10 Gb Ethernet 上の RDMA 転送機能による仮想マシン移動の設計と評価

中島耕太[†,††] 佐藤充[†]
久門耕一[†] 谷口秀夫[††]

本論文では，10 Gb Ethernet 上の RDMA 転送機能による仮想マシン移動の設計と評価について述べる．転送時間の削減のために，RDMA 転送機能の適用と，NIC による通信処理と CPU によるページのマップ/アンマップ処理のオーバラップ化を図る．また，転送処理が消費する CPU 時間の削減のためには，1 ページ転送あたりの CPU 時間と総転送ページ数を削減する必要がある．仮想マシン上でアプリケーションを動作させたまま転送する場合は，転送処理の間，アプリケーションがページを更新するため，更新ページの再送が生じる．そこで，転送時間の削減により，再送ページ数を削減する．そして，RDMA 転送機能の適用により，1 ページ転送あたりの CPU 時間を削減する．RDMA 転送を適用した結果，TCP/IP による転送時と比較して，アプリケーションが動作している 2 GB の仮想マシンの転送時間を 40.7%削減し 6.40 s（336 MB/s 相当）を達成した．また，転送処理が消費する CPU 時間を最大 73.6%削減し，仮想マシン上で動作するアプリケーション性能を最大 2.68 倍に改善した．さらに，オーバラップ化を適用した結果，オーバラップ化非適用時と比較して，転送時間を 50.8%削減し 3.15 s（681 MB/s 相当）を達成した．また，転送処理が消費する CPU 時間を最大 11.7%削減し，仮想マシン上で動作するアプリケーション性能を最大 6.4%改善した．

Design and Evaluation of a Virtual Machine Migration Using RDMA Data Transfer Mechanism over 10 Gb Ethernet

Kohta Nakashima,[†,††] Mitsuru Sato,[†] Kouichi Kumon[†] and Hideo Taniguchi[††]

This paper describes design and evaluation of a virtual machine (VM) migration using RDMA data transfer mechanism over 10 Gb Ethernet. In order to reduce elapsed time, we apply RDMA data transfer mechanism and overlap data transfer processing by NIC and page map/unmap processing by CPU. In order to reduce CPU time of VM migration, it is necessary that reduction of CPU time per a page transfer and total number of transfer pages. We apply RDMA to reduce CPU time per a page transfer. And in running application on VM, the reduction of elapsed time reduces total number of transfer pages. By using RDMA data transfer, the migration time of the 2 GB VM on which application was running was shorter in 40.7% than using TCP/IP data transfer, and 6.40 s (suitable to 336 MB/s) was achieved. Moreover, CPU time of VM migration was reduced in 73.6% and the performance of application on VM is improved 2.68 times. In addition, the migration time applied the overlap method was shorter in 50.8% than applied only RDMA, and 3.15 s (suitable to 681 MB/s) was achieved. CPU time of VM migration was reduced in 11.7% and the performance of application on VM is improved 6.4%.

1. はじめに

近年，PC を構成する CPU やメモリといった部品のコモディティ化が進み，PC サーバの低価格化と高性能化が著しい．このため，従来は大規模 SMP サーバを用いて実現していた Web サーバや DB サーバといったサービスが，価格性能比の高い PC サーバ上で実現されるようになった．また，1U サーバやブレードサーバのような高密度実装された PC サーバが広がっており，これを用いて，従来は分散配置されていたサーバをブレードサーバ上に配置し，集約管理する動きが広がりつつある．

1U サーバやブレードサーバのような多ノードサーバにより構成されるシステムの管理を行ううえで，近

† 株式会社富士通研究所
Fujitsu Laboratories Ltd.
†† 岡山大学
Okayama University

年，仮想マシン技術が注目されている．PC サーバ用の仮想マシンの実装である VMWare[1] や Xen[2] では，仮想マシンのノード間移動機能が実現されている．この移動機能の利用により，可用性や保守性の向上，動的負荷分散，省電力化が実現でき，多ノードサーバシステムの保守管理が柔軟かつ容易に行えるようになる．

多ノードサーバシステム上で仮想マシンを利用する際には，通常数 100 MB〜数 GB 程度のメモリを仮想マシンに割り当てる．また，5〜10 台程度の仮想マシンを 1 つの物理サーバ上に配置する場合が多い．将来においては，搭載メモリ量の増加や CPU 性能の向上により，仮想マシンへ割り当てるメモリ量や 1 つの物理サーバに配置する仮想マシン数は，増加すると予測される．このため，将来的には，数十〜数百台の各物理サーバ上で，数 GB のメモリを持つ仮想マシンを数十台配置し，サービスを提供するシステムが広く用いられるようになると考えられる．

このような環境で仮想マシン移動を利用すると，多数かつ大容量の仮想マシン移動処理が発生する可能性がある．この場合，転送時間の長大化や転送処理によるアプリケーション性能に対する影響が増大するため，仮想マシン移動の利用は困難になる．したがって，より柔軟かつ容易な保守管理実現のためには，高速かつ低負荷な移動処理を実現する必要がある．

仮想マシン移動処理では，大量のページ転送処理が発生するため，高速化のためには，現在広く用いられている 1 Gbps クラスの通信路よりも高速な通信路，たとえば，InfiniBand[3] や 10 Gb Ethernet（10 GbE）といった 10 Gbps クラスの通信路を使用すべきである．この際，単純に通信路を 10 Gbps クラスのものに置き換えるだけでは，ホスト OS 上でプロトコル処理やコピー処理がボトルネックとなり十分な転送性能は達成できない．したがって，TOE（TCP/IP Offload Engine）や RDMA（Remote Direct Memory Access）に代表されるオフロード技術を使用する必要がある．

そこで，高速かつ低負荷な仮想マシン移動処理を実現するため，10 GbE-NIC UZURA[4] 上に実装した RDMA 機能を Xen の仮想マシン移動処理に適用する手法を検討し，実装および評価を行う．まず，従来 TCP/IP 通信によって実装されていた仮想マシンのページ転送処理に対して RDMA 機能を適用し転送時間の短縮を図る．そして，NIC が実行するページ転送処理と CPU が実行するマップ/アンマップ処理を並列に動作させ，オーバラップ化させることで，さらなる転送時間を短縮を図る．また，これらの改良により，仮想マシン上でのアプリケーション動作時において，転送処理が消費する CPU 時間の削減を図る．

本論文では，10 GbE 上の RDMA 転送機能による仮想マシン移動の設計とその評価について述べる．

2. 仮想マシン移動の課題

2.1 解決すべき課題

仮想マシン移動は，異常を検知したサーバから仮想マシンを退避させることによる可用性の向上，保守時に対象サーバから仮想マシンを退避させることによる保守性の向上，高負荷サーバから低負荷サーバへ仮想マシンを退避させることによる動的負荷分散の実現，複数の仮想マシンを一部のサーバに集約させることによる省電力化の実現といった用途に応用できる．

これらの機能をより利用しやすくするためには，以下の 3 つを実現する必要がある．

(1) 転送時間の削減
(2) 転送処理が消費する CPU 時間の削減
(3) 移動時における仮想マシンの停止時間削減

転送時間の削減により，負荷の不均衡を解消するまでの時間や，保守開始や省電力開始までの待ち時間を短縮することができる．また，異常検知から退避完了までの時間短縮により，サーバダウンの可能性が低減する．CPU 時間の削減により，仮想マシン上で動作するアプリケーションに，より多くの CPU 資源を提供でき，アプリケーション性能を改善できる．仮想マシンの停止時間の削減により，仮想マシン上で動作するアプリケーションの停止時間が短縮できる．

これらの課題のうち，(3) の停止時間削減については，文献 5) によると，Xen の仮想マシン移動処理に実装されている Live モード機能により実現されている．そこで，本論文では，(1) 転送時間の短縮と (2) 転送処理の CPU 時間の削減を仮想マシン移動処理において解決すべき課題とする．

2.2 転送時間の短縮

仮想マシン移動処理では大量のページ転送が発生するため，転送時間を短縮するためには，InfiniBand や 10 Gb Ethernet といった高速通信路を使用する必要がある．しかし，単純に通信路を置き換えるだけでは十分に短い転送時間は得られない．

表 1 に示す環境における Gigabit Ethernet（GbE）と 10 GbE のバースト転送時のスループットを**図 1** に，CPU 使用率の内訳を**図 2** に示す．GbE では，ホスト OS による TCP/IP 転送性能を，10 GbE では，TCP/IP 転送性能と UZURA の RDMA 転送性能を測定した．CPU 使用率の測定には，Xen 環境で動作するプロファイラである Xenoprof[6] を用い，図 2 中

表 1 評価環境
Table 1 Measurement environment.

CPU	Opteron 254 (2.8 GHz)
メモリ	4 GB
ページサイズ	4 KB
PCI バス	64 bit 133 MHz PCI-X
GbE	Broadcom NetXtreme BCM5704
10 GbE	UZURA
ホスト OS	Fedora Core 5 (2.6.16.29-xen)
仮想マシンモニタ	Xen 3.0.3-0

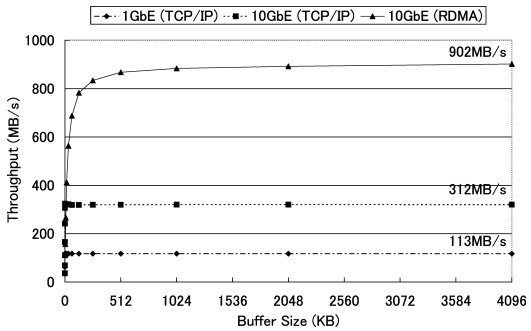


図 1 バースト転送性能
Fig. 1 Performance of burst data transfer.

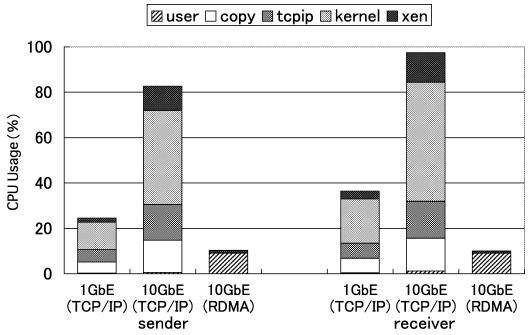


図 2 バースト転送時の CPU 使用率
Fig. 2 CPU usage in burst data transfer.

の「user」は転送性能測定プログラム,「copy」はコピー処理,「tcpip」は TCP/IP 処理,「kernel」はそれ以外のカーネル処理,「xen」は仮想マシンモニタ処理を示している.

GbE では,転送性能の実測値は 113 MB/s であり,ハードウェアの理論性能(125 MB/s)の 90.4%の性能を実現している.このとき,受信側の CPU 使用率は 36.4%である.この性能値をもとに通信路を高速化した場合を考えると,通信性能が 2.75 倍となった時点で,受信側の CPU 使用率が 100%となると推測できる.実際に,10 GbE による TCP/IP 通信では,転送

表 2 2 GB の仮想マシンのページ転送時間(見積り)
Table 2 Elapsed time of page transfer of 2 GB VM (estimation).

処理	時間 (s)
ページ転送時間(TCP/IP)	9.63
データ転送時間(TCP/IP)	6.88
通信以外の処理時間	2.75
データ転送時間(RDMA)	2.38
ページ転送時間(RDMA,見積り)	5.13

性能の実測値は 312 MB/s であり,GbE の通信性能の 2.66 倍で抑えられている.また,受信側の CPU 使用率は 97.4%である.このことから,受信側の CPU 処理がボトルネックとなることが確認できる.

これに対し,NIC 上でプロトコル処理やコピー処理をオフロードする RDMA 転送により CPU のボトルネックを解消すると,転送性能は 902 MB/s となり,ハードウェアの理論性能(1,067 MB/s)☆の 84.6%を達成できる.このように十分に高い転送性能を得るためには,CPU 処理のボトルネック解消が必要である.

しかし,単純に RDMA 機能を仮想マシン移動処理に適用するだけでは十分な高速化は実現できない.単純に通信処理を RDMA 機能によって置き換えた場合の性能を検討するため,表 1 の環境において,10 GbE(TCP/IP)を用いた 2 GB の仮想マシンを移動させた場合のページ転送処理時間を計測し,この値をもとに RDMA 機能を用いた場合の性能を見積もった.**表 2** に見積りを示す.ページ転送処理時間から,TCP/IP による 2 GB のデータ転送時間☆☆を差し引くと,通信以外の処理時間が算出される.この通信以外の処理時間と RDMA による 2 GB のデータ転送時間を加えると RDMA によるページ転送時間は 5.13 s(419 MB/s 相当)となる.これは,通信路の転送性能(902 MB/s)と比較すると,半分以下であり,単純に通信処理を RDMA 機能に置き換えるだけでは,十分な高速化は見込めないことが分かる.

このように,十分に転送時間を短縮するためには,通信処理の高速化のみでなく,通信処理以外の処理時間を削減する必要がある.

2.3 CPU 時間の短縮

仮想マシン上のアプリケーションに対して十分な CPU 時間を提供するために,転送処理が必要とする CPU 時間を削減する必要がある.この転送処理の CPU 時間は,1 ページ転送あたりの CPU 時間と総転送ページ数の積となる.そこで,この両者の削減が必要である.

☆ 10 GbE の理論性能は本来 1,250 MB/s であるが,PCI-X の理論性能に律速されるため 1,067 MB/s となる.
☆☆ スループットから算出.

表 3 1 ページ転送あたりの CPU 時間
Table 3 CPU time per 1 page transfer.

転送方式	TCP/IP	RDMA	相対比
送信側（μs）	10.3	0.0541	190:1
受信側（μs）	12.1	0.0409	296:1

2.3.1 1 ページ転送あたりの CPU 時間の削減

1 ページ転送あたりの CPU 時間は，ページ転送方式に依存する．**表 3** に 10 GbE を用いた場合の 1 ページ転送あたりの CPU 時間☆を示す．1 ページ転送あたりの CPU 時間を比較すると，RDMA 転送は TCP/IP 転送と比較して，送信側では約 1/190，受信側では約 1/296 と圧倒的に低い CPU 時間でページ転送を実現している．したがって，RDMA 適用により 1 ページ転送あたりの CPU 時間は大幅に削減可能である．

一方，1 ページあたりの CPU 時間の削減により，総転送ページ数は若干増加する可能性がある．Xen の Live モードによる仮想マシン移動処理では，転送処理中も仮想マシン上でアプリケーションが動作している．アプリケーションは仮想マシン上のメモリを更新しながら処理を進める．仮想マシン上のメモリはページ単位で管理され，転送もページ単位で行われており，更新したメモリはページ単位で再送される．転送処理の CPU 時間を削減すると，これにともないアプリケーション処理量の増加するため，再送ページ数が増加する．このため，総転送ページ数は増加する．

送信側の 1 ページ転送あたりの CPU 時間は，RDMA 転送の適用により，1/190 に削減されたため，RDMA 適用による総転送ページ数の増加が 190 倍以内であれば，転送処理が必要とする CPU 時間は削減できるといえる．

2.3.2 総転送ページ数の削減

総転送ページ数の削減のためには，再送されるページ数を削減する必要がある．アプリケーションに対しては十分な CPU 時間を提供する必要があるため，転送時間の短縮により，ページの更新を削減し，再送量を削減する必要がある．

このように，転送処理の CPU 時間を削減するためには，総転送ページ数の増加の影響よりも転送処理の CPU 負荷削減の効果の方が高い場合は，RDMA 機能のような転送処理の CPU 負荷を削減する機構を用いる必要がある．さらに CPU 時間を削減するためには，転送時間を短縮する必要がある．

☆ ユーザの CPU 時間を除いたシステムの CPU 時間

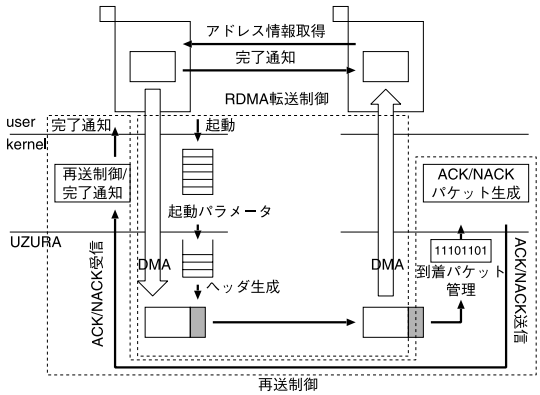


図 3 RDMA 機能の構成
Fig. 3 Structure of RDMA module.

3. 10 GbE-NIC UZURA

3.1 概　　要

UZURA [4] は FPGA 搭載の 10 GbE を用いた実験用 NIC である．この FPGA 上に独自の回路を実装する[7] ことにより，ハードウェア上で様々な機能を実現できる．本論文では，FPGA 上に以下の 2 つの機能を実装し，用いている．

- 通常 NIC 機能
- RDMA 機能

通常 NIC 機能は，通常の Ethernet パケットを送受信する機能を提供し，ホスト OS に実装された TCP/IP プロトコル処理により TCP/IP 通信を実現している．このプロトコル処理を高速化する機能は特に実装されていない．RDMA 機能は，RDMA 転送を実現する機能を提供する．詳細は，3.2 節で説明する．なお，通常 NIC 機能と RDMA 機能は同時に使用することができ，特に転送モードの切替えを行うことなく，1 つの NIC で両方の通信処理を行うことができる．

ホストとのインタフェースは PCI-X（1,067 MB/s）を採用している．このため，NIC 全体の理論性能は，1,067 MB/s に律速されている．

3.2 RDMA 機能

UZURA の FPGA 上には，近接ノード間転送☆☆を目的としたページ転送を高効率に実行する RDMA 機能が実装されている．**図 3** に概略を示す．

Ethernet フレーム上に独自の軽量 RDMA プロトコルを実装しており，UZURA は，コピー処理とプロトコル処理の一部をオフロードする機能を持つ．RDMA 機能は，FPGA とデバイスドライバから構成されており，以下の 2 つの機能を持つ．

☆☆ L3 ルータを経由する広域通信はサポートしていない．

(1) RDMA 転送機能
(2) 再送制御

RDMA 転送機能は，FPGA とアプリケーションの間で転送データを直接転送する機能を提供している．これにより，ゼロコピー通信を実現している．転送処理は，4 KB ページ単位で実行されるため，ページ単位での転送処理を行う場合のデータ転送が最も効率的である．

再送制御は，FPGA による到着パケット管理とデバイスドライバによる ACK/NACK パケット生成/送信，再送/完了通知により構成されている．到着パケット管理を FPGA 上で行うことで，プロトコル処理負荷を削減している．

UZURA を用いた RDMA 転送は，InfiniBand や iWARP[8] における RDMA 転送の手順と同様，以下の手順で実行する必要がある．

(1) 領域情報取得
(2) RDMA 転送
(3) 完了通知

RDMA 転送の起動には，転送先のアドレスに相当する領域情報が必要であるので，起動前にこれを取得する．また，転送元が RDMA 完了後の ACK を受け取ると，この完了通知を転送先に送信する．この完了通知は通常の TCP/IP 通信を用いて行う．

デバイスドライバは，RDMA 転送起動について同期/非同期の両方のインタフェースを提供している．非同期のインタフェースにより，RDMA 転送と CPU 処理のオーバラップ化が可能である．

これらの機能を提供する API を**表 4** に示す．「getreginfo」は，領域情報を取得する API であり，転送領域のアドレスとサイズから領域情報を取得する．「rdmastart」は，RDMA 転送を起動する API であり，領域情報で示された領域のデータを転送する．通信コンテキストを区別することで，複数のアプリケーションが同時に RDMA 機能を使用できる．「poll」は，起動した RDMA 転送の完了を検知する API であり，完了/非完了を戻り値により判別する．このように詳細は異なるものの，基本的には InfiniBand や iWARP が提供する RDMA 機能と同等の API を提供している．

なお，この軽量 RDMA プロトコルではフロー制御は実装されていない．RDMA 転送を行う場合は，転送先領域はあらかじめ確保されている状態で転送処理を行うため，受信バッファが溢れることなく，最大限のスループットで転送可能である．また，経由するスイッチ上のバッファにおいてパケットが溢れる可能性はあるが，これについては，IEEE 802.3x の PAUSE フレームによるフロー制御によって回避している．

表 4 UZURA の RDMA 機能が提供する API
Table 4 RDMA APIs of UZURA.

API 名	引数	説明
getreginfo	addr	転送領域のアドレス
	size	転送領域のサイズ
	info	領域情報
rdmastart	sinfo	転送元領域情報
	dinfo	転送先領域情報
	size	転送サイズ
	cid	コンテキスト番号
	wait	完了待ち
poll	cid	コンテキスト番号
	wait	完了待ち

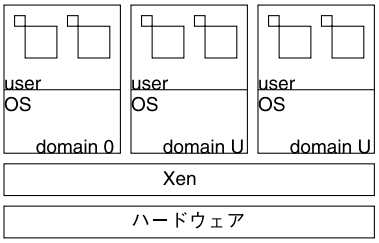


図 4 Xen
Fig. 4 Xen.

4. Xen と仮想マシン移動

4.1 Xen 概 要

Xen は，オープンソースで開発が進められている仮想マシンモニタである．**図 4** に Xen の動作概要を示す．Xen 上には，Domain 0 と Domain U と呼ばれる 2 種類の仮想マシンが動作する．Xen は，ハードウェアの CPU やメモリ資源を管理し，各仮想マシンに割り当て管理する．Domain 0 は，管理用の仮想マシンであり，Xen の起動から終了まで，1 つ必ず動作し，他の仮想マシンの生成，移動，削除を管理する．Domain 0 は他の計算機へ移動できない．Domain U は，Domain 0 上から生成された仮想マシンであり，Xen 上に 0 個以上存在する．Domain 0 の指示により，他の計算機へ移動することができる．

4.2 仮想マシン移動

仮想マシン移動の動作概要を**図 5** に示す．転送元/先の双方の Domain 0 上で動作する転送制御プロセスに転送対象となる仮想マシンのページをマップし，ページ上のデータを転送することにより，仮想マシン移動を実現している．

また，仮想マシンが継続して動作するため，Live モードによる転送処理が実装されている．この Live モードでは，転送処理中も転送対象の仮想マシンを動作させ続け，転送中に更新されたページは再送される．

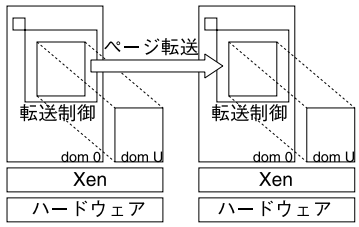


図 5 仮想マシン移動
Fig. 5 Virtual machine migration.

以降，動作の詳細について説明する．仮想マシン移動処理は，大きく以下の処理に分かれる．

(1) 前処理
(2) ページ転送処理
(3) 後処理

前処理は，転送元の Domain 0 上より転送処理が起動されることで開始する．双方で転送制御プロセスを起動し，転送先では，受け入れ用の空の仮想マシンを生成する．

ページ転送処理の動作を**図 6** に示す．ページ転送処理は，通信処理である転送処理と，通信以外の処理であるマップ/アンマップ処理から構成される．これらの処理は，転送制御プロセスが実施し，1,024 ページ単位に行われる．まず，転送元で 1,024 ページを転送制御プロセスのアドレス空間にマップし，その後，転送先で同様にマップする．そして，マップされたページを転送し，双方でアンマップする操作を行う．現在の Xen の実装では，この転送処理はソケット経由で行われ，1 ページずつ 1,024 回に分けて転送している．この操作を，全ページを転送するまで繰り返す．

Live モードで動作する場合は，このページ転送処理中においても転送対象の仮想マシンを動作させている．このため，転送処理の間にページ更新が発生する．この更新されたページは，通常の転送同様に再送される．この再送を下記の条件を満たすまで繰り返す．

(1) 再送ページ数が 50 以下になる．
(2) 繰返し回数が 30 回に達する．
(3) 総転送ページ数が全ページ数の 3 倍を超える．

その後，仮想マシンを停止させ，最後の更新ページを転送する．

後処理では，ページ以外のコンテキスト情報の転送，転送元での仮想マシン削除，転送先での仮想マシン開始を行う．

このように，仮想マシン移動は，大量のページ転送処理が発生するアプリケーションである．この処理は，RDMA 転送が最も得意とする処理であり，RDMA 転送の適用により，高速化の効果が高いと期待できる．

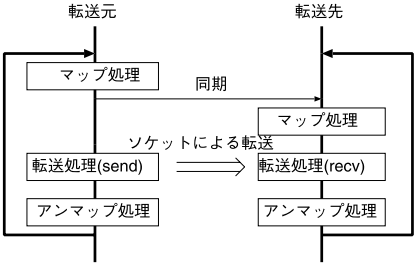


図 6 ページ転送処理
Fig. 6 Process of page transfer.

5. 設　　計

5.1 設計方針

転送時間と転送処理が消費する CPU 時間の削減を目標に仮想マシン移動処理の設計を行う．

転送時間を削減するためには，2.2 節で述べたように，通信処理の高速化と通信以外の処理時間の削減が必要である．そこで，RDMA 機能の適用により，通信処理の高速化を実現し，通信以外の処理であるマップ/アンマップ処理の処理時間削減により，通信以外の処理時間を削減する．

転送処理が消費する CPU 時間を削減するために，2.3 節で述べたように，1 ページ転送あたりの CPU 時間の削減と総転送ページ数の削減が必要である．まず，RDMA 転送の適用により，1 ページ転送あたりの CPU 時間を削減する．RDMA 転送の適用は，総転送ページ数の増加をもたらす負の効果もあるが，4.2 節より，Xen の仮想マシン移動処理では，総転送ページ数は全ページ数の 3 倍以下であるため，2.3.1 項の議論より，この影響を考慮しても，RDMA 転送の適用により，CPU 時間を削減できる．そして，転送時間の短縮により，総転送ページ数の削減する．

したがって，設計方針としては，通信処理には RDMA 機能を用い，マップ/アンマップ処理の処理時間を削減することで，転送時間の短縮と CPU 時間の短縮を実現する．

5.2 RDMA を用いた仮想マシン移動の設計

5.2.1 RDMA の適用

まず，Xen の仮想マシン移動処理の実装に対し，転送処理を RDMA 通信に置き換えることを考える．

Xen の実装では，1 ページずつ転送処理を行っているが，RDMA 転送を用いる場合は，図 1 に示すように，1 ページ（4 KB）ずつの転送では十分な性能に達しないため，マップ/アンマップ処理の単位である 1,024 ページを単位として転送処理を行う．そして，**図 7** のように，ソケット通信を RDMA 転送に置き換

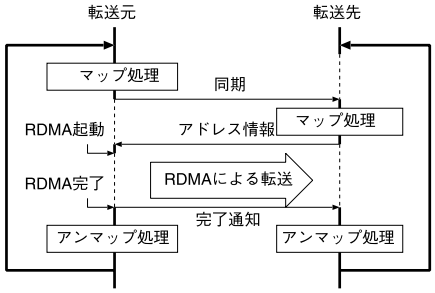


図 7 ページ転送処理への RDMA 転送の適用
Fig. 7 Application of RDMA to page transfer.

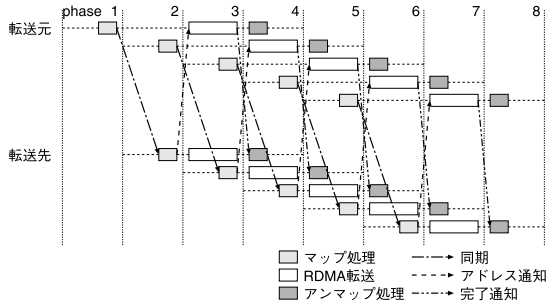


図 8 オーバラップ化
Fig. 8 Overlap method.

表 5 RDMA 適用後の CPU 時間の予測値
Table 5 Estimation of CPU time using RDMA.

	送信側 (s)	受信側 (s)
全体（TCP/IP）	7.66	9.00
通信処理（TCP/IP）	5.53	6.62
通信以外	2.13	2.38
通信処理（RDMA）	0.03	0.02
全体（RDMA）	2.16	2.40

表 6 仮想マシン移動処理の内訳（TCP/IP）
Table 6 Each part of elapsed time in VM migration (TCP/IP).

	送信側	受信側	経過時間 (s)
マップ処理	0.723		0.723
		0.773	0.773
転送処理	6.88		6.88
アンマップ処理	0.509	0.419	0.509
その他	0.740		0.740
合計			9.63

え，RDMA 転送の前後で領域情報取得と転送完了通知を行うことで，RDMA 転送を適用する．この場合，転送単位の変更が必要ではあるものの，変更箇所は，転送処理に関する部分であり，送信側は 37 行，受信側は 101 行の範囲に限られる．

このように RDMA 機能を適用させた場合，表 2 から転送時間は 5.13 s になると予測できる．また，**表 5** に示すように，10 GbE（TCP/IP）による 2 GB の仮想マシン移動処理の CPU 時間を計測し，表 3 より算出される TCP/IP 通信と RDMA 通信の CPU 時間を差し引くと，RDMA によるページ転送処理の CPU 時間は，送信側は 2.16 s，受信側は 2.40 s と予測できる．

5.2.2 オーバラップ化

次に，通信以外の処理時間の削減を検討する．図 7 のように，転送元/先の双方において，マップ処理，転送処理，アンマップ処理は逐次実行されているため，1 ループあたりの転送時間は，各処理時間の和となる．しかし，RDMA 転送中はほとんど CPU を必要としないため，RDMA 転送とマップ処理やアンマップ処理をオーバラップさせ，処理時間の短縮を図る．

このパイプライン化では，ページ転送処理を以下の 4 つのフェーズに分割する．

(1) 転送元マップ処理
(2) 転送先マップ処理
(3) RDMA 転送
(4) 転送元/先アンマップ処理

そして，**図 8** のように，各フェーズにおいて，転送元では，k 番目のページのアンマップ処理と $k+1$ 番目のページの RDMA 転送と $k+3$ 番目のページのマップ処理を実行する．転送先では，k 番目のページのアンマップ処理と $k+1$ 番目のページの RDMA 転送と $k+2$ 番目のページのマップ処理を実行する．

送信側，受信側ともマップ処理，転送処理，アンマップ処理を並列に実行するため，各フェーズごとの変数を用意する必要がある．また，転送処理とマップ/アンマップ処理を並列動作させるよう変更するため，これらの処理を全面的に変更する必要があり，変更作業は比較的難しい．しかし，変更範囲は，送信側は 168 行，受信側は 162 行の範囲に限られ，変更量は少ない．

ここで，設計のため，表 1 の環境において，10 GbE（TCP/IP）を用いて，2 GB の仮想マシンを移動させた場合のページ転送処理の内訳を計測すると各処理の内訳は**表 6** のようになった．この結果より，オーバラップ化させた場合の処理時間を予測すると**表 7** のようになる．RDMA 転送とマップ/アンマップ処理は並列に動作するため，合計時間は 3.12 s であると予測できる．このように，マップ/アンマップ処理を転送処理の背後に隠蔽することで，転送処理時間を削減する．

また，オーバラップ化では CPU 処理自体の削減は行っていないため，CPU 時間については，転送ページ数が同一であれば，オーバラップ化非適用時とほぼ同一になると予測される．

なお，本論文では UZURA が提供する RDMA 機

表 7 仮想マシン移動処理の内訳予測（overlap）
Table 7 Estimation of each part of elapsed time in VM migration (overlap).

	CPU 処理		転送処理	経過時間
	送信側	受信側		(s)
マップ処理	0.723	0.773	2.38	2.38
アンマップ処理	0.509	0.419		
その他		0.740		0.740
合計				3.12

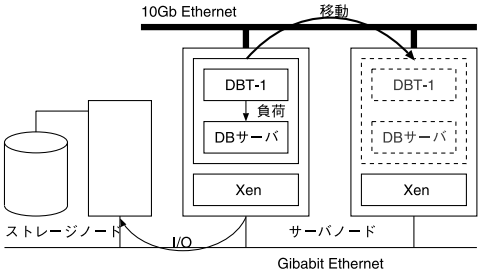


図 9 評価環境の構成
Fig. 9 Structure of measurement environment.

能を適用するが，3.2 節で述べたように，API の機能は基本的には InfiniBand や iWARP と同等であるため同様の手順で InfiniBand や iWARP の RDMA 機能を適用することもできる．

また，仮想マシン上でアプリケーションが動作している場合の性能は，アプリケーションの挙動に依存することから，評価において詳細を解析する必要がある．

6. 評　　価

6.1 評価環境

Xen 仮想マシン移動処理に対し，5 章での設計に基づいた方式を実装し，評価した．評価環境を表 1 および**図 9** に示す．2 つの 10 GbE で接続されているサーバノード間で，仮想マシンの移動処理を行い，性能を評価する．仮想マシンのシステムディスクイメージは，GbE で接続されるストレージノード上にファイルとして格納されており，NBD[9] により各サーバノードへ提供している．

基本性能として，仮想マシン上でアプリケーション動作時の移動処理性能の評価を行い，これに基づき，アプリケーション動作時の評価および解析を行う．

なお，各測定項目について，「TCP/IP」は仮想マシン移動処理にホスト OS による TCP/IP 通信を用いた場合を，「RDMA」は従来方式の TCP/IP 通信を単純に RDMA 機能で置き換えた場合を，「overlap」は設計したオーバラップ化を用いた場合を示している．

6.2 基本性能

基本性能として，アプリケーション非動作時の仮想マシン移動処理の転送時間と CPU 時間を評価した．仮想マシンサイズを 256 MB〜2 GB まで変化させ，転送時間として仮想マシン移動処理におけるページ転送処理の時間を計測し，そのときの CPU 時間を Xenoprof を用いて計測した．

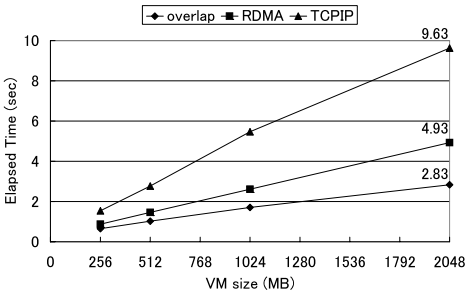


図 10 転送時間
Fig. 10 Elapsed time of VM migration.

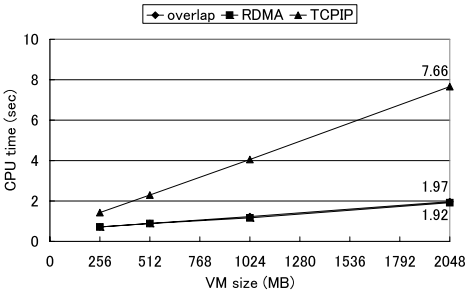


図 11 CPU 時間（送信側）
Fig. 11 CPU time (sender side).

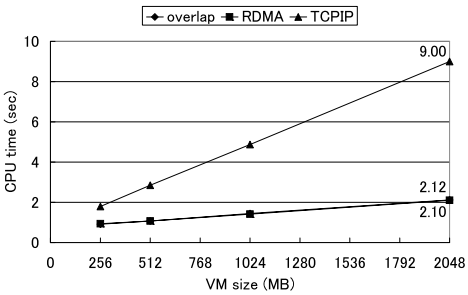


図 12 CPU 時間（受信側）
Fig. 12 CPU time (receiver side).

転送時間を**図 10** に，CPU 時間を**図 11** および**図 12** に示す．また，転送時間から算出した転送スループットを**図 13** に示す．

「RDMA」と「overlap」について，実測値と，5.2 節で議論した表 2 と表 7 に示す転送時間および表 5 に示す CPU 時間の予測を比較すると，ほぼ一致する．したがって，設計どおりの性能が達成できたといえる．

また，「RDMA」は，「TCP/IP」と比較し，転送時間，CPU 時間をともに削減している．これは，NIC

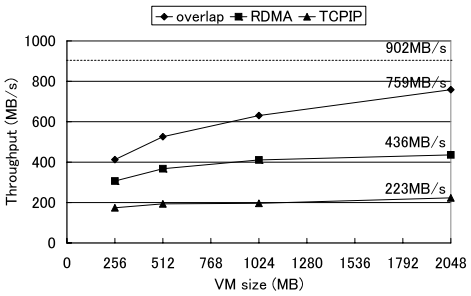


図 13 転送スループット
Fig. 13 Throughput of data transfer.

上でのプロトコル処理とコピー処理のオフロードにより転送性能が向上し，かつ，CPU 使用率を削減できたためである．しかし,「RDMA」の転送スループットは最大 436 MB/s であり，通信路のスループット（902 MB/s）の 48.3%と，十分に高いとはいえない．

これに対し,「overlap」は,「RDMA」と比較し，転送時間をさらに削減しており，転送スループットは最大 759 MB/s と「RDMA」の 1.74 倍を達成している．これは，通信路が提供するスループットの 84.1%であり，十分に高速な転送が実現できているといえる．

さらに図 10 におけるグラフの傾きから，仮想マシンサイズが増加すると転送時間の差が拡大し，性能の改善効果は高くなるといえる．また，性能の改善効果は，仮想マシンの個数に比例して増加する．このため，サーバの搭載メモリ量の増加により，より大きな仮想マシン，または，より多くの仮想マシンを扱う場合には，性能改善の効果がより高まると期待できる．

なお,「RDMA」と「overlap」では，CPU 処理自体は同等の処理を行っており，転送ページ数も同じであるため，CPU 時間はほぼ同一である．

6.3 アプリケーション動作時の性能

6.3.1 概　　要

仮想マシン上でアプリケーションを動作させた場合の性能評価を行った．アプリケーションとしては，電子商取引システムや基幹業務システムにおいて重要なコンポーネントの 1 つである DB サーバを対象とした．

DB サーバを仮想マシン上で動作させ，これに対し，DBT-1 ベンチマークにより負荷を与える．DB サーバとしては，PostgreSQL 7.4.14 を用いる．DB サーバがアクセスするデータベース本体をストレージノード上の RAM ディスク上に展開し，これを NBD により各サーバノードへ提供している．このため，ディスク I/O は発生しない．また仮想マシンサイズは 2 GB としている．DBT-1 ベンチマークは，Web オンライン書店のデータベースへの負荷をシミュレートしたベンチマークであり，仮想ユーザ数（eus）に比例した負荷を与えることができる．eus 以外の DBT1 および PostgreSQL の設定パラメータを**表 8** に示す．これ以外はデフォルト値を用いている．

表 8 DBT1 および PostgreSQL のパラメータ
Table 8 Parameters of DMT1 and PostgreSQL.

	パラメータ	値
DBT1	rconnection	240
	think_time	0.5
	duration	100
PostgreSQL	max_connections	512
	shared_buffers	10000

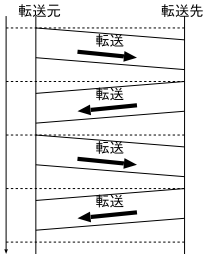


図 14 仮想マシン移動パターン
Fig. 14 Pattern of VM migration.

評価では，DB サーバが与える移動処理性能への影響を評価するため，DB サーバ稼働中の仮想マシン移動処理性能の評価を行う．

また，移動処理が与える DB サーバの性能への影響を評価する．移動処理中，DB サーバが使用できる CPU 時間は，全体の CPU 時間から移動処理が消費する CPU 時間を差し引いたものとなる．この影響を評価するため，一定期間の間に一定回数の移動処理を行った場合の性能を比較する．具体的には，DB サーバが稼働する仮想マシンを**図 14** に示すように，2 ノード間で一定間隔で転送させた場合の転送性能および DB サーバのスループット（BT/s: Bogo Transaction per sec）を評価する．

また，転送間隔を，10 s，15 s とする．これは，数十台の物理サーバ上にそれぞれ 10 台の仮想マシンが配置される状況において，省電力化のため，10 分間隔で負荷に応じて仮想マシンの集約を行う場合を想定したためである．この場合，仮に全体負荷が 1/5 になると，物理サーバ 1 台あたりの仮想マシン数を 5 倍に集約するため，10 分間に 40 台が移動する．つまり 1 台あたり 15 s の移動時間となる．

この評価では，DB サーバが転送処理中に更新するページ量が大きく性能に影響する．このため，まず，DB サーバによる更新ページ量の評価を行う．次に，転送性能の評価として，転送時間と移動処理が消費す

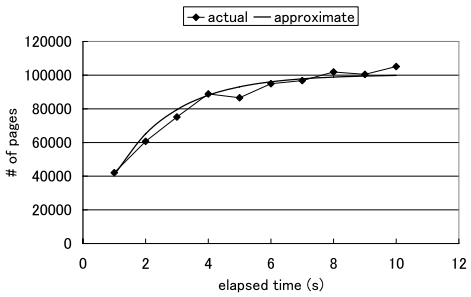


図 15 更新ページ数
Fig. 15 Number of dirty pages.

る CPU 時間を評価および解析を行う．そして，アプリケーション性能の評価として，DB サーバのスループットの評価および解析を行う．

6.3.2 仮想マシン上のページ更新性能

DB サーバによる更新ページ量を評価した．eus は 300 とし，更新時間を変化させた場合の更新ページ数を測定した．**図 15** は，横軸に示す時間の間に DB サーバが 1 回以上更新したページ数を示している．この間に同じページが複数回更新されても 1 回とカウントする．

時間経過が長くなると，DB サーバがページを更新する際に，以前に更新したページを再更新する可能性が高くなる．また，DB サーバが更新する領域は一定量に限られる．このため，時間経過とともに，更新ページ数の増加は少なくなり，一定数に近付くようになる．

仮に，DB サーバが一定数 M のページを完全にランダムに更新しているとすると，時間経過 t と更新ページ数 D は以下の式で表せる．

$$D = M - M\left(\frac{1}{2}\right)^{\frac{t}{\tau}}$$

そこで，測定値から最小二乗法により M および τ を求めると $(M, \tau) = (1.00 \times 10^5, 1.32)$ となる．

6.3.3 転送性能の評価

DB サーバが稼働している 2 GB の仮想マシンを転送させたときの転送時間，転送スループット，転送処理中の CPU 時間を**表 9** に示す．表中における括弧内の数値は，6.2 節で示した仮想マシン上でアプリケーション非動作時の転送時間および CPU 時間を基準とした比率である．また,「RDMA」と「overlap」を比較した場合の性能改善率を**表 10** に示す．

アプリケーション非動作時と比較すると，アプリケーションが動作することによる性能劣化の割合は転送時間，CPU 時間ともに,「RDMA」よりも「overlap」の方が低く抑えられている．

表 9 アプリケーション動作中の転送性能
Table 9 Performance during application program running.

	TCP/IP	RDMA	overlap
転送時間（s）	10.8(1.12)	6.40(1.30)	3.15(1.10)
CPU 時間（送）	8.67(1.13)	2.65(1.38)	2.34(1.19)
CPU 時間（受）	9.54(1.06)	2.52(1.20)	2.30(1.08)
スループット（MB/s）	200(0.92)	336(0.77)	681(0.91)

表 10 転送性能の改善率
Table 10 Improvement ratio of performance.

	RDMA(s)	overlap(s)	改善率 (%)
転送時間	6.40	3.15	50.8
CPU 時間（送）	2.65	2.34	11.7
CPU 時間（受）	2.52	2.30	10.0

アプリケーション動作時の「RDMA」と「overlap」を比較すると，アプリケーション非動作時と同様,「overlap」の方が転送時間を大きく削減している．また,「overlap」の方が CPU 時間を短縮できている．

複数の仮想マシンを転送させる場合，転送時間と CPU 時間の短縮効果は転送させる仮想マシンの個数に比例して増加する．したがって，サーバの搭載メモリ量の増加により，より多くの仮想マシンを転送させる場合には，さらに性能改善の効果が高まると期待できる．

6.3.4 転送性能の解析

アプリケーション非動作時と比較すると，各場合において転送時間と CPU 時間は増加し，転送スループットは減少している．これは，転送処理中に仮想マシン上のページが更新されたページを再送するためである．そこで，6.3.2 項でのページ更新性能とアプリケーション非動作時の転送性能からアプリケーション動作時の再送ページ量および転送時間を算出した．転送処理中は，仮想マシン上のアプリケーションが使用できる CPU 時間は，全体から転送処理が必要とする CPU 時間を差し引いた時間となる．したがって，転送処理動作中の CPU 使用率を β とすると t 秒間に更新されるページ数 D は，式 (1) で表される．

$$D = M - M\left(\frac{1}{2}\right)^{\frac{t(1-\beta)}{\tau}} \quad (1)$$

そこで，**表 11** に示す転送時間と CPU 時間から算出した CPU 使用率と図 10 に示すアプリケーション非動作時の転送時間から，1 回目のページ転送処理中に更新されたページ数を算出する．再送により生じる 2 回目以降のページ転送も基本的には同様に計算を行う．ただし，CPU のキャッシュの影響により 2 回目以降の再送処理は 1 回目より高速となるため，2 回目

表 11 転送処理の CPU 使用率（%）
Table 11 CPU usage of transfer processing.

	TCP/IP	RDMA	overlap
送信側	80.6	42.3	74.2
受信側	88.3	39.4	75.6

表 12 2 回め以降の転送時間
Table 12 Elapsed time after 2nd times.

仮想マシンサイズ	32 MB	64 MB	128 MB	256 MB
TCP/IP (ms)	157.7	313.9	596.1	122.6
RDMA (ms)	95.3	160.7	280.6	588.9
overlap (ms)	70.8	111.5	184.6	325.2

表 13 2 回め以降の転送時間の近似式
Table 13 Approximation of elapsed time after 2nd times.

	近似式
TCP/IP	$y = 4.76 \times 10^{-3}x + 2.50 \times 10^{-3}$
RDMA	$y = 2.21 \times 10^{-3}x + 1.66 \times 10^{-2}$
overlap	$y = 1.13 \times 10^{-3}x + 3.76 \times 10^{-2}$

x:転送サイズ（MB），y:転送時間（s）

以降の転送時間を計測し，再送処理の転送時間の近似式を算出し，これを算出に用いる．2 回目以降の転送時間と近似式を**表 12** と**表 13** に示す．CPU 使用率については 1 回目と同一であると仮定する．転送処理は，4.2 節で示した再送の停止条件を満たすまで繰り返すと仮定する．

このようにして算出した更新ページ数および転送時間を**表 14** に示す．表 9 の転送時間の実測値と比較すると，おおむね結果が一致することから算出値の信頼性が確認できる．

また,「overlap」は,「RDMA」と比較して，総転送ページ数を 13.3%削減できている．これは，表 10 に示す CPU 時間の削減率とほぼ一致することから，総転送ページ数の削減により CPU 時間が削減できたといえる．

更新ページ数が削減できた要因として，式 (1) より，オーバラップ化による転送時間短縮の効果とアプリケーションが使用する CPU 使用率の低下が考えられる．そこで，CPU 使用率の低下による影響を排除して検証するため，アプリケーションが 100%CPU を使用できると仮定した場合の各場合の転送時間および更新ページ数を算出した．算出結果を**表 15** に示す．算出結果の「overlap」と「RDMA」を比較すると，アプリケーションが 100%CPU を使用した場合でも，転送時間を 55.5%削減しており，これにより，更新ページ数を 43.8%削減している．このことから，オーバラップ化による転送時間短縮の効果により更新ページ数が削減できることが確認できる．また，総転送ページ数

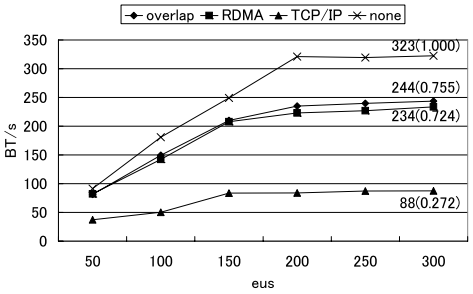


図 16 DB サーバ性能（15 s 間隔）
Fig. 16 Performance of DB server (15 s interval).

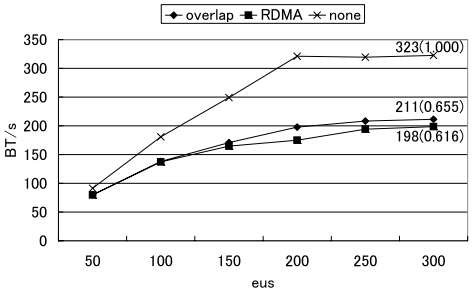


図 17 DB サーバ性能（10 s 間隔）
Fig. 17 Performance of DB server (10 s interval).

は 11.9%削減していることから，CPU 時間も同程度削減できると予測できる．

そして，メモリを増加させ，さらに大規模な DB サーバを構築した場合においては，M の値が増加する．仮に M の値が 2 倍，4 倍とした場合を想定して，転送時間と更新ページ数を算出すると，**表 16** のようになる．表 16 より，M の増加に対し，転送時間と更新ページ数の削減率が増加することから，大規模なサーバでは「overlap」の優位性がより高まると推測できる．

6.3.5 アプリケーション性能の評価

2 つのサーバノード間で仮想マシンを転送させた場合の仮想マシン上の DB サーバ性能を評価した．転送間隔を 15 s，10 s とした場合の各場合について評価を行った．なお,「TCP/IP」では転送時間が 10 s を超えるため 15 s とした場合のみ評価を行っている．

転送間隔を 15 s とした場合の性能を**図 16** に，10 s とした場合の性能を**図 17** に示す．なお，図中の「none」は，仮想マシンを移動させない場合の性能値を示しており,「none」に対する相対性能を括弧内に示している．各場合とも，グラフの概形から，おおむね eus が 300 程度で最大性能に達している．そこで，以降の議論では，eus が 300 の場合について述べる．

転送間隔 15 s の場合,「TCP/IP」は 88 BT/s であるのに対し「RDMA」は 234 BT/s であり，2.68 倍の

表 14 更新ページ数と転送時間（算出値）
Table 14 Number of dirty pages and elapsed time (calculated).

	TCP/IP		RDMA		overlap	
回数	転送時間（s）	更新ページ数	転送時間（s）	更新ページ数	転送時間（s）	更新ページ数
1	9.63	62,680	4.92	77,686	2.86	32,148
2	1.17	11,424	0.69	18,799	0.17	2,218
3	0.21	2,136	0.18	5,277	0.01	202
4 回以降	0.06	617	0.66	20,015	0.11	1,489
合計	11.07	76,675	6.45	121,777	3.15	36,055
転送回数		7		30		30
総転送ページ数		600,963		646,005		560,283

表 15 アプリケーションの CPU 使用率を 100%とした場合
Table 15 The case that CPU usage of application is 100%.

	TCP/IP		RDMA		overlap	
回数	転送時間（s）	更新ページ数	転送時間（s）	更新ページ数	転送時間（s）	更新ページ数
1	9.63	99,731	4.92	92,750	2.86	77,899
2	1.86	62,409	0.82	34,911	0.40	18,763
3	1.16	45,775	0.32	15,380	0.10	5,023
4 回以降	7.12	339,220	1.03	53,363	0.17	8,933
合計	19.76	547,135	7.08	196,405	3.52	110,618
転送回数		30		30		30
総転送ページ数		1,071,423		720,633		634,846

表 16 M の値が増加した場合（予測値）
Table 16 Performance of various value of M (estimated).

	RDMA		overlap		削減率	
M の値	転送時間（s）	更新ページ数	転送時間（s）	更新ページ数	転送時間	更新ページ数
1.00×10^5	6.45	121,777	3.15	36,055	51.2%	70.4%
2.01×10^5	8.35	344,300	3.66	77,619	59.8%	77.5%
4.02×10^5	29.53	2,860,490	3.88	182,570	86.9%	93.6%

性能を達成している．また,「overlap」は 244 BT/s であり,「RDMA」より 4.3%改善されている．また，転送間隔 10 s の場合,「RDMA」は 198 BT/s であるのに対し,「overlap」は 211 BT/s であり，6.4%改善している．

6.3.6 アプリケーション性能の解析

詳細な解析を行うため，各場合の転送間隔あたりの domU の CPU 時間を計測する．また，転送完了後，仮想マシン起動準備処理のため，仮想マシンが停止させられる．この休止時間を計測したところ DB サーバ稼働中においては，各場合とも，平均 1.35 秒であった☆．これらの計測結果と転送処理の CPU 時間を転送間隔 15 s については**表 17** に，転送間隔 10 s については**表 18** に示す☆☆．なお，括弧内に全体に対する各 CPU 時間の割合を示している．

「none」の場合は，ほぼ CPU を 100%使用して DB サーバを実行している．これに対し，転送をともなう各場合は，転送処理および休止時間の分を差し引いた CPU 時間により DB サーバを実行しているため，性能劣化する．

図 16，図 17 と表 17，表 18 を比較すると，転送間隔 15 s の場合および 10 s の場合について，domU の CPU 時間の割合と「none」に対する DB サーバの性能比はほぼ一致することが分かる．

このことから,「overlap」が「RDMA」より高い性能を示すのは，転送処理の CPU 時間の削減分を DB サーバ実行が利用できたためであるといえる．

また，6.3.4 項で述べたように，さらに大規模なサーバでは「RDMA」に対する「overlap」の転送時間および更新ページ数の削減率が増加するため,「RDMA」と「overlap」の転送処理が必要とする CPU 時間の差は拡大すると推測できる．したがって，サーバの規模が大きくなるほど,「overlap」の「RDMA」に対するアプリケーション性能の改善率は改善されると推測で

☆ 文献 5) によると，2005 年の実装では，Live モード使用時の休止時間は 60 ms と報告されており，本評価と大きく異なる．Xen-3.0.3 への改版で，転送元のデバイスが完了後に転送先で仮想マシンを起動するように変更されたことが主な原因である．RDMA 機能やオーバラップ化により生じたものではない．

☆☆ 両者とも合計がそれぞれ 15 s，10 s となっていないが，これは様々な外乱により，転送間隔が若干増減するためである．

表 17 15 s あたりの各部の時間（s）
Table 17 Elapsed time of each part in 15 s.

	none	TCP/IP	RDMA	overlap
domU	14.69	4.95	10.93	11.46
	(1.000)	(0.331)	(0.732)	(0.756)
転送処理	—	8.67	2.65	2.34
		(0.579)	(0.177)	(0.145)
休止	—	1.35	1.35	1.35
		(0.090)	(0.090)	(0.89)
合計	14.69	14.97	14.93	15.15

表 18 10 s あたりの各部の時間（s）
Table 18 Elapsed time of each part in 10 s.

	none	RDMA	overlap
domU	9.79	6.28	6.59
	(1.000)	(0.611)	(0.641)
転送処理	—	2.65	2.34
		(0.258)	(0.228)
休止	—	1.35	1.35
		(0.131)	(0.131)
合計	9.79	10.28	10.28

きる．

7. 関 連 研 究

RDMA 転送を用いた高速化に関しては様々な研究がなされている．特に文献 10) では，本論文と同様にオーバラップ化により高速化を行っている．文献 10) では，InfiniBand の RDMA 転送の際に必要となる領域登録処理を RDMA 転送のバックグラウンドで実行することで CPU 処理である領域登録処理時間を隠蔽している．CPU 処理を RDMA 転送とオーバラップさせる点において本論文と共通しているが，文献 10) では，通信処理に付随する領域登録処理のみを隠蔽しているのに対し，本論文では，アプリケーション処理であるページマップ処理もオーバラップさせることで，さらに多くの CPU 処理を隠蔽している点が異なる．

仮想マシン移動に関する事例として，以下の事例が報告されている．文献 5) では，Xen の仮想マシン移動の詳細について報告されており，単位時間あたりの転送量を調整する検討がなされている．しかし，GbE を通信路として用いられており，オーバラップ化のような転送時間短縮手法は実施されていない．

Mellanox 社のプレスリリース[11] では，InfiniBand RDMA 機能による仮想マシン移動について言及されている．しかし，性能や実装方式の詳細は報告されていない．

その他の仮想マシン移動の事例として，VMWare[1] の VMotion や，Qemu[12] を用いて実装された Quasar[13] がある．いずれの事例においても，現時点では RDMA 機能は適用されていない．

8. お わ り に

本論文では，10 Gb Ethernet 上の RDMA 転送機能による仮想マシン移動の設計と評価について述べた．

高速かつ低負荷な仮想マシン移動を実現するため，10 GbE-NIC UZURA 上に実装した RDMA 機能を Xen の仮想マシン移動処理に適用する手法を検討し，実装および評価を行った．RDMA 機能の適用に際し，単純に通信部分を RDMA 転送に置き換えるだけでなく，NIC が実行するページ転送処理と CPU が実行するマップ/アンマップ処理を並列に動作させ，オーバラップ化することで，転送時間を短縮を図った．また，仮想マシン上でのアプリケーション動作時において，転送処理が消費する CPU 時間の削減を図った．

RDMA 機能適用の結果，2 GB の仮想マシンの転送において TCP/IP による転送時と比較して，アプリケーション非動作時において転送時間を 48.8%削減し 4.92 s（436 MB/s 相当）を，アプリケーション動作時において転送時間を 40.7%削減し 6.40 s（336 MB/s 相当）を達成した．また，転送処理が消費する CPU 時間をアプリケーション非動作時においては最大 76.7%，アプリケーション動作時においては最大 73.6%削減し，CPU 時間の削減により，仮想マシン上で動作するアプリケーション性能を最大 2.68 倍に改善した．

さらに，オーバラップ化を適用した結果，オーバラップ化非適用時と比較して，アプリケーション非動作時において転送時間を 42.6%削減し 2.83 s（752 MB/s 相当）を，アプリケーション動作時において転送時間を 50.8%削減し 3.15 s（681 MB/s 相当）を達成した．また，アプリケーション動作時において，転送処理中のページ更新量を解析した結果，オーバラップ化による転送時間短縮により更新ページ数を削減できることを確認した．これにより，転送処理が消費する CPU 時間を最大 11.7%削減し，仮想マシン上で動作するアプリケーション性能を最大 6.4%改善した．

RDMA 機能適用とオーバラップ化の適用により，仮想マシン移動処理時間を大幅に削減し，転送処理がアプリケーション性能に与える影響を軽減できた．これにより，大規模な多ノードサーバシステム上においてより柔軟かつ容易な保守管理が実現可能になった．

残された課題として，Web サーバや科学技術計算アプリケーションにおける本転送方式の評価や，本転送方式を用いた異常ノードからの退避機構や，保守容易なサーバ管理機構，省電力化機構，動的負荷分散機構の実現がある．

参 考 文 献

1) VMWare, VMware. http://www.vmware.com
2) Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, L. and Warfield, A.: Xen and the art of virtualization, *Proc. ACM Symposium on Operating Systems Principles* (2003).
3) InfiniBand Architecture Specification. http://www.infinibandta.org
4) 中島耕太，佐藤　充，住元真司，久門耕一，石川　裕：高性能通信処理オフロードエンジンUZURA 実現に向けて，情報処理学会研究会報告，Vol.2005, No.81(2005-HPC-103), pp.103–108 (2005).
5) Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I. and Warfield, A.: Live Migration of Virtual Machines, *Proc. 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation* (*NSDI*), Boston, MA, pp.273–286 (2005).
6) Menon, A., et al.: Diagnosing Performance Overheads in the Xen Virtual Machine Environment, *Proc. 1st ACM/USENIX International Conference on Virtual Execution Environments* (2005).
7) 中島耕太，佐藤　充，後藤正徳，住元真司，久門耕一，石川　裕：配列転置データ転送を高速化する 10 Gb Ethernet インタフェースカードの設計，情報処理学会論文誌：コンピューテングシステム，Vol.47, No.SIG12(ACS15), pp.74–85 (2006).
8) Hilland, J., Culley, P., Pinkerton, J. and Recio, R.: RDMA protocol verbs specification (v1.0) (2003). http://www.rdmaconsortium.org
9) Machek, P., et al.: Network Block Device. http://nbd.sourceforge.net
10) Shipman, G.M., Woodall, T.S., Bosilca, G., Graham, R.L. and Maccabe, A.B.: High Performance RDMA Protocols in HPC, *13th European PVM/MPI Users' Group Meeting* (2006).
11) Mellanox Technologies: Mellanox and Novell Drive High Bandwidth Virtualization Into Data Centers, Press Release (May 2006). http://www.mellanox.com/news/press_releases/pr_050306.php
12) QEMU CPU Emulator. http://fabrice.bellard.free.fr/qemu/
13) 尾上浩一，大山恵弘，米澤明憲：Quasar: CPU エミュレータ QEMU を利用した移動計算システム，第 8 回プログラミングおよび応用のシステムに関するワークショップ，pp.84–92 (March 2005).

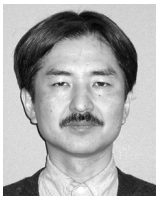
(平成 19 年 5 月 7 日受付)
(平成 19 年 8 月 22 日採録)

中島　耕太（正会員）

2000 年九州大学工学部電気情報工学科卒業．2002 年同大学大学院システム情報科学府情報工学専攻修士課程修了．同年富士通（株）入社．現在，(株) 富士通研究所勤務．高速通信機構に関する研究開発に従事．

佐藤　　充（正会員）

1969 年生．1992 年東京大学工学部電気工学科卒業．1997 年同大学大学院工学系研究科情報工学専攻博士課程修了．博士（工学）．同年富士通（株）入社．現在，(株) 富士通研究所勤務．並列システムアーキテクチャの研究に従事．IEEE，ACM 各会員．



久門　耕一（正会員）

1979 年東京大学工学部電気工学科卒業．1981 年同大学大学院電子工学専門課程修士課程修了．1984 年同大学院博士課程中退．同年（株）富士通研究所入社．現在，同社 IT システム研究所に所属．CPU，メモリ，並列計算機アーキテクチャに関する研究に従事．GCC，Linux カーネル等の改良にも興味を持つ．

谷口　秀夫（正会員）

1978 年九州大学工学部電子工学科卒業．1980 年同大学大学院修士課程修了．同年日本電信電話公社電気通信研究所入所．1987 年同所主任研究員．1988 年 NTT データ通信（株）開発本部移籍．1992 年同本部主幹技師．1993 年九州大学工学部助教授．2003 年岡山大学工学部教授．博士（工学）．オペレーティングシステム，実時間処理，分散処理に興味を持つ．著書『オペレーティングシステム』（昭晃堂）等．電子情報通信学会，ACM 各会員．