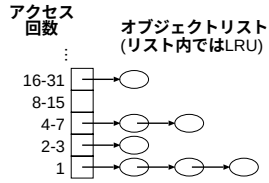

図 3 疑似 LFU のデータ構造
Fig. 3 Data structure of pseudo LFU.

```
for(;;){
  obj = gc_next_obj();
  if(obj->last_access + hold_min
     + hold_inc * obj->n_access < now)
    gc(obj);
}
```

図 4 エージング処理の疑似コード
Fig. 4 Pseudo C code for aging objects.

インタフェースとして提供されることが多いが，ストリーミングデータ自体は HTTP 以外の独自プロトコルによって提供される☆．このようなサービスは，リバースプロキシサーバによる HTTP 通信の中継だけでは IPv6 に対応できない．それらのサーバおよびクライアントを含むサービス全体を IPv6 に対応する必要があるからである．しかし，現在様々なサービスが IPv6 への対応を進めている段階であり，IPv6 対応が完了しているサービスはまだ少数にとどまっているのが現状である．このような状況で WWW だけ IPv6 に対応しても，IPv6 に対応していないサービスにおいてクライアントでエラーが発生する結果となり，商用 WWW サービスでは許容できない．

そこで本研究では，IPv6 に対応していないサービスのリストをプロキシサーバにおいて一括管理し，それらについてはクライアントに対して IPv6 に未対応のサービスであるという内容のメッセージを送信する機構を開発した．これにより，クライアントで発生するエラーを防ぐことができる．具体的には，IPv6 未対応のサービスに対するフィルタを用意し，そのようなサービスに対するリクエストを捕捉し，ユーザへの説明を表示するページにリクエストを転送するようクライアントに通知する．フィルタには，削除したいサービスの URL をファイル拡張子もしくはその完全パス名で指定する．また，削除した URL のそれぞれについて，未対応であるというメッセージを伝える代替ページの URL を指定することができる．ユーザへの通知には，HTTP 応答におけるステータスコードである “`301 Moved Temporarily`” によるリクエストのリダイレクションを用いた．

4.2 実　　装

これまでに筆者らは，全国高校野球選手権大会のインターネット中継実験において，新しく開発したリバースプロキシサーバ Chamomile を用いた WWW システムを運用し，多くの知見を得てきた[7]．本システムは，その Chamomile を基に 4.1 節で述べた各種機能を追加している．

Chamomile の実装は，pthread ライブラリを用いたマルチスレッドアーキテクチャを採用し，性能を重視した設計になっている．特に，SMP ホスト上で運用した場合に性能のスケーラビリティを確保するよう，各種 thread の生成数はホストの構成および負荷の状況に応じて柔軟に設定可能となっている．一方で，移植性にも配慮した実装を行っており，現在 Linux，FreeBSD，Solaris 上での動作を確認している．

☆ HTTP 以外の通信を遮断しているファイアウォール環境のために，HTTP を用いるストリーミングサービスも存在する．

5. 性能評価および運用

Chamomile の基本的な性能評価として，プロキシベンチマークによる実験を行った．また，本システムを実際の IPv6 ネットワーク上で運用し，実証実験を行った．本章では，それらの結果についてまとめる．

5.1 Web Polygraph によるベンチマークテスト

商用 WWW サービスでは高いサービス品質が求められるため，本方式で採用したリバースプロキシサーバは高い性能を有していなければならない．本研究では，開発したリバースプロキシサーバ Chamomile の基本性能を総合的に評価するために，プロキシベンチマーク Web Polygraph[8] を用いた．本節では，実験の概要およびその結果についてまとめる．

5.1.1 リバースプロキシ用ワークロード WebAxe-4

Web Polygraph では，WebAxe-4 と呼ばれるワークロードを用いてリバースプロキシの性能を評価することができる．WebAxe-4 によるテストでは，クライアントホスト群およびサーバホスト群を用い，スイッチに接続されたプロキシサーバに対して負荷を与え，性能を計測する．**図 5** に用いた機器群およびそれらのネットワーク構成を示す．

WebAxe-4 では，現実的な環境を再現するために，ネットワーク環境やワークロードなどに関して詳細な条件が定められている．その中で重要なのが遅延時間およびワーキングセットである．遅延時間については，クライアント側のネットワークインタフェースにおいて，dummynet[9] を用いて平均 40 ミリ秒の遅延時間および 0.05%のパケットロス率を設定するように定められている．さらに，オリジンサーバにおいて，プロキシサーバからリクエストを受け取って応答を返すまで平均 300 ミリ秒のサービス遅延が Polygraph のサーバプロセス側で設定される．また，サーバプロセスにおけるコンテンツのワーキングセットのサイズは

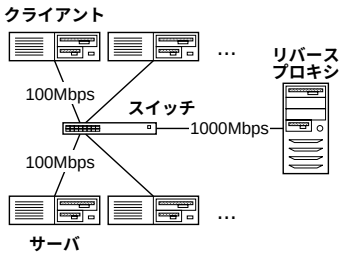

図 5 WebAxe-4 の機器およびネットワーク構成
Fig. 5 Hosts and networks for WebAxe-4 tests.

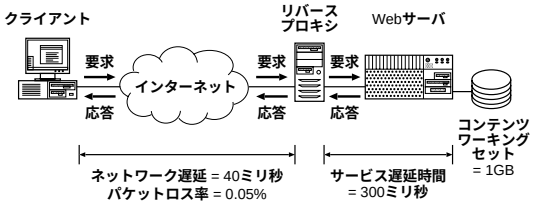

図 6 WebAxe-4 おけるネットワーク環境
Fig. 6 Emulated network environment for WebAxe-4 tests.

表 1 Polygraph によるテストで用いた機器
Table 1 Equipments used for Polygraph tests.

サーバ	Pentium III 866 MHz，メモリ 512 MB
クライアント	Pentium III 1.4 GHz，メモリ 512 MB
プロキシ	Pentium III 800 MHz × 2，メモリ 2 GB
スイッチ	3 Com SuperStack3 4900 (1000 baseT × 12)

1 GB に設定され，コンテンツタイプに応じた時間経過にともなう更新も考慮されている．たとえば，HTML オブジェクトは平均 1 週間で更新され，画像オブジェクトは平均 1 カ月で更新されるよう設定される．さらに，CGI に代表されるような動的なコンテンツも考慮されており，総リクエストの約 20%に対してキャッシュ不可となるよう設定される．以上より，**図 6** に示すような環境を想定した負荷実験が実現される．

ここで，ネットワーク環境に関して注意すべき点が 1 つある．それは，Polygraph は IPv4 ネットワーク上でのベンチマークテストのみをサポートしており，IPv6 ネットワーク上での性能を計測することができない点である．しかし，本研究の目的は，高性能なリバースプロキシの開発であり，高性能な IPv6 プロトコルスタックの開発ではない．また，リバースプロキシにおけるカーネル内部の IPv4 および IPv6 の処理では複雑なルーティングやオプションの処理は不要であり，ネットワークアドレスの取扱いが若干異なる以外は同様の処理を行う．以上より，IPv4 ネットワーク上での計測で検証可能であると判断した．

5.1.2 A-LFU と LRU の性能比較

まず，本研究で開発したメモリキャッシュアルゴリズムの性能を評価するために，オブジェクトキャッシュ領域としてのメモリ使用量を変化させながら応答時間およびヒット率の変化を計測した．ここでは，Chamomile のスループットの計測が実験目的ではないため，リクエストレートを Chamomile が十分余裕を持って処理が可能な 1,000 リクエスト/秒に設定した．また，比較対象としては LRU を用いた．なお，実験に用いた機材は**表 1** のとおりである．

表 2 および**表 3** は，A-LFU および LRU における

表 2 エージング LFU の性能（スループット：1,000 リクエスト/秒）
Table 2 Performance of A-LFU (request rate: 1,000 req/s).

メモリ量	128	256	512	1024
応答時間：平均	412.96	352.07	331.55	326.81
応答時間：ミス	557.77	569.12	579.72	587.22
応答時間：ヒット	219.44	218.52	220.07	223.61
ヒット率	45.00	65.13	72.45	75.15
バイトヒット率	33.18	49.08	55.57	59.44
提供ヒット率	77.30	77.26	77.30	77.27
提供バイトヒット率	63.97	64.09	63.89	64.13
正規化ヒット率	58.21	84.30	93.73	97.26
正規化バイトヒット率	51.87	76.58	86.98	92.69

メモリ量は MB，時間はミリ秒，率は百分率

表 3 LRU の性能（スループット：1,000 リクエスト/秒）
Table 3 Performance of LRU (request rate: 1,000 req/s).

メモリ量	128	256	512	1024
応答時間：平均	423.19	367.58	334.94	321.32
応答時間：ミス	554.63	561.11	571.03	575.69
応答時間：ヒット	220.90	217.92	218.42	219.89
ヒット率	41.41	59.32	70.33	75.01
バイトヒット率	30.68	44.72	54.08	59.51
提供ヒット率	77.11	77.29	77.24	77.36
提供バイトヒット率	63.85	64.13	64.16	64.31
正規化ヒット率	53.70	76.75	91.05	96.96
正規化バイトヒット率	48.05	69.73	84.29	92.54

メモリ量は MB，時間はミリ秒，率は百分率

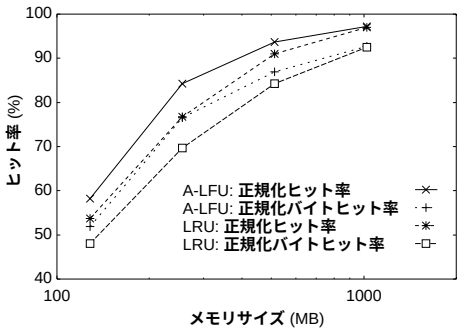

図 7 A-LFU および LRU の正規化ヒット率の比較
Fig. 7 Comparison between normalized hit ratios of A-LFU and LRU.

測定結果である．Polygraph の実験は厳密には再現性がなく，各実験で Polygraph から供されるリクエスト列に対するキャッシュ容量を無視した最大ヒット率（提供ヒット率および提供バイトヒット率と呼ぶ）が若干異なる．そのため，この最大ヒット率によって実際のヒット率を正規化し（正規化ヒット率および正規化バイトヒット率と呼ぶ），評価指標として用いることにする．**図 7** は，これらの正規化したヒット率を A-LFU および LRU で比較したグラフである．この

表 4 Chamomile のスループット性能
Table 4 Throughput performance of Chamomile.

リクエストレート	応答時間平均	応答時間ミス	応答時間ヒット
1000	326.81	587.22	223.61
1200	330.51	593.37	225.90
1400	334.79	600.08	228.85
1600	341.02	610.31	233.09
1800	354.29	634.03	242.83
2000	406.48	706.47	286.11
2200	10301.26	14460.52	8674.46

リクエストレートはリクエスト/秒
応答時間はミリ秒

グラフから，A-LFU は LRU と比較して特にメモリ資源が限られている場合に良い性能を有していることが分かる．

5.1.3 Chamomile のスループットに関する評価

次に，Chamomile が商用 WWW サービスに十分なスループット性能を有しているか評価するために，リクエストレートを変化させながら応答時間の変化を計測した．本実験では，Chamomile がオブジェクトをキャッシュするメモリ領域として 1 GB を確保している．リクエストレートを毎秒 1,000 から 2,200 まで 200 きざみで変化させた際の結果を**表 4** に示す．この結果から，Chamomile はおよそ毎秒 1,800 リクエストから 2,000 リクエスト程度の処理能力があることが分かる．毎秒 2,000 リクエストの負荷を与えた場合，クライアントとプロキシの間のトラヒック量は平均 92.48 Mbps であった．このことから，Chamomile は非常に高いスループット性能を有していることが分かる．

5.2 IPv6 ネットワークにおける実運用

本研究では，実証実験として Chamomile を 2001 年 12 月に開催された Net.Liferium2001 [10] において実運用を行った．**図 8** に，本運用におけるネットワーク構成を示す．

まず，IPv6 ネットワークに接続されたクライアントから，実際の商用 WWW サイトである毎日放送の IPv4 サービス用の WWW サーバを Chamomile 経由でアクセスし，リバースプロキシサーバにおける IPv6–IPv4 中継機能が正しく動作していることを確認した．今回は，動画像によるストリームサービスがまだ IPv6 に対応していなかったため，サービスフィルタによるリクエストのナビゲーションを行った．これについても，正しく動作していることを確認した．

6. まとめと今後の課題

本研究では，商用 WWW サービスの IPv6 への移行を実現するために，リバースプロキシサーバ Chamomile を開発した．これにより，システム変更を最小限におさえつつ既存の IPv4 環境の IPv6 への移行を実現した．本システムの利点は，以下のようにまとめることができる．

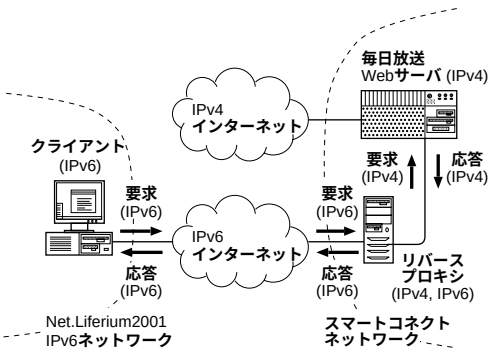

図 8 Net.Liferium2001 における実験ネットワーク
Fig. 8 Network for Net.Liferium2001 exhibition.

- IPv6 の WWW サービスを IPv4 サービスと分離した形態で提供することができる．
- キャッシュ技術により高い単体性能を達成することができる．
- 従来の IPv4 負荷分散装置やファイアウォールなどとの併用が容易である．
- IPv6 未対応なサービスを Chamomile において一元的に管理でき，コンテンツ管理コストが低い．

このように，Chamomile は既存の IPv4 WWW サービスの IPv6 化を容易にしたが，実際の運用を通じてさらなる課題も見つかっている．その中で最も重要なものとして，複数のサイトにまたがって構成される WWW サービスの包括的なサポートがあげられる．ここでは，構成する WWW サイトが単一の組織内に収まっている場合とそうでない場合に分けて考える．

前者の例としては，各部署ごとに仮想ホスト機能を利用してサーバを立ち上げたり，実際に個別にサーバを設置したりすることがあげられる．この場合は，複数のプロキシサーバを立ち上げることによって解決することができるが，Chamomile におけるメモリキャッシュの容量の制約などから難しい場合もある．そのため，リバースプロキシにおける仮想ホスト機能を実現し，仮想リバースプロキシホスト間でキャッシュメモリ領域を共有する必要があるだろう．

後者の例としては，WWW 広告のような，外部の WWW サービスを導入することがあげられる．外部 WWW サイトが IPv6 に対応していない場合，エラー

が発生する可能性がある．この場合，同一組織内のWWW サービスの場合と異なり，他組織の WWW サイトのリバースプロキシを勝手に立ち上げることは問題がある．そのため，WWW コンテンツにおけるリンクそのものを直接書き換える機能が必要となるだろう．

謝辞 本研究における運用実験でご協力いただいたサイバー関西プロジェクト（CKP）に感謝いたします．

参 考 文 献

1) Gilligan, R. and Nordmark, E.: Transition Mechanisms for IPv6 Hosts and Routers, RFC 2893 (2000). http://www.ietf.org/rfc/rfc2893.txt
2) Kimoto, M., Suenaga, H., Kimura, T. and Ohno, H.: Construction of IPv6 Small Network Using INS router, *Proc. Inet 2000*, Yokohama, Japan (2000).
3) Carmès, E.: The Transition to IPv6, ISOC MEMBER BRIEFING #6 (2002). http://www.isoc.org/briefings/006/
4) Luotonen, A.: *Web Proxy Servers*, Prentice-Hall (1998).
5) Joubert, P., King, R.B., Neves, R., Russinovich, M. and Tracy, J.M.: High-Performance Memory-Based Web Servers: Kernel and User-Space Performance, *Proc. 2001 USENIX Annual Technical Conference*, Boston, Massachusetts, USA (2001).
6) Barford, P. and Crovella, M.: Generating Representative Web Workloads for Network and Server Performance Evaluation, *Proc. ACM SIGMETRICS '98* (1998).
7) 西馬一郎，河合栄治，知念賢一，吉田豊一，香取啓志，山口 英：WWW クラスタにおける同期を考慮したコンテンツ更新機構の設計と実装—夏の高校野球 WWW 中継における運用，インターネットコンファレンス (2001).
8) The Measurement Factory: Web Polygraph. http://www.web-polygraph.org/
9) Rizzo, L.: Dummynet: A simple approach to the evaluation of network protocols, *ACM Computer Communication Review*, Vol.27, No.1 (1997).
10) Net.Liferium 実行委員会：Net.Liferium 2001 (2001). http://www.key3media.co.jp/net-life/2001/

(平成 14 年 5 月 2 日受付)
(平成 14 年 11 月 5 日採録)

河合 栄治（正会員）

1996 年京都大学理学部数学科卒業．1998 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．2001 年同大学同研究科博士後期課程修了．2000 年 10 月より，科学技術振興事業団さきがけ研究 21「機能と構成」領域研究員．分散計算環境の構築，インターネットにおける大規模情報配信システムの開発，高速 I/O 指向オペレーティングシステムの研究に従事．博士（工学）．

白波瀬 章

1992 年 3 月京都大学大学院工学研究科博士前期課程修了．同年 4 月日本電信電話株式会社入社．2000 年 3 月より NTT スマートコネクト株式会社に出向中．同社にてデータセンタ事業におけるビジネス開発業務に従事．大規模インターネット情報配信システムの構築・運用技術，Diff-Serv をベースとした QoS システムの研究等に従事．

塚田 清志

1979 年大阪大学工学部通信工学科卒業．同年株式会社毎日放送入社．撮像現業部門を経て送出部門へ移り 1990 年新社屋移転時テレビ送出設計を担当，技術開発部門を経てインターネット等マルチメディア部門および社内情報ネットワークの開発を担当．サイバー関西プロジェクトの活動に参加し通信と放送の融合点での技術開発を推進．現在，同社メディア開発局デジタルセンター長．

山口 英（正会員）

1990 年 10 月大阪大学大学院基礎工学研究科情報工学専攻博士後期課程を中退し，大阪大学情報処理教育センター助手として着任．1992 年 10 月奈良先端科学技術大学院大学情報科学センター助教授．1993 年 4 月より，同大学情報科学研究科助教授．2000 年 4 月より，同大学情報科学研究科教授．大規模分散処理環境構築，ネットワークセキュリティ等の研究を行う．また，WIDE Project のメンバーとして，広域コンピュータネットワークの構築・研究に従事する．工学博士．