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

近堂徹 † 西村浩二 † 相原玲二 † 前田香織 ‡

† 広島大学情報メディア教育研究センター
‡ 広島市立大学大学院情報科学研究科

あらまし 近年，計算機資源の柔軟な利用を目的とした仮想化技術が一般化しつつある．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となる．しかしながら，サイトをまたが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．そのため，セッションを継続的に確立するアプリケーションへの適用が困難となる．本研究では仮想計算機に移動透過通信機能を組み込むことでこれらの問題を解決し，仮想計算機のセッションモビリティの実現とそれらに付随する問題点について考える．

1 はじめに

クライアント端末の小型化・省電力化やネットワークインフラの普遍化に伴い，移動透過性を有する通信への要求が高まってきている．IP ネットワークにおける移動透過性を実現することにより，複数ネットワーク間の移動による IP アドレス等の変化を隠匿することが可能となり，トランスポート層以上のセッションに影響を与えることなく継続的な通信が可能となる．これまで，様々な手法が提案され，最近では端末への実装と広域ネットワーク環境での検証利用が行われるようになってきている [1][2]．このように，移動端末における IP 移動透過性については技術的に実用可能な段階に入ってきたといえる．

一方，CPU やメモリの高性能化・低価格化により，マルチコアプロセッサやデュアルチャネルメモリ機構などが身近な存在となっている．これに伴い，ハードウェアの仮想化技術もより一層注目を集めてきている．この仮想化技術では，オペレーティングシステム（以下 OS）の動作環境から実行プラットフォームとなっているハードウェアを隠匿することができ，1 つのハードウェアで複数の OS を同時並行に実行することが可能となる．また，OS をひとつのプロセスとして扱うことが可能となり，メモリやネットワークセッションなどの OS 内部状態を保持したまま，一時停止（サスペンド）や移動（マイグレーション）を行うことができる．これにより，OS 上で動くアプリケーションを，仮想計算機群で構成するプラットフォーム内で容易に展開や移行することができるようになる．

しかしながら，広域インターネット環境における仮想計算機の移動を考える場合，IP 到達性が問題となる．仮想計算機があるネットワークから他のネットワークに移動する際に物理的ネットワークが変更となるが，仮想計算機自体はネットワークの変化を意識することなく IP アドレス等は変更されない．故に，ネットワーク到達性が失われると同時に，移動前に確立していた上位層のセッションに影響を与える [3][4]．

本研究では，仮想化環境で動作する計算機に IP 移動透過性を持たせることで，その上で動くアプリケーションのセッションモビリティを実現する．ここで,「セッションモビリティ」を「仮想計算機が動作している実計算機が切り替わっても、仮想計算機上で実行中のアプリケーションの動作およびその通信セッションが継続できること」と定義する．これにより，ネットワークの場所に依存せずアプリケーションへの透過的なアクセスが可能となり，ユーザとのセッションを維持するネットワークサービスを連続的に提供できるサービスモビリティへ応用することができる．

本稿では，その基本システムの概要と実装について述べ，基礎的な性能評価について示す．それらの

結果から，複数のネットワーク環境下でも仮想 OS 上のセッションが維持できることを示すとともに，通信に与える影響についても考察する．

本稿の構成は以下の通りである．まず，2. にて本研究が対象とするモビリティ要件について示し，それを解決するための提案システムを 3. で述べる．4. では実装したプロトタイプシステムにて基礎性能評価を行うとともに，その結果から実用性に関する考察を行う．最後に，5. にてまとめと今後の課題について述べる．

2 広域インターネット環境におけるセッションモビリティの実現

モビリティ技術の進展により，移動端末における IP 移動透過性だけでなく，様々なレイヤでの移動透過性確保に向けた検討が行われている．その中で，広域インターネットにおけるセッションモビリティは，通信インフラの拡充と計算機資源の有効活用の観点から，今後様々な環境において重要となってくると考えられる．ここでセッションモビリティの活用事例として，ストリームのリアルタイム広域配送におけるネットワーク状況に応じたサーバの動的資源割当を挙げる．広域インターネット環境で，映像などのリアルタイムストリームの多地点配送を行う場合，ストリーミングサーバや中継サーバをどのように配置し配送ネットワークを構築するかは重要な問題となる [6]．セッションモビリティを応用することで，ストリーム配送ネットワークにおいて，時々変化するネットワーク状況や参加ノードの状況に応じてサーバの資源を動的配置することが可能となる．また，サーバ機能自体をセッション情報を保持しながら適切なネットワークに移動させることにより，ユーザからの透過的なアクセスを可能にしつつ，ユーザの偏りによるネットワークトラフィックの負荷分散も実現できるようになる．

これまで提案されてきたセッションモビリティを実現するプロトコルとして，アプリケーション層で実現する SIP Mobility [5] やセッション層で実現するモビリティ [7] などが挙げられる．これらは主にエンドユーザが利用する端末間移動を想定したものであり，より上位の細かなセッション単位での移動が可能となり柔軟性は大きく向上する一方で，アプリケーションを改変する必要性が生じるなどの問題点もある．

本研究では，サービスの効率性，柔軟性などを考慮したセッションモビリティ機能の実現を考える．このために，前述したトランスポート層以上のプロトコルを利用するのではなく，ハードウェアの仮想化技術と IP モビリティ機能を融合させた方法を実現する．文献 [3] では，モバイル IP と仮想化技術を用いたサーバの透過的利用について提案されているが，広域ネットワークをまたがって仮想環境を移動させる際，セッションを維持することへの検討および評価が不十分であった．本研究では，プロトタイプ実装とその実現性についても検討する．

3 IP 移動透過性を有する仮想計算機

本節では，本研究の目的である IP 移動透過性を有する仮想計算機を実現するにあたり，必要な要素技術について述べ，提案システムおよびその実装方法について述べる．

3.1 要素技術

ここでは，提案システムで利用する，ハードウェア仮想化技術とモビリティサポートアーキテクチャ MAT について述べる．

3.1.1 ハードウェア仮想化技術

ハードウェア仮想化とは，ハードウェア資源を抽象化することにより，あるコンピュータ・ハードウェアを複数台のコンピュータ・ハードウェアのようにエミュレートして動作させる技術である．この仮想化技術のメリットとして，先述した仮想計算機のマイグレーション（移動）機能がある．この機能により，仮想計算機の内部状態（メモリイメージ，レジスタ内容，ネットワークセッションなど）のみを移動させることで，ある実計算機で動作している仮想計算機を稼動させたまま別の実計算機に移動させることが可能となる．しかしながら，現状これを実現するためには，マイグレーション前後のネットワークを同一ネットワークにしておかなければならず，ネットワークの異なるサイト間で利用することはで

きない．これは，仮想計算機自身がネットワークの移動を検知することができず，マイグレーション後でも古い IP アドレスを維持した状態となることで，ネットワーク到達性が失われてしまうためである．この点が，セッションモビリティの広域展開を考える場合に非常に大きな問題となる．VLAN 技術などにより広域イーサネットを構築することも可能 [4] であるが，通信にかかるオーバーヘッドや，仮想計算機インフラストラクチャのスケーラビリティを考えると，IP ルーティングによる広域インターネット網で展開できることが望ましい．

3.1.2 モビリティサポートアーキテクチャ MAT

MAT（Mobility Support Architecture and Technologies） はアドレス変換方式を採用することで，TCP/IP においてトランスポート層以上に対して移動透過通信を可能にするアーキテクチャである [13]．現在，ノードの移動をサポートする MAT-HOST, およびネットワークの移動をサポートする MAT-NET が提案されている．

MAT-HOST の概要は以下の通りである．MAT 機能を有する移動ノード（MN: Mobile Node) は，移動先で付与される一時的なモバイルアドレス (MA: Mobile Address) とアプリケーションが通信を行う際に使用する恒久的なホームアドレス（HA: Home Address) を持つ．MN は移動に伴って MA が変更となるたびに，IP アドレスのマッピング情報を管理する IMS(IP address Mapping Server) が保持する HA と MA の対応表を更新し，移動端末内でのアドレス変換を行うことで，トランスポート層以上で移動透過な環境を提供する．MAT は常に最適経路による端末間通信を実現し，トンネリング技術を使用しないためトンネルオーバヘッドによるパケット長の増加が発生しないなど，多くの特長を持っている．また，既に Windows(MAT-HOST)/Linux(MAT-HOST/MAT-NET) での動作実績があり，単一インタフェース/複数インタフェースでのハンドオーバが可能であることも確認できている．本研究では，プロトコルオーバーヘッドの優位性および動作実績の点から移動透過アーキテクチャとして MAT を採用する．

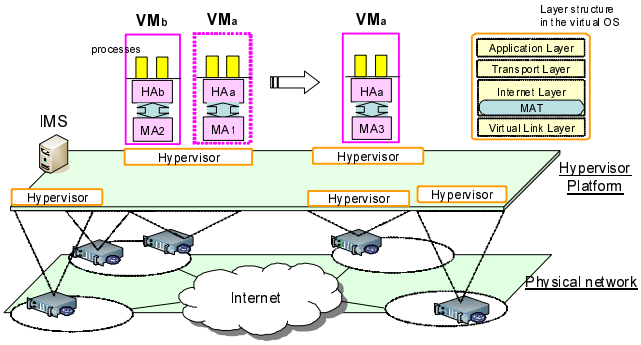

図 1: システム構成図.

3.2 提案システムの概要

本研究で提案するシステム構成を図 1 に示す．

まず各要素について説明する．各実計算機上で動く仮想マシンモニタをハイパーバイザとよび，ハイパーバイザが動作する実計算機群で構成する広域プラットフォームをハイパーバイザプラットフォームと定義する．ハイパーバイザ上の仮想計算機で動く OS をゲスト OS と呼び，ユーザへサービスを提供するためのアプリケーションが動く OS となる．ネットワーク接続環境として，各ゲスト OS に対して仮想ブリッジを提供することで外部接続性を確保する．その他，プラットフォーム上には MAT で利用する IMS が必要となる．

次に，具体的な処理内容について図 2[1] に示す．本システムでは，前節にて説明した仮想計算機のマイグレーション機能を利用することで，その上で動くサービスを OS ごと任意の計算機に移動させ，ネットワーク位置に依存しない一意なアクセスを実現する．具体的な処理は大きく分けて以下の 2 つである．

(I) OS のマイグレーション処理を行い，OS のスナップショットを別計算機に移動，復元させる

(II) 新しいネットワークアドレスを取得し，ハンドオーバ処理を行う

まず (I) の OS のマイグレーション処理では，ユーザからの要求に応じて，移動元のハイパーバイザがゲスト OS のメモリ状態を保存し，ネットワークを通して移動先のハイパーバイザ上に転送する．移動先のハイパーバイザでは転送されたイメージを

[1] 図中の (1)-(5) については 4.1 評価実験の際に参照する．

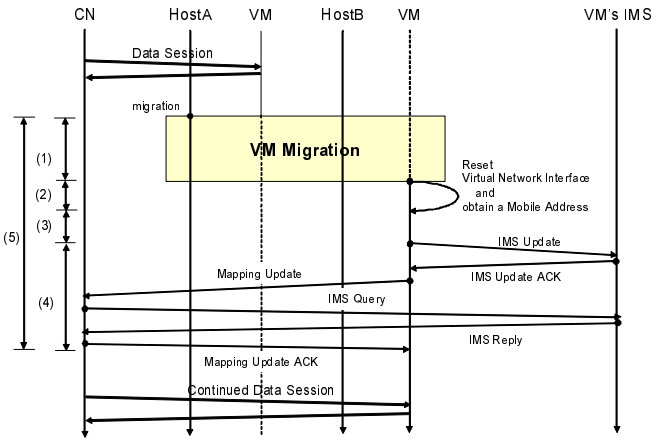

図 2: マイグレーション処理の流れ.

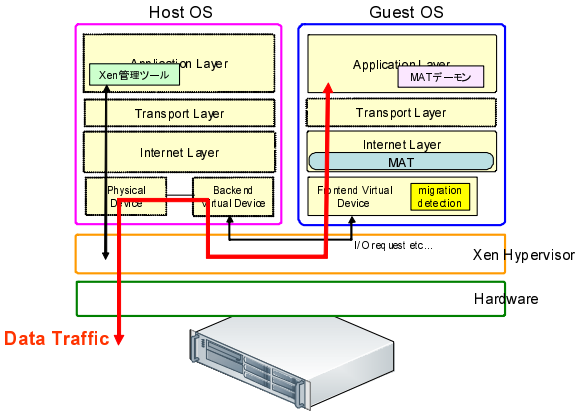

図 3: システムの実装.

自身の仮想計算機上に展開することで，元の OS を再開させる．なお今回構築するシステムでは，動作する仮想 OS のファイルシステムはディスクイメージとして作成し，予めプラットフォーム上の各ノードに配布しておくこととする．プラットフォーム内での OS イメージ共有手法は，この他に Gfarm[9] や iSCSI[10] などの分散ストレージを利用した共有手法が考えられるが，OS イメージの配布・管理スキームについては今後の課題である．

次に (II) の MAT によるモビリティ機構は，マイグレーション処理における仮想計算機のネットワークアドレスの変化を隠匿するために用い，ゲスト OS の IP 層に位置する．移動後にはゲスト OS 自身が，ネットワークを移動したことを検知して MA を取得し，HA と MA の対を IMS へ通知する．IMS からの応答が帰ってくると同時に，通信相手ホストに対して移動通知を行う．ゲスト OS の通信継続性は MAT によって保証される．このように，ゲスト OS の IP 層にて MAT によるアドレス変換を行うことで，上位層アプリケーションを改変する必要がなくセッションモビリティが低コストで実現可能となる．

3.3 プロトタイプ実装

本節では，本提案システムの実現性を検証するためのプロトタイプの実装方法について示す．本システムの内部構成は図 3 に示す通りである．プロトタイプ実装では，仮想計算機システムとして仮想計算機モニタの Xen[11][2]，MAT の実装として Linux 版 MAT2.0[3] を利用した．

プロトタイプ実装における Xen では，仮想マシンのパフォーマンスを重視し Para-Virtualization（準仮想化）で稼動させることとした．Para-Virtualization はゲスト OS のソースコードを一部修正し仮想化 API を利用できるようにすることで，仮想化処理のオーバーヘッドを低減させ，仮想計算機モニタを利用しない場合と同程度のパフォーマンスを維持できるものである．実装では，Xen 対応カーネル（カーネル 2.6.16.29）の IP スタックに MAT 処理部を追加した．加えて，マイグレーション処理前後のハンドオーバを管理するために，MAT デーモン [12] を動作させている． MAT デーモンは，インタフェースの状態監視を行い，ネットワークセグメントの移動を検知した時点で，RS (Router Solicitation）を送出して新しい MA の取得を行う．アドレスが取得できれば，HA とのマッピング情報を IMS に通知することで，ハンドオーバを実現する．仮想計算機のマイグレーションでは，有線のシングルインタフェースハンドオーバと同様の動きとなるため，現状の MAT ツールをそのまま利用している．

今回，仮想計算機の広域マイグレーションに伴い，新たに実装したのは以下の点である．マイグレーション時のゲスト OS の挙動として，実ネットワークの移動を意識することなく継続利用可能にするため，インタフェース状態はアップかつ稼動状態

[2] http://www.xen.org
[3] http://www.mat6.org

でサスペンドし，移動後に復帰させるようになっている．そのため，MAT デーモンではネットワークの移動を検出することができず，新しいネットワークに移行できない．これを解決するために，ゲスト OS のフロントエンドドライバを改良し，ネットワークが切り替った際に仮想インタフェースを強制的にリセットし，その変化を MAT デーモンに伝えるように変更した．これにより，マイグレーション終了時に即時に MAT デーモンから RS が送出され，新しいネットワークのアドレスを付与される．その間，上位層のアプリケーションは HA を用いてセッションを生成しているため，不具合は生じない．

実装では，ゲスト OS のマイグレーションの命令などは Xen の機能を利用しており，各ハイパーバイザで動作するホスト OS より行われる．また，ネットワーク接続性についても，ホスト OS が提供する仮想ブリッジを経由して実ネットワークと接続されるようになっている．

4 基本性能評価

本節では，実装したプロトタイプを用いたマイグレーションにおける仮想計算機の通信継続性に関する評価を行う．得られた結果などからセッションモビリティの実現方法としての本提案システムの有効性や問題点について考察する．

4.1 評価実験

実験構成図を図 4 に示し，機器の仕様を表 1 に示す．仮想計算機 (VM) の仮想 CPU として各ホストの 1 コア分を割当て，仮想メモリは 32, 64, 128, 256 (MB) の 4 種類の場合で実験した．仮想計算機で動作させる OS は Debian/GNU Linux 4.0 としている．実験では，IPv4/v6 ルータ (有線インタフェース 100Mbps) を介して 3 つのネットワークを用意し，ホスト C とのセッションを維持したまま，VM がネットワーク A とネットワーク B の間を移動する環境を構築した．ルータは各々のネットワーク内に RA(Router Advertisement) にて IPv6 アドレスを配布しており，VM は移動先のネットワークで RA から MA を生成する．なお，マイグレーション処理自体は，Xen の仕様により IPv4 で行っている．検証では，以下の 2 点について確認する．

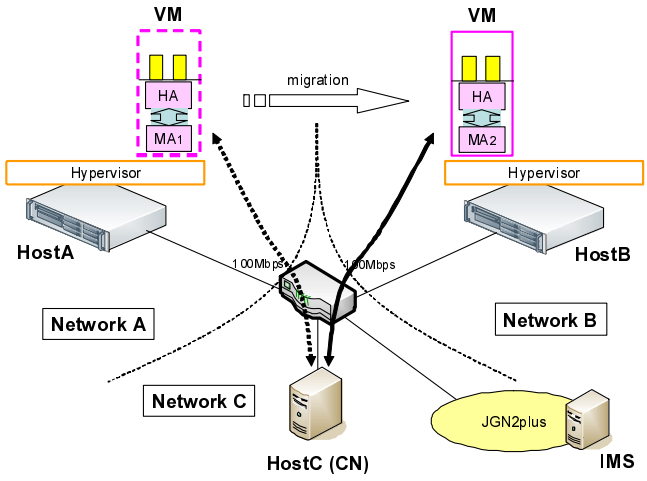

図 4: 実験環境.

表 1: 評価実験における使用機器.

	CPU (GHz)	Memory (MB)	OS / kernel
HostA,B	Core2Quad 2.6	2045	Debian 4.0 2.6.16.29
Host C (CN)	Pentium 4 2.8	512	Debian 4.0 2.6.16.29
Router	Allied Telesis CentreCOM AR450S		

(a) マイグレーション後にネットワーク B の MA が割当られ，MAT の動作により TCP セッションが途切れることなく通信が継続できること

(b) マイグレーション命令を発行した時点から通信が再開されるまでの時間

まず，(a) については，VM を SSH サーバ，CN を SSH クライアントとして両者の間にコネクションを張り，マイグレーションを行った場合でも SSH セッションが継続できることを確認した．これはメモリサイズがいずれの場合でも正常に継続できることを確認している．

次に，(b) の計測方法を示す．プローブとして，VM から ICMP echo リクエストパケットを 100ms 間隔で CN に送出し続け，マイグレーションを発生させる．VM の仮想インタフェースおよび CN の物理インタフェースにて通過パケットをキャプチャすることで，マイグレーション処理中の通信途絶時間とその内訳を計測した．計測項目としては，図 2 の (1)-(5) で示す以下の 5 項目である．

表 2: マイグレーション処理時間.

memory size (MB)	time (msec)				
	(1)	(2)	(3)	(4)	(5)
32	4,206	2,203	2,013	1.30	8,449
64	7,447	2,022	2,011	0.91	11,481
128	13,060	2,731	2,010	1.04	17,801
256	27,065	1,802	2,032	0.92	29,876

(1) マイグレーション開始から OS 復帰までの時間

(2) VM インタフェースの再起動から RA 受信までの時間

(3) RA を受信してから IMS Update を送出するまでの時間

(4) IMS Update を送出してから Mapping Update ACK を受信するまでの時間

(5) 全通信途絶時間

測定結果を表 2 に示す．値は，それぞれ 3 回計測した平均値を代表値としている．設定する全仮想メモリ量が大きくなるに従いマイグレーションよる通信途絶時間 (5) も大きくなっていることがわかる．内訳を見てみると，(1) の時間は仮想化ソフトウェアの特性によるものであり，設定したメモリ量を転送する時間とほぼ一致している．メモリサイズが大きい場合，全通信途絶時間は (1) が支配的となる．

(2)-(4) は本提案システムの MAT を取り入れたことによるオーバーヘッドである．これはメモリ量に関係なく一定時間必要であることが分かる．今回の実験では，インタフェースの再起動から通信開始まで大よそ 5 秒程度要していることが分かる．(4) については，ネットワーク接続環境によっても変動する値であり，実環境では増加する可能性が高いが，メモリデータのコピー時間と比較すると十分小さな値になることが予想されるため，動作に影響はないと考えられる．(2)-(3) はインタフェース操作および MAT 処理に要する時間である．

今回の実験では TCP セッションが切れるほどの影響は出なかったが，より停止時間を短くしたい場合にはメモリサイズの最小化とインタフェース操作の高速化等を考える必要がある．

4.2 考察

前節の評価実験結果より，本システムの有効性および付随する問題点について考える．

今回の実験では，安定性および基本性能評価の観点から Xen の通常マイグレーション機能を利用した．その結果，メモリデータのコピー時間が多くを占めるものとなったが，より高速なリンクの利用や，ライブマイグレーションなどの他のマイグレーション手法などを利用することで，この時間は短縮できる．この辺りは，安定性や仮想化ソフトウェアの実装とのトレードオフであるともいえ，アプリケーションの要求条件に応じた使い分けが必要であると考えられる．

本システムは，仮想化技術と IP 移動透過性を利用することでセッションモビリティを実現している．仮想化を利用することで処理のオーバーヘッドが懸念されるが，Xen の Para-virtualization やハードウェアの仮想化支援技術を利用することで，実計算機での動作と同等の性能を得ることができる．逆に，ハードウェア仮想化のメリットを活かし「1OS・1 サービス」という考え方に立つと，他のアプリケーションやシステムに影響を与えることなく，サービスごとネットワークを移動させることが可能となる．さらに，ハードウェアの仮想化および IP 層の仮想化という 2 つの仮想化を融合させることで，アプリケーションの改変も必要としないため，あらゆるアプリケーションを対象とすることができる．また IP 移動透過通信（MAT 通信）によるオーバーヘッドについては，文献 [13] に示されている通り，広帯域アプリケーション利用においても十分実用に耐えうるものであるため問題はないと考えている．

問題点としては，仮想計算機との移動透過通信を実現しようとする場合には，通信相手も MAT に対応していなければならない点が挙げられる．この点については，MAT 実装に依存する部分でもあり，トンネリングオーバーヘッドの削減およびエンドホスト間での最適経路通信を優先した結果である．しかしながら，MAT については，Linux および WindowsOS での MAT-HOST 実装および MAT 実装を持たないノードに対する移動透過性を提供する MAT-NET 実装が既に存在するため，導入の敷居は比較的低いといえる．なお，ハードウェアの仮想化および IP 層の仮想化という観点から考えると，

IP 層での移動透過性が確保できるのであれば，他のアーキテクチャでも実現可能である．

5　まとめと今後の課題

本稿では，仮想計算機に移動透過機能を持たせることでセッションモビリティを実現する手法について述べた．本提案手法では，アプリケーションのセッション状態など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でき，仮想計算機の IP 移動透過性を利用することでアプリケーションの改変を必要とせず低コストで導入可能となる．

本評価実験の結果からマイグレーションにおける接続性の保証およびハンドオーバ処理時間を示すことができ，これは今後の実利用の際に重要な指標とすることができる．今後の課題として，仮想計算機の移動透過通信の安定化・マイグレーションの高速化を検討するとともに，広域インターネット環境でのセッション指向アプリケーションを用いた検証実験などを考えている．

謝辞

本研究にあたって，システム設計および実装について議論にご参加頂いた MAT プロジェクトメンバーの皆様に感謝します．なお本研究の一部は，日本学術振興会 科学研究費補助金（課題番号 20700066, 20300029, 19300019），総務省戦略的情報通信研究開発推進制度（SCOPE-地域 ICT, 082308001）の支援を受けて実施しています．ここに記して感謝の意を示します．

参考文献

[1] 桐山沢子，谷山健太，藤巻聡美，岡田耕司，湧川隆二，寺岡文男，中村修，”Performance Evaluation of IPv6 and NEMO technologies over Mobile WiMAX testbed”, 情報処理学会研究報告 ITS, Vol.2007, No.116, pp. 75-81, 2007

[2] 森廣勇人, 畠中翔, 前田香織, 井上博之, 相原玲二, 岸場清悟, ”移動透過アーキテクチャMAT のスケーラビリティに関する評価”, 電子情報通信学会技術研究報告. IA, Vol. 108, No. 74, pp.49-54, 2008.

[3] 久行恵美，井上伸二，角田良明，戸田賢二，須崎有康，”「ネットワークを渡り歩けるコンピュータ」を利用したネットワークトラフィック削減のための負荷分散手法，電子情報通信学会論文誌 B Vol.J89-B No.4 pp.443-453，2006.

[4] 立薗真樹, 中田秀基, 松岡聡, “ 仮想計算機を用いたグリッド上での MPI 実行環境 ”, *Proceedings of Symposium on Advanced Computing Systems and Infrastructures*, pp.525–532, 2006 .

[5] Schulzrinne, H., Wedlund, E., ”Application-layer mobility using SIP.”, *Proceedings of ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 4, No. 3*, July 2000.

[6] 近堂徹，岸田崇志，西村浩二，前田香織，相原玲二，”アプリケーションゲートウェイを利用したハイビジョン映像広域伝送実験”，情報処理学会 マルチメディア・分散・協調とモバイル (DICOMO) シンポジウム 2005 論文集，pp.521–524，2005.

[7] 金子晋丈, 河内祐介, 森川博之, 青山友紀, 中山雅哉, ”セッションレイヤにおけるエンドツーエンド型モビリティサポートの実装と評価”, 電子情報通信学会技術研究報告　MoMuC, Vol.102, No.87, pp. 51-56, 2002.

[8] 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.

[9] 建部修見，曽田哲之，”広域分散ファイルシステム Gfarm v2 の実装と評価”， 情報処理学会研究報告, 2007-HPC-113， pp.7-12，2007

[10] 広渕崇宏，谷村勇輔，中田秀基，田中良夫，関口智嗣，”複数サイトにまたがる仮想クラスタの構築”, 情報処理学会 SWoPP2007, 2007.

[11] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A., ”Xen and the Art of Virtualization”, *Proceedings of the ACM Symposium on Operation Systems Principles*, 2003.

[12] 畠中翔, 森廣勇人, 前田香織, 相原玲二, 岸場清悟,” 最適経路通信を行う階層ネットワークモビリティの実装と評価,”インターネットコンファレンス 2007 論文集, pp.59-67, 2007.

[13] 相原玲二, 藤田貴大, 岸場清悟, 田島浩一, 西村浩二, 前田香織,” 常に最適経路で通信を行う移動透過アーキテクチャMAT の性能評価”, インターネットコンファレンス 2006 論文集, pp.9-17, 2006.