

## 移動透過 IP マルチキャストに対応する グローバルライブマイグレーションの設計と性能評価

鎌田 恵介<sup>†1</sup> 近堂 徹<sup>†2</sup>  
西村 浩二<sup>†2</sup> 相原 玲二<sup>†2</sup>

仮想化技術の発展により、規模や負荷に応じて仮想計算機の構成を動的に変更できるライブマイグレーションが注目されている。また、同一ネットワーク内に限定されているライブマイグレーションを拡張し、IP 移動透過性をサポートしたグローバルライブマイグレーションも提案されている。しかし既存方式では、その対象が IP ユニキャスト通信のみであり、IP マルチキャストの移動透過性まで実現できていない。そこで本研究では、仮想計算機上でマルチキャストとユニキャストを組み合わせて利用する移動透過 IP マルチキャスト方式について提案し、プロトタイプシステムを用いた基本性能評価について述べる。評価結果より、ネットワークセグメントが異なるマイグレーションにおいても、同一ネットワーク内のマイグレーションと同等の途絶時間に短縮できることを確認した。

### Design and Evaluation of Global Live Migration with mobility support for IP multicast

KEISUKE KAMADA,<sup>†1</sup> TOHRU KONDO,<sup>†2</sup> KOUJI NISHIMURA<sup>†2</sup>  
and REIJI AIBARA<sup>†2</sup>

Virtualization technologies are widely used, and include Live Migration which changes resource allocation for virtual machines according to scale or load. In addition, Global Live Migration with IP mobility is also proposed. It enables migration among distributed sites and provides continuation even if the network of the virtual machines was changed. However, it supports only IP unicast communication, not IP multicast communication. In this paper, we propose a mobility support mechanism for IP multicast on virtual machines, and evaluate its basic performance using prototype system. As a result, the proposed method provides migration function continuously receiving multicast stream stopped same as traditional Live Migration.

#### 1. はじめに

インターネットユーザの増加に伴い、ネットワークトラフィックは日々増加している。アクセスが集中するサーバに対しては、並列化・冗長化による処理速度の向上が行われてきた。しかし、サーバが物理的に遠距離にある場合など、伝送速度や途中経路のルータ処理による遅延がレスポンスを悪化させている。

これを解消する方法として、近年 CDN (Contents Delivery Network) と呼ばれる配信網が利用されている。CDN は、予めアクセスが集中しやすい大容量コンテンツを世界中のキャッシュサーバに分散配置しておき、リクエストしたユーザに直近のキャッシュサーバか

らデータを送信することによって、レスポンスの向上と負荷の集中を解消している。これらの通信方式としてはユニキャストが使用されることが多いが、CDN のような管理されたネットワークにおいてはマルチキャストを利用することで、網内の帯域効率化を図ることが出来る。さらに、マルチキャストアドレスによってコンテンツを識別し、IGMP<sup>1)</sup>、MLD<sup>2)</sup>、PIM<sup>3)</sup>などのプロトコルにより IP 層で配送ツリーを構築してパケットが中継されるため、動的なキャッシュノードの配置に対してもスケーラビリティを確保できる。

一方で近年、仮想化技術の発展により各種サーバを仮想計算機上に構築する仮想化技術が注目されている。仮想化技術の特徴として、セッション状態などアプリケーションの内部状態を保持したまま仮想計算機を任意の実計算機に移動させることが可能である。しかしながら、ネットワークセグメントが異なる広域ネットワーク環境でこれを展開しようとする場合、IP 層での

<sup>†1</sup> 広島大学大学院工学研究科  
Graduate School of Engineering, Hiroshima University  
<sup>†2</sup> 広島大学情報メディア教育研究センター  
Information Media Center, Hiroshima University

移動透過性をサポートしなければ通信の継続ができない<sup>4)</sup>。この問題に対してグローバルライブマイグレーション<sup>5)</sup>が提案されており、セッションを引き継いだまま動的な配置転換ができるものの、ユニキャスト通信のみ(ユニキャストモビリティ)に限定されておりマルチキャストに対する移動透過性(マルチキャストモビリティ)は維持されない。マルチキャストの移動透過性を実現するには、移行先ネットワークにおけるマルチキャストツリーの再構築を早期に完了させることと、移行先ネットワークがユニキャストに限定されたエリアの場合への対応が必須となる。

そこで本論文では、仮想計算機(以下、VM; Virtual Machine)がグローバルライブマイグレーションを行う際、VMを稼働させる実計算機(以下、VMS; Virtual Machine Server)がマルチキャスト受信の困難なネットワーク上に設置されているケースを考慮した、ユニキャスト・マルチキャスト両対応のマイグレーションを提供する手法を設計する。具体的にはユニキャストについて、既存のIPモビリティのうち経路最適化に優れたMAT<sup>6)</sup>を用い、マルチキャストについては、VMS単位に設置するエージェントノードがMLDの先行送信とユニキャスト中継処理を自律的に行う。これにより、VMの移行先ネットワークのマルチキャスト対応状況に依らず、VM上のアプリケーションは継続的なストリーム受信が可能となる。

以下に、本論文の残りの構成を示す。2章は関連技術について概説し、3章は提案手法の設計について、4章は実装について述べる。そして、5章ではマルチキャストマイグレーションにおける性能評価と考察を行う。最後に、6章でまとめと今後の課題について述べる。

## 2. 関連研究

### 2.1 仮想計算機を用いたCDNの構築

実計算機を用いて構成された既存のCDNでは、各サーバの負荷や帯域に応じた動的な構成変更は困難であった。この問題を、VMを用いることによって解決にした手法<sup>7)8)</sup>が提案されており、VMの構成を動的に変更しながらノードを追加・削除することで柔軟な運用が可能である。しかし、VM上で動作するアプリケーションは実計算機と同様に独立しているため、例えば高負荷時にVMが追加されたとしても、アプリケーションがセッションを切り替えない限り負荷を軽減できないという問題点がある。

### 2.2 仮想計算機とグローバルライブマイグレーション

2.1節で述べた問題を解決する方法として、ライブマイグレーションが挙げられる。ライブマイグレーションは、VMのメモリ情報をVMS間で移行させることにより、VM内のアプリケーションセッションを保つ

たまま、最小限の停止時間で動的な計算資源の変更が可能となる。しかし、移行後のネットワークアドレス設定についてはサポートされていない。したがって、異なるサブネットへマイグレーションを行うと到達性が無くなり、アプリケーション層で確立されたセッションが切断されてしまうため、マイグレーションは同一サブネット内のみという制限があった。

この問題に対し、IPモビリティと複数インタフェースを用いたグローバルライブマイグレーション<sup>5)</sup>が提案されているが、提案手法ではユニキャスト通信に限定される。VMがマルチキャストを受信している際にグローバルライブマイグレーションを行った時、移行先ネットワークに同一のマルチキャストストリームが配送されていない場合にはVM内のマルチキャストアプリケーションに通信途絶が発生してしまう。途絶時間はネットワークによって異なるが、マルチキャストストリームを受信するためには、受信ノードがマルチキャストアドレスをMLDを直近のルータと交換し、そのルータはさらに上位のルータとPIM<sup>3)</sup>等を使ってマルチキャストツリーを構築する必要がある。マルチキャストアプリケーションが受信状態を変更する場合以外にも、ノードのマルチキャスト受信状態を報告するMulticast Listener Report(以下、MLD Report)が送信されるが、その間隔はルータが定期的に送信するMulticast Listener Query(以下、MLD Query)と等しい。MLD Queryの送信間隔はデフォルト値が125秒と定義<sup>2)</sup>されており、マルチキャストツリーの構築には一般的に数秒から数分程度必要である。すなわち、継続的なマルチキャスト受信のためには、VMがマイグレーションを行う前に、VMが受信中のマルチキャストに関するツリーを移行先ネットワークで構築しなければならない。

## 3. 移動透過マルチキャストに対応するグローバルライブマイグレーション

### 3.1 提案手法の考え方

マルチキャストモビリティに対応するグローバルライブマイグレーションを実現するには、次の二点が重要である。

一つ目は、グローバルライブマイグレーションが完了するまでに移行先ネットワークでマルチキャストが受信可能となっていることである。したがって、マイグレーション時点で該当VMがJoinするマルチキャストを把握し、移行先ネットワークのルータに対してもMLD Reportを通知しておく必要がある。MLDはリンクローカルなプロトコルであり、該当VMはマイグレーションが完了するまで移行先ネットワークに接続されないことから、他のノードが代理でJoinを行う。

二つ目は、移行先ネットワークにおいてマルチキャ

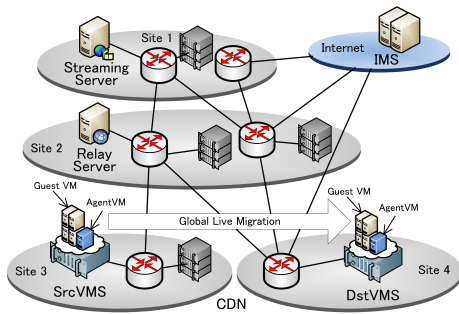


図 1 ネットワーク構成例

Fig. 1 An example of network structure.

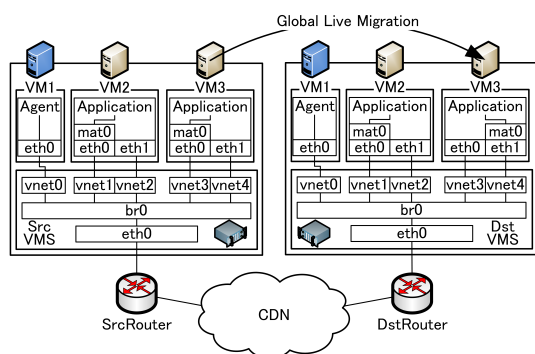


図 2 VMS 内部の構成

Fig. 2 Internal structure of VMS.

ストが受信できない場合に、ユニキャストを用いた伝送方式に切り替えた上で、さらに VM にそれを隠蔽することである。ユニキャスト中継を VM に直接行った場合、VM ではマルチキャストを直接受信する場合とは異なる宛先アドレスとして受信されるため、その対応関係をアプリケーションが管理しなければならない。さらに、多数の VM が中継のために同一のユニキャストセッションを複数確立して輻輳を発生させないように、ユニキャスト中継を VMS 単位で管理し、セッション数を受信ノード数に比例させないように工夫する必要がある。

これらの要素技術として、筆者らが提案しているユニキャストを併用する移動透過 IP マルチキャスト<sup>9)</sup>を用い、それを仮想化環境に拡張することにより実現する。

### 3.2 システムの全体構成

提案手法では、図 1 に示すように VMS と異なるネットワークにマルチキャストの送信ノードが存在し、VMS 上の Guest VM (以下、VM) が Join するというモデルを想定する。そして、各 VMS に管理エージェント (以下、Agent) を収容する Agent VM を設置する。Agent は、同一 VMS 内の全ての VM のマルチキャストモビリティを維持するために 3.4 節で示した動作を行う。本論文では、SrcVMS (Source VMS; マ

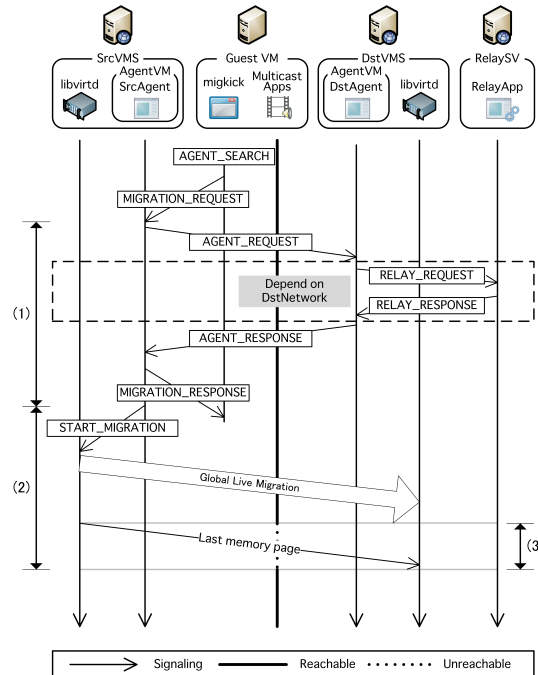


図 3 提案手法によるマルチキャストマイグレーションの流れ

Fig. 3 Sequence on proposed multicast migration method.

イグレーション前の VMS) 上の Agent を SrcAgent, DstVMS (Destination VMS; マイグレーション後の VMS) 上の Agent を DstAgent と定義する。

VMS 内部のネットワーク構成は、図 2 に示すようなブリッジ構成を基本とする。VM 内のユニキャストアプリケーションは、ルーティングテーブルにより後述する移動透過アーキテクチャ MAT の仮想インタフェース mat0 に一旦集約され、現在グローバル接続が可能なインタフェースから送受信される。一方、マルチキャストアプリケーションは mat0 に対して Join を行い、ユニキャスト通信に使用されるインタフェースと同一のインタフェースから送受信される。VM 上のインタフェース eth0 や eth1 は、VMS 上では vnet0 といった TAP デバイスと 1 対 1 に対応し、VMS 上のブリッジデバイス br0 を経由して物理インタフェース eth0 へと接続される。

### 3.3 基本動作概要

マルチキャストに対応したグローバルライブマイグレーションでは、マルチキャストとユニキャストそれぞれが独立した手順でモビリティの確保を行う。その各シーケンスについて、マルチキャストモビリティに関する対応を 3.3.1 節に、ユニキャストモビリティに関する対応を 3.3.2 節に述べる。

#### 3.3.1 マルチキャストモビリティ

提案手法におけるマルチキャストマイグレーション

の流れを図 3<sup>\*1</sup>に示す。VM には migkick というマイグレーションアプリケーションを用意する。migkick が起動点となり、SrcAgent に対してマイグレーション要求を送信したり、移行に必要な情報を受信する役割を担う。マルチキャストモビリティの実現のために定義したメッセージを表 1 に示す。

まず、マイグレーション対象の VM が現在受信しているマルチキャストを、移行先ネットワークでも継続的に受信できるようにするため、migkick が SrcVMS の Agent を探索 (AGENT\_SEARCH) し、マイグレーション要求 (MIGRATION\_REQUEST) を送信する。

次に、マイグレーション要求を受信した Agent は、DstVMS に対しマルチキャストの代理 Join と移行先ネットワークプレフィックス情報 (プレフィックス及びプレフィックス長) を要求 (AGENT\_REQUEST) する。マルチキャストが移行先ネットワークで受信可能である場合、代理 Join を行ってからストリームが受信できた時点で応答を返す事が可能だが、逆に受信できないと判断するにはタイムアウトを設定して判定するしかない<sup>9)</sup> ゆえに、一定時間の間にストリームが受信できるかどうかを確認し、その可否とプレフィックス情報を応答 (AGENT\_RESPONSE) する。もし、受信できないと判断した場合には、ユニキャスト中継が可能な中継サーバに対し、中継の依頼 (RELAY\_REQUEST, RELAY\_RESPONSE) を行う。そして、Agent が代理でユニキャストを受信し、マルチキャストとして再送信を行う。ユニキャスト中継に関しては、3.4.3 節に後述する。

その後、Agent は要求を出した VM に対して先ほどのプレフィックス情報を応答 (MIGRATION\_RESPONSE) し、マイグレーションの準備が整った事を通知する。VM は、現在通信に使用していないインタフェース (初期状態であれば eth1) に対して、RA<sup>10)</sup> の受信を無効化した上で、新しいプレフィックスから自動生成されたアドレスを設定する。

以上の事前手順の追加により、移行先ネットワーク上での該当マルチキャスト受信と、移行先ネットワークで使用するユニキャストアドレスの準備が整った。

最後に、Agent がマイグレーションを SrcVMS に要求 (START\_MIGRATION) し、ライブマイグレーションが開始される。マイグレーションの最終段階では、最後のメモリページのコピーのために一時的に VM がサスペンドするが、その後は予め準備されたマルチキャストストリームをすぐ受信可能であり、マルチキャストの移動透過性が実現される。

### 3.3.2 ユニキャストモビリティ

提案手法ではユニキャストの移動透過性を実現するためのアーキテクチャとして MAT<sup>6)</sup> を使用し、文献<sup>5)</sup>と同様に複数インタフェース構成を利用し、VM 上で

\*1 図中の (1) ~ (3) については、5 章の評価実験において参照する。

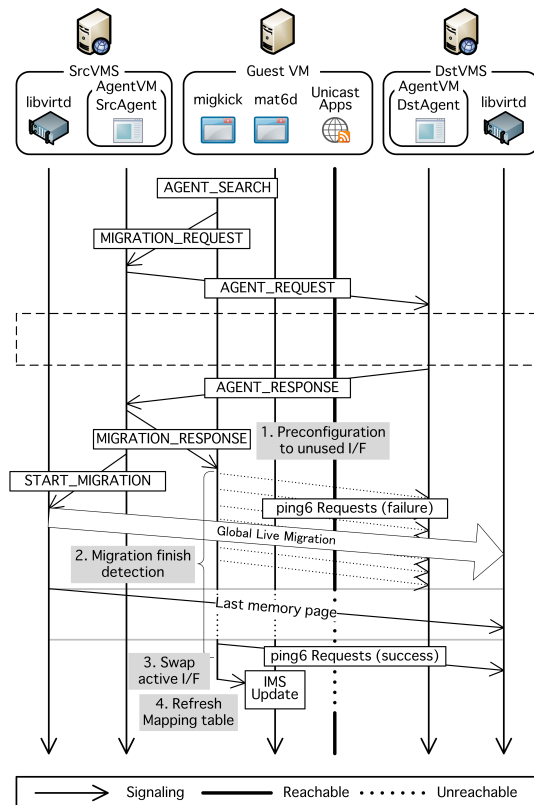


図 4 提案手法によるユニキャストマイグレーションの流れ  
Fig. 4 Sequence on proposed unicast migration method.

ハンドオーバーを行うことにより IP 層でのモビリティを確保する。

提案手法におけるユニキャストマイグレーションの流れを図 4 に示す。マルチキャストモビリティ同様に、migkick がマイグレーション起動点となる。マイグレーション前の VM が eth0 経由で通信しているとする<sup>\*2</sup>と、マイグレーションを行う前に移行先ネットワークの IP アドレス・ルーティングを eth1 に設定する。マイグレーション操作の最後でサスペンドされるまでは mat0 を通じて eth0 経由でアクセスし、VM が DstVMS 上でレジュームされた後に素早く eth1 経由に切り替えることにより、アドレス取得にかかる時間を短縮できる。最後に IMS に対してマッピングの更新を行う事で、ユニキャストの移動透過性が実現される。

ここで問題となるのは、VM 自身がレジュームしたタイミングをどのように検知するかである。提案手法では ICMPv6 を用いて DstAgent との到達性を確認する方法によって解決した。これは eth1 に移行先ネットワークのアドレスを設定すると、そのアドレスと同

\*2 初期状態が eth1 経由の通信の場合、eth0 を eth1 に、eth1 を eth0 に置き換えた同様の手順となる。

表 1 新しく定義したメッセージプロトコル  
Table 1 Additional messages for proposed protocol.

メッセージタイプ	名称	概要
AGENT_SEARCH	Agent 探索	自 VMS の Agent を特定するためのリンクローカル全ノードマルチキャスト.
AGENT_SEARCH_REPLY	Agent 探索応答	自 VMS が送信した VM を格納している場合に応答する Agent の応答.
MIGRATION_REQUEST	マイグレーション要求	VM がマイグレーション先 VMS の DstAgent を SrcAgent に通知する.
MIGRATION_RESPONSE	マイグレーション応答	ACK 及び, 移行先ネットワークのプレフィックス情報を通知する.
AGENT_REQUEST	Agent 代理 Join 要求	DstAgent が代理 Join すべきマルチキャストアドレスを通知する.
AGENT_RESPONSE	Agent 代理 Join 応答	移行先ネットワークのプレフィックス情報及び, マルチキャストアドレス毎に DstAgent がとる受信方法を通知する.
START_MIGRATION	マイグレーション開始	ライブマイグレーションの開始を指示する. (libvirt API)
RELAY_REQUEST	ユニキャスト中継要求	マルチキャストを受信しユニキャストで再送信するユニキャスト中継の要請.
RELAY_RESPONSE	ユニキャスト中継応答	ユニキャスト中継要求への ACK.

ーリンク上の DstAgent に対するルーティングは eth0 よりも eth1 が上位となり, eth1 経由のルーティングはマイグレーション後のみ有効となる特徴を利用している.

### 3.4 管理エージェントの動作

Agent は以下の 3 種類の動作を行う.

#### 3.4.1 マルチキャストアドレスの監視

マルチキャストアドレスへ Join を行う MLD Report は, ルータが定期的に送信する MLD Query に対する応答を行う場合と, 上位マルチキャストアプリケーションが受信開始・停止の操作を行った場合に送信される. MLD Report はマルチキャストアドレス宛へ送信されるため, ルータや該当 VM へ機能を追加することなく, リンクローカル他ノードの受信状態を把握できる.

そのため, Agent は各 VM が送信する MLD Report を監視し, その送信元 MAC アドレスと要請ノードマルチキャスト (ff02::1:ff00:0/104) 以外のマルチキャストアドレスを抽出する. これにより, マイグレーション時に該当の VM が受信しているマルチキャストを即時検索することが可能となる.

#### 3.4.2 マルチキャストへの代理 Join

移行先ネットワークでの Join を出来る限り早く完了させるために, グローバルライブマイグレーションが開始された時点で DstAgent が代理 Join を行う.

DstVMS 上にマルチキャストを受信しているノードが DstAgent だけであった場合, DstAgent が Leave するとマルチキャストツリーの再構築が中断されたり, 配送が停止してしまう恐れがある. したがって, Agent はマイグレーション中も Join 状態を維持する必要がある. マイグレーション後の VM が MLD Query に応答する MLD Report を送信するまで Leave してはならない.

#### 3.4.3 ユニキャスト中継の管理と代理受信

Agent は管理するマルチキャストアドレス全てに対し, そのアドレスを宛先とするストリームが継続して VMS に受信されているかを確認する. 一定時間以上受信が確認できない場合, マルチキャストが受信でき

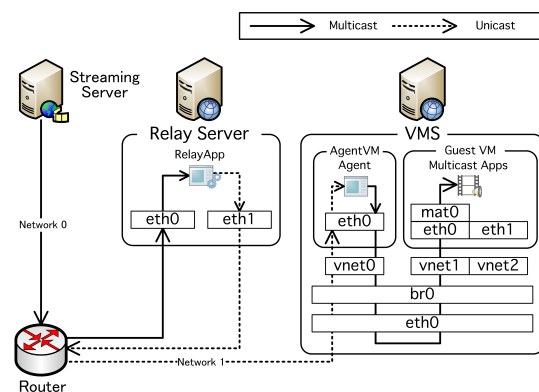


図 5 ユニキャスト中継を用いる際のストリームの流れ  
Fig. 5 Stream flow using unicast relaying.

ないネットワークの可能性があると判断しユニキャスト中継に切り替える.

ユニキャスト中継を使用する際のストリームの流れを図 5 に示す. Network 1 はマルチキャストが直接受信できないので, マルチキャストが受信可能な中継ノードで一旦受信し, それを Agent に向けてユニキャストで再送信する. そして Agent は, 受信したユニキャストストリームを, 再度 TTL を 1 としたリンクローカルなマルチキャストストリームとして再送信することにより, VM 上のマルチキャストアプリケーションへ到達する. なお, ユニキャスト中継を行うサーバのアドレスは, 文献<sup>5)</sup>で提案されている MCS (Migration Control Server) や, エニーキャストアドレス, DNS ラウンドロビン, SDP<sup>11)</sup>などの手法を用いて解決する.

このような手順により, VMS のネットワークで直接マルチキャストを受信できない場合でも, 最終的に VM ではマルチキャストとして受信され, 中継による帯域コストを最小限に抑える事が可能となる.

## 4. 実装環境

本章では, 提案手法の実現のために使用した仮想化



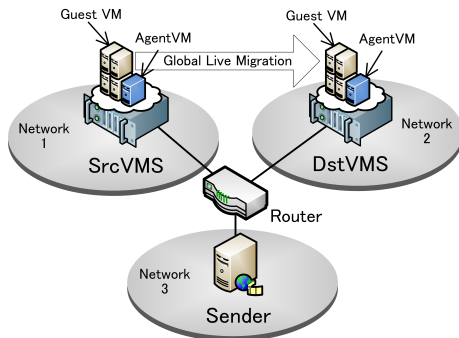


図 6 グローバルライブマイグレーション (GLM) の実験環境  
Fig. 6 Experiment network for Global Live Migration.

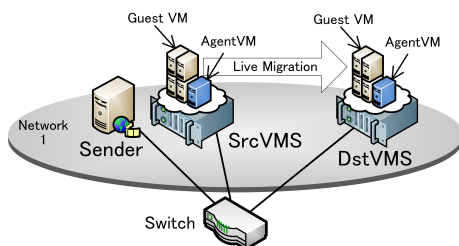


図 7 従来のライブマイグレーション (LM) の実験環境  
Fig. 7 Experiment network for traditional Live Migration.

環境について述べる。

VMSのハイパーバイザとしてLinux Kernelに統合されたKVM<sup>\*1</sup>を使用し、完全仮想化環境を提供するエミュレータQEMU<sup>\*2</sup>と組合わせたQEMU/KVM構成上に実装を行った。さらに仮想化APIとして各種仮想化環境をサポートしているlibvirt<sup>\*3</sup>を用い、Agentはlibvirt APIを通してQEMU/KVMの制御を行う。したがって、本論文で示した提案手法は特定の仮想化環境に依存するものではなく、他のハイパーバイザに対しても適用可能である。なお、QEMU/KVM、libvirtともに、現時点でIPv6によるマイグレーションが未サポートであったため、マイグレーションソケットにIPv6が利用できるよう修正を行った。

## 5. 基本性能評価

提案手法における効果を定量的に示すため、本章ではグローバルライブマイグレーションにおけるマルチキャスト受信の通信継続性と、移動透過IPマルチキャストのために追加した操作に要する時間に関する評価実験について述べる。

検証環境を構成するノード群を表2に示す。VMを稼働させる2台のVMS及び、マルチキャストスト

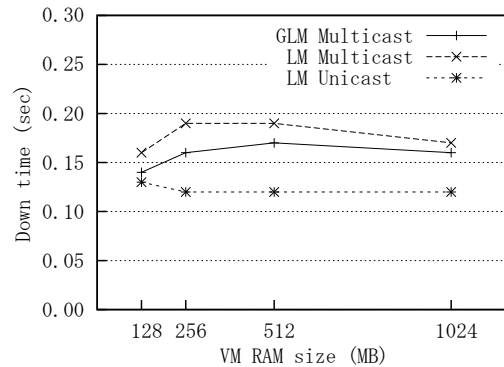


図 8 GLM 及び LM における通信途絶時間  
Fig. 8 Down time of GLM and LM.

リームを送信する Sender は、全て同一の Gigabit L3 スイッチ AlaxalA AX3630S-24T に接続されているが、ネットワーク構成については各測定により異なるため後述する。Sender は、Iperf<sup>\*4</sup>を用いて任意ビットレートのマルチキャストストリームを送信する。

VMの構成は、CPUは仮想2コアとし、メモリは128MB、256MB、512MB、1GBの4種類においてそれぞれ実験した。VMのシステムを格納するストレージは、外部ストレージへのアクセスが与える影響を取り除くため、事前に双方のVMS上のローカルディスクに共通のストレージファイルを配置し、スワッピングを無効化して測定を行った。

### 5.1 グローバルライブマイグレーションによるマルチキャストストリームの途絶時間

本測定では、提案手法を用いて異なるネットワーク上の別VMSへ移行するグローバルライブマイグレーション(以下、GLM)と、同一ネットワーク内の別VMSへ移行する従来のライブマイグレーション(以下、LM)について、VM内のマルチキャスト受信アプリケーションが受けるストリームへの影響を明らかにする。なお、VMとの通信が途絶する時間は、マイグレーションにおいて一時的にサスペンドされる図3の(3)の区間である。

図6及び図7に示す実験環境において、SenderからUDPマルチキャストストリームを送信し、VM内の測定アプリケーションでシーケンス番号を記録することで途絶時間を算出した。なお、本測定では20Mbpsのストリームを利用し、1725ppsの値をもとに計算した。

実験結果を図8に示す。各測定値は、5回測定した平均値である。マルチキャストアプリケーション上での途絶時間は平均で0.18秒であった。一方、移動透過アーキテクチャを含まないLMにおいて、外部ホストからping6を10ミリ秒間隔で送信して計測したユニキャストの途絶時間は、平均0.12秒であった。提

\*1 <http://www.linux-kvm.org/>

\*2 <http://qemu.org/>

\*3 <http://libvirt.org/>

\*4 <http://sourceforge.net/projects/iperf/>

表 2 実験機材

Table 2 Experiment machines.

	Sender	VMS	VM
CPU	Intel Core Solo U1400 1.20GHz	Intel Core i7 2600K 3.40GHz	Virtual 2-core CPU
RAM	1.5GB	8GB	128MB, 256MB, 512MB, 1GB
OS		Fedora release 14 (Laughlin)	
Kernel	linux-2.6.35.14-96.fc14.i686.PAE SMP		linux-2.6.32.16 SMP
Network Speed	100Mbps		1Gbps
Hypervisor	-	QEMU/KVM 0.13.0-1	-
API	-	libvirt-0.8.3-10	-

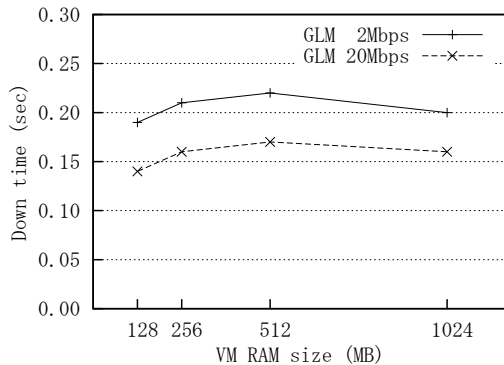


図 9 GLM におけるマルチキャストストリーム途絶時間  
Fig. 9 Down time of multicast stream on GLM.

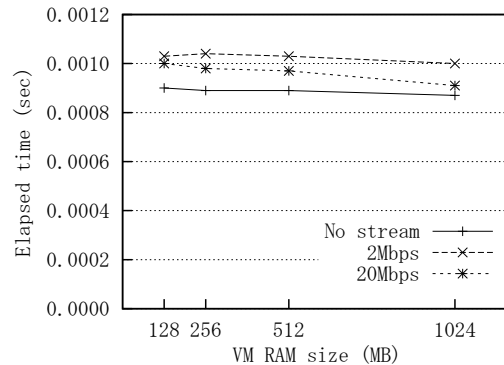


図 10 提案手法のための追加操作に必要な時間  
Fig. 10 Necessary time for additional operation of proposed method.

表 3 GLM におけるマルチキャストストリーム途絶時間  
Table 3 Down time of multicast stream on GLM.

RAM size (MB)	Stream bitrate / Down time (sec)					
	2Mbps			20Mbps		
	min	avg	max	min	avg	max
128	0.00	0.19	0.37	0.10	0.14	0.22
256	0.15	0.21	0.39	0.11	0.16	0.20
512	0.10	0.22	0.37	0.08	0.17	0.36
1024	0.11	0.20	0.37	0.07	0.16	0.32

案手法により、GLM においても LM 同等の途絶時間に抑えられている。

マルチキャストを受信しながらマイグレーションすると、GLM においても LM においても重複したパケットが計測された。重複した時間は、計測ごとにばらつきが見られたが平均して 0.3 秒であった。これは、ライブマイグレーションの終盤で VM がサスペンドされる直前に VM 内に受信されたパケットが、DstVMS でレジューム後にアプリケーションへ遅れて到達するために生じると考えられる。

### 5.2 受信ビットレートとストリーム途絶時間

本測定では、Sender が送信するマルチキャストストリームのビットレートが 2Mbps と 20Mbps の 2 種類の場合について、GLM を行う VM 上アプリのマルチキャストストリームに与える影響を、図 6 に示す環境で測定した。測定方法は、5.1 節と同様である。

図 9 及び表 3 に実験結果を示す。図 9 より、受信ビットレートに関して 20Mbps よりも 2Mbps の方が 0.04 秒ほど途絶時間が長くなっている。この原因とし

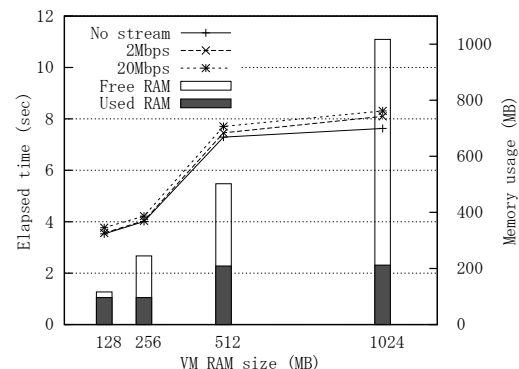


図 11 提案手法におけるマイグレーションに必要な時間  
Fig. 11 Necessary time for migrate operation of proposed method.

ては、ビットレートの低い方がメモリの変化が緩やかなため、最後のメモリページのサイズが大きくコピーに時間を要すると考えられる。なお、VM のメモリサイズの違いは途絶時間にほとんど影響を与えないことがわかる。表 3 より、今回測定した条件での途絶時間は 0.4 秒以内であり、同一ネットワーク内のマイグレーションと同等の途絶時間であることを確認した。

### 5.3 Agent の追加操作及びマイグレーションに要する時間

本測定では、マイグレーション開始から終了までの経過時間を、VM のメモリサイズとマルチキャストス

トリーム別に測定した。図 3 に示す (1) 及び (2) の区間について、図 6 に示す環境の SrcAgent 上で測定した。

- (1) 提案手法において追加した代理 Join のためのシグナリング時間
- (2) 実際に VM のメモリ情報などを SrcVMS から DstVMS にコピーする時間

図 10 及び図 11 に実験結果を示す。(1) の区間では、DstAgent において VM が受信しているマルチキャストストリームの代理 Join が行われているため、マルチキャストを受信していない時よりも受信しているときの方が若干時間がかかっている。また、DstAgent がマルチキャストの代理 Join を行った後は、ストリームが受信できる事を確認しているため、パケットの到着間隔が疎である 2Mbps よりも、密である 20Mbps の方が時間が短くなる。なお、ここでは VM のメモリサイズに依存する処理はないため、メモリサイズによって経過時間は変化しない。

(2) の区間では、VM のメモリ内容や CPU・VGA の情報を SrcVMS から DstVMS へと転送しているため、VM のメモリサイズが大きいほど経過時間が長くなっている。しかし仮想化ソフトウェアの特性により、空メモリ部分の転送が高速化されているため、経過時間とメモリ割当量は比例せず、実際のメモリ使用量及び、ストリーム受信によって書き換えられるメモリ量に応じて増加している。

(1) におけるシグナリング時間は、SrcAgent と DstAgent 間のネットワーク距離に応じて増加する可能性はあるが、従来手法のマイグレーション後に VM が MLD Query を受信して MLD Report を送信するまでの時間よりも十分小さく、マイグレーション全体に必要な時間は (2) の部分が支配的である。以上の結果から、提案手法を用いることによりマルチキャストの移動透過性が期待できる。

## 6. まとめと今後の課題

本論文は、ユニキャストに限定されたグローバルライブマイグレーションに対し、マルチキャストに対する移動透過性を実現する手法について述べた。提案手法では、Agent においてマルチキャストストリーム及び MLD による受信状態を管理することにより、マルチキャストアプリケーションの変更を必要とせず、マイグレーション後のマルチキャスト途絶時間を短縮することが可能となる。

今後の課題として、マルチキャストが利用できないエリアでの動作検証及び性能評価を行うことが挙げられる。さらに、ユニキャストそのものの移動透過アーキテクチャを組み込むことで、ユニキャスト・マルチキャスト両対応のグローバルライブマイグレーションの実現を進めていく予定である。

謝辞 本システムの設計や技術的問題の解決方法についてご意見くださいました、広島大学情報メディア教育研究センター情報基盤研究部門、情報セキュリティ研究部門、広島市立大学インターネット工学研究室の関係者に心から感謝いたします。本研究の一部は、科学研究費補助金(23300026, 22700075)の支援を受けて実施しています。ここに記して謝意を表します。

## 参考文献

- 1) Cain, B., Deering, S., Kouvelas, I., Fenner, B. and Thyagarajan, A.: Internet Group Management Protocol, Version 3, RFC 3376, IETF (2002).
- 2) Vida, R., Ed., Costa, L. and Ed.: Multicast Listener Discovery Version 2 (MLDv2) for IPv6, RFC 3810, IETF (2004).
- 3) Fenner, B., Handley, M., Holbrook, H. and Kouvelas, I.: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised), RFC 4601, IETF (2006).
- 4) 近堂 徹, 西村浩二, 相原玲二: 移動透過通信機能を持つ仮想計算機によるセッションモビリティの実現, インターネットコンファレンス論文集, Vol.2008, pp.42-48 (2008).
- 5) 渡邊英伸, 大東俊博, 近堂 徹, 西村浩二, 相原玲二: IP モビリティと複数インタフェースを用いたグローバルライブマイグレーション, 電子情報通信学会論文誌. B, 通信, Vol.93, No.7, pp. 893-901 (2010).
- 6) 相原玲二, 藤田貴大, 前田香織, 野村嘉洋: アドレス変換方式による移動透過性インターネットアーキテクチャ, 情報処理学会論文誌, Vol.43, No.12, pp.3889-3897 (2002).
- 7) 神屋郁子, 下川俊彦, 吉田紀彦: 柔軟な構成変更を可能とする広帯域配信システムの構築, 電子情報通信学会技術研究報告. IA, インターネットアーキテクチャ, Vol.109, No.421, pp.13-16 (2010).
- 8) 宮城亮太, 池部 実, 猪俣敦夫: クラウド環境におけるサーバ負荷に応じた動的計算資源割当システムの提案と評価, 電子情報通信学会技術研究報告, Vol.110, No.430, pp.17-22 (2011).
- 9) 鎌田恵介, 近堂 徹, 相原玲二: ユニキャストを併用する移動透過 IP マルチキャストの設計, 電子情報通信学会技術研究報告. IA, インターネットアーキテクチャ, Vol.110, No.304, pp.13-18 (2010).
- 10) Thomson, S., Narten, T. and Jinmei, T.: IPv6 Stateless Address Autoconfiguration, RFC 4862, IETF (2007).
- 11) Handley, M., Jacobson, V. and Perkins, C.: SDP: Session Description Protocol, RFC 4566, IETF (2006).