社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

10Gb Ethernet 上の RDMA を用いた遠隔スワップメモリの実装

後藤　正徳†　　佐藤　　充†　　中島　耕太†　　久門　耕一†

† 株式会社 富士通研究所　〒 211–8588 神奈川県川崎市中原区上小田中 4–1–1
E-mail: †{gotom,msato,kota,kumon}@labs.fujitsu.com

あらまし　近年の PC クラスタで使用されているインタコネクトは高性能化し，メモリバンド幅のスループット性能に近付きつつある．そこで，我々は高速インタコネクトを使用し，遠隔ノードのメモリをスワップとして用いる遠隔スワップメモリ技術の実現可能性を検討している．評価を行うために，我々は遠隔スワップメモリシステム Nuzura を実装した．Nuzura は 10Gb Ethernet 上で RDMA を実現する NIC UZURA と，これを用いたネットワークブロックデバイス RNBD をスワップデバイスとして用いる．評価に際しては，本システム上で搭載メモリの数倍を要求する複数の HPC アプリケーションを実行し，性能を測定した．実験結果から，アプリケーションのメモリアクセスパターンや遠隔スワップメモリの使用比率に応じて性能オーバヘッドが異なることを示した．また，ページ置換方式の変更によって姫野ベンチマークの性能が 4 倍近く向上することを示した．

キーワード　スワップ，遠隔スワップメモリ，PC クラスタ，10Gb Ethernet，RDMA，ブロックデバイス

Implementing Remote Swap Memory using RDMA over 10Gb Ethernet

Masanori GOTO†, Mitsuru SATO†, Kohta NAKASHIMA†, and Kouichi KUMON†

† Fujitsu Laboratories LTD.　1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki, 211–8588 Japan
E-mail: †{gotom,msato,kota,kumon}@labs.fujitsu.com

Abstract　In recent years, interconnects on PC clusters have exploited higher performance. Its throughput closes to memory bandwidth. We examine the availability of remote swap memory technology using remote node memory as swap via high-speed interconnects. We implemented to evaluate a remote swap memory system Nuzura. It uses swap devices provided by the 10Gb Ethernet NIC UZURA with RDMA and the network block device RNBD. We measured the performance of 4 HPC applications which required several times of local memory on the system. The result showed the overhead was varied depended on the application memory access pattern and the usage of remote swap memory. We also show that the performance of Himeno benchmark can be improved up to 4 times by changing page replacement method.

Key words　Swap, Remote Swap Memory, PC Cluster, 10Gb Ethernet, RDMA, Block Device

1. はじめに

ノード間を高速インタコネクトで接続した PC クラスタは，安価・高性能という点から大規模数値計算 (HPC) 分野で幅広く普及している．しかし，複数ノードにメモリ資源が分散しているため，例えば各 4GB メモリを有する 8 ノードを用意しても 32GB メモリが必要なアプリケーションを実行できない．

アプリケーションが利用可能なメモリを増やす方法として，ハードディスクベースのスワップが利用されてきた．しかし，ハードディスクは機械的動作を伴うことからランダムアクセス性能が低く，メモリの代替として使用する局面は限られている．

ところで，高速インタコネクトのスループット性能はメモリバンド幅に近付きつつある．例えば，10GbE (10Gb Ethernet) のスループットは 1.25GB/s，InfiniBand は 1GB/s または 2GB/s であるのに対し，Opteron に接続したローカルメモリは 6.4GB/s または 8.0GB/s である．特に，ネットワークスループットの伸び率はメモリと比較して大きいと予測されており，この差は今後さらに小さくなる可能性がある [3].

そこで，我々は高速インタコネクトを利用した遠隔スワップメモリシステムの実現可能性について検討している．その評価を行うにあたり，我々は遠隔スワップメモリシステム Nuzura を実装した．Nuzura システムは複数の計算ノードに存在する遠隔メモリを高速インタコネクト経由でアクセスし，スワップデバイスとして用いる．そして，計算ノードが持つメモリを越えるメモリサイズを要求するアプリケーションを動作させるものである．

本論文では，Nuzura システムで使用する RDMA (Remote Direct Memory Access) 機能付き NIC (Network Interface Card) と，それを用いたネットワークブロックデバイスの実装について述べる．さらに，Nuzura システム上で複数の HPC アプリケーションを計測し，高速インタコネクト経由の遠隔スワップメモリの実用性について評価する．また，アプリケーションのメモリアクセスパターンを考慮し，遠隔スワップメモリを用いた場合に性能向上するページ置換方式について考察を行う．最後に以上の結果をまとめる．

2. 関 連 研 究

遠隔スワップメモリに関しては Network RAM Disk [1] や Nswap [9] などの研究が行われている．これらはスワップデバイスとしてネットワークに結合したブロックデバイスを実装している点で，我々のアプローチと似ている．しかし，実験は 100Base-T といった遅いネットワークを使用しており，単純なベンチマークによって評価しているにとどまる．このため近年の高速インタコネクトを用いた遠隔スワップメモリの実用性を示していない．

高速インタコネクトを用いて遠隔スワップメモリを実装している研究の 1 つに InfiniBand を使用した HPBD がある [5]. HPBD では最大 16 台の遠隔スワップメモリを利用して qsort の性能を評価している．しかし，HPBD では InfiniBand の RDMA 性能を出すために事前登録済のメモリプールを用いたアロケータを使用していること，また全体の実行時間に対して遠隔スワップメモリを頻繁に使用しないアプリケーションである qsort による測定を行っていることから，ここから実用性を判断することは難しい．

3. 遠隔スワップメモリの概念

遠隔スワップメモリの概念を図 1 に示す．本論文では各ノードは独立した計算機であり，高速インタコネクトといったネットワークで接続されているものとする．また，ネットワーク越しにアクセスする他ノードのメモリを遠隔メモリ，遠隔メモリを用いたスワップデバイスを遠隔スワップメモリと呼ぶ．そして，遠隔メモリを提供する各ノードをサーバとし，サーバの遠隔スワップメモリを利用するノードをクライアントと呼ぶ．クライアントに搭載のメモリをクライアント物理メモリと呼ぶ．

クライアントの OS はシステムのメモリが足りないことを検出すると，使用頻度の低いページをサーバの遠隔メモリへスワップアウトして空きメモリを確保する．また，クライアントの OS はアプリケーションが以前スワップアウトしたページへアクセスしたことを検出するとアプリケーションの実行を一旦中断し，サーバの遠隔メモリから当該ページをメモリへ転送してからアプリケーションの動作を再開させる．スワップ処理はページサイズ単位（IA-32 では 4KB）で行う．

スワップ処理は OS 内でソフトウェア実装されているため，実行に際してオーバヘッドが大きい．さらに，一般に高速インタコネクトはスループット性能（〜2GB/s）・レイテンシ性能（数百 ns〜数十 μs）ともメモリより遅い（それぞれ〜10GB/s, 50〜200ns）．このためメモリアクセスに時間的・空間的局所性が全くなく，アクセス毎にスワップ処理が入る場合には，ソフトウェアオーバヘッドとハードウェアレイテンシの時間が大きな性能ペナルティとなる．

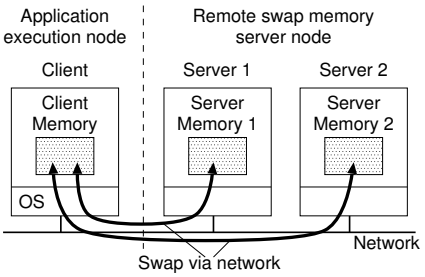

図 1　遠隔スワップメモリの概念
Fig. 1　Concept of remote swap memory

しかし，現実のアプリケーションはメモリアクセス局所性を持ち，スワップ発生後に再びスワップが発生する間隔はアプリケーションのメモリアクセスパターンによって変化する．また，高速インタコネクトのレイテンシやスループット特性，OS のスワップ処理にかかる時間によっても性能は変わってくる．

そこで，我々は本論文においてクライアント物理メモリと遠隔スワップメモリの使用比率を変化させ，性質の異なる複数のアプリケーションを評価することで，その実用性を検証する．

4. 遠隔スワップメモリシステム Nuzura の実装

4.1 遠隔スワップメモリシステムの構成

我々が開発した遠隔スワップメモリシステム Nuzura の実装を図 2 に示す．図 2 では本システムを構成するクライアントとサーバが各 1 台示されている．それぞれのノードでは Linux カーネルとともに以下に述べるコンポーネントが動作する．

- **管理デーモン** … サーバ・クライアント間の通信を管理する．クライアントの管理デーモンは起動されるとサーバの管理デーモンへ問い合わせを行う．すると，サーバの管理デーモンは一定量事前確保した遠隔スワップメモリ領域の物理アドレス情報をクライアントへ返答する．そして，クライアントの管理デーモンは取得した物理アドレス情報を RNBD へ登録する．
- **RNBD** … RNBD (Remote Network Block Device) はカーネルの仮想メモリ (VM) サブシステムから発行されたスワップ I/O アクセスを処理するブロックデバイスドライバである．RNBD は I/O アクセスされたアドレスをサーバの遠隔スワップメモリ領域内の物理メモリアドレスへと変換し，NIC ドライバへ要求を送出する．
- **NIC と NIC ドライバ** … Nuzura システムで使用する NIC と，NIC を制御するデバイスドライバである．RNBD から渡されたアクセス要求は NIC ドライバから NIC ハードウェアへ出力され，サーバへ送信される．

4.2 10GbE NIC UZURA の概要

遠隔スワップメモリシステムでは高性能なインタコネクトを必要とする．我々はそのインタコネクトとして 10GbE NIC UZURA [7] を使用する．UZURA は FPGA 搭載の 10GbE 実験用 NIC であり，NIC 機能とともに RDMA 通信機能をサポートする．Nuzura システムにおけるサーバ・クライアント間の

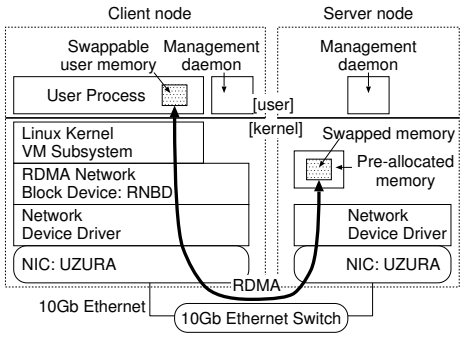

図 2　遠隔スワップメモリシステム: Nuzura
Fig. 2　Remote swap memory system: Nuzura

データ転送は全て RDMA によって通信する．TCP/IP ではなく RDMA を用いた理由は，RDMA のゼロコピー通信によってレイテンシと CPU 使用率を小さくしながらハードウェアスループットを最大限に出すためである．また，Linux の TCP/IP はパケットの送受信時にメモリ確保が行われることがあり，スワップのようなメモリ枯渇条件では安定動作が期待できない．

UZURA の RDMA ではスキャッタギャザリストを用いている．RDMA 転送はクライアント・サーバの各物理メモリアドレスをペアにしたリストを UZURA へ与えることで行われる．このため InfiniBand とは異なり，ソフトウェア側に文献 [5] のようなメモリプールが必要ないという特徴がある．

4.3　RNBD と NIC ドライバ

遠隔メモリをスワップ用のブロックデバイスとして見せるため，我々は RNBD ドライバを実装した．これは Linux カーネルに含まれている NBD (Network Block Device) [6] を元に TCP/IP 通信部分を RDMA 通信を行うインタフェースへ置換え，アドレス変換処理を追加したものである．

また，RNBD のようなカーネル内から RDMA 通信を行うインタフェースを NIC ドライバ内へ新たに用意した．

4.4　Nuzura システムにおけるスワップ処理動作の概要

Nuzura システムにおいて，クライアント上の OS でスワップイン・アウト処理が発生すると，次のような動作が行われる．

- **スワップアウト** … クライアント物理メモリが不足してくると，OS の VM サブシステムはスワップアウト処理を行い，使用頻度の低いページを RNBD に対して書込む．その要求は RNBD から NIC ドライバ・NIC を経由し，サーバメモリへゼロコピーにて送出される．送出が完了したページは未使用領域として再利用される．
- **スワップイン** …クライアント上でスワップイン処理が発生すると，OS の VM サブシステムはメモリ内に空きページを用意し，読込み要求を RNBD に対して発行する．要求が RNBD から NIC ドライバ・NIC を経由してサーバへ到達すると，サーバは該当ページをクライアント物理メモリへゼロコピーにて送出する．

5.　性 能 評 価

5.1　性能評価環境

本論文で性能評価に用いた計算機の仕様を表 1 に示す．この計算機をサーバ・クライアントとして 2 台用意し，10GbE スイッチ富士通 XG700 を経由して接続した．また，サーバの遠隔メモリとして 800MB を確保した．

表 1　評価用計算機の仕様
Table 1　Specification of PC for evaluation

CPU	AMD Opteron 248 (2.2GHz)
メモリ	1GB (PC-3200 512MBx2，メモリバンド幅 6.4GB/s)
ディスク	Maxtor DiamondMax D540X (80GB, Ultra ATA/100)
I/O バス	PCI-X 133MHz バス (UZURA を接続)
OS	Fedora Core 3 (Linux カーネル 2.6.17)

5.2　基本性能の評価

Nuzura システムの基本性能を計測した．まず，NIC のハードウェア通信性能を知るために NIC ドライバへ直接 RDMA 通信を発行したときの性能 “RDMA” を採取した．また，ディスクベンチマークツール iozone の Direct I/O 性能を “RNBD” と “ハードディスク” に対して採取した．各 I/O はランダムアクセスとした．

結果を図 3 に示す．横軸にアクセス時の I/O サイズを，縦軸にスループット性能をとる．RDMA 性能は 155〜872MB/s, RNBD 性能は 122〜740MB/s，ハードディスク性能は 0.7〜27MB/s であった．

I/O サイズ 4KB 時の Read 性能は 1 ページのみスワップインする際の I/O 性能に相当する．I/O サイズ 4KB 時の RNBD 性能 122MB/s は，I/O サイズ 4MB 時に出る 740MB/s の 1/6 でしかない．これは I/O 発行にかかるソフトウェアとハードウェアレイテンシによるものである．従って，スワップ時にアクセスする I/O サイズは，まとめて大きく発行させた方が性能上有利である．

I/O サイズ 4KB 時のレイテンシを計算すると，RDMA では 25.2μs，RNBD では 32.0μs となる．従って，RNBD レイヤを追加したことによるソフトウェアオーバヘッドは 6.8μs であった．また，同様にハードディスクに対してレイテンシを計算すると I/O サイズ 4KB 時に 5.6ms かかっており，RNBD と比較して 174 倍遅い．このためランダムアクセスが多発する場合，ハードディスクベースの遠隔スワップメモリでは性能を出すことが難しい．

RNBD はそのオーバヘッドにより RDMA と比較して I/O サイズ 128KB まで 10〜20%性能が低い．しかし，I/O サイズ 128KB 以降では RNBD と RDMA の性能差が開いている．これは RNBD レイヤから発行する I/O サイズの上限として Linux カーネルのデフォルト値 128KB を用いたためである．この値を 2MB に変更することで最大 827MB/s までスループット性能を引き上げられるが，以降の実験で安定した性能が得られなかった．このため本論文ではこの値を 128KB に据え置くとともに，他のパラメータも OS デフォルト値のままとする．

5.3　スワップの基本性能

Nuzura システムにおける遠隔スワップメモリの性能を評価した．スワップ処理はカーネルが自動的に行うため性能を直接計測することは難しい．そこで，クライアント物理メモリを

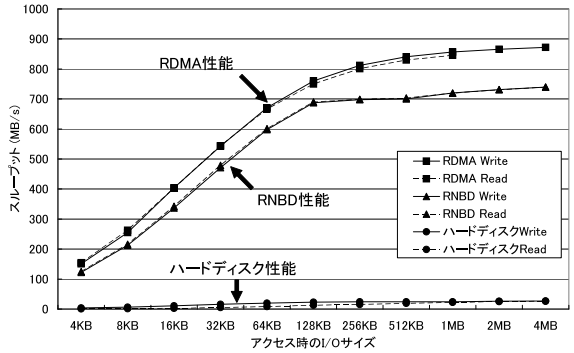

図 3　デバイスの基本性能

Fig. 3　Basic performance of underlying devices

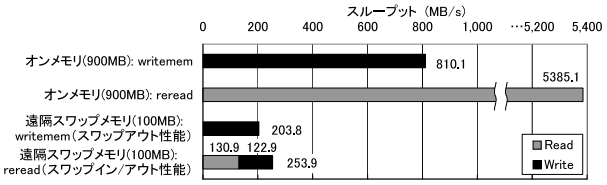

図 4　メモリ・遠隔スワップメモリの性能

Fig. 4　Performance of memory and remote swap memory

900MB（オンメモリと呼ぶ）と 100MB （非オンメモリと呼ぶ）に設定し，700MB の仮想空間に対して以下の処理を順に行うことで，非オンメモリ時に発生するスワップ性能とオンメモリ性能とを比較した．

（1） **writemem** … malloc した直後のメモリ領域 700MB に対し，キャッシュライン毎 (Opteron では 64 バイト) に 4 バイトずつ，値を順に書込んでいく際のアクセス速度とする．オンメモリ時はページフォールトを含めたメモリ Write 性能，非オンメモリ時は主にスワップアウトにかかる性能が求められる．

（2） **reread** … 前記 writemem 処理を行った後，再度領域の先頭から writemem と同じアクセスパターンで値を読込む際のアクセス速度とする．オンメモリ時はメモリ Read 性能，非オンメモリ時はスワップインにかかる性能が求められる．ただし，この条件ではスワップインするページを確保するためにスワップアウトも同等量発生していたことに注意する．

結果を図 4 に示す．遠隔スワップメモリの writemem 性能 204MB/s は，メモリの writemem 性能 810MB/s の 1/4 である．これは遠隔スワップメモリの I/O とともに，スワップアウト処理のオーバヘッドが加わったためである．なお，オンメモリの writemem 性能がオンメモリの reread 性能よりも低い理由は，Linux では malloc した直後のメモリ領域は物理ページが割当てられていないため，物理ページを新たに割当てる Copy On Write 処理のオーバヘッドが入ったためである．

オンメモリ reread 性能 5385MB/s と非オンメモリ reread 性能 131MB/s は 41 倍の差がある．これはオンメモリの Read 性能がメモリバンド幅まで出るのに対し，非オンメモリではスワップイン・アウト両方のコストがかかっているためである．従って，Nuzura の遠隔スワップメモリは reread （スワップイン）性能のペナルティが大きいと言える．

表 2　評価に用いた NAS Parallel Benchmark とその設定

Table 2　Evaluated NAS Parallel Benchmark and its configuration

ベンチマーク名	bt.C	lu-hp.C	lu.C	mg.B	sp.C	ua.C
使用メモリ量 (MB)	707	628	592	443	740	478
ループ回数削減度	1/40	1/25	1/25	-	1/40	1/20

5.4　アプリケーションの適用と評価

5.4.1　性能評価アプリケーション

Nuzura システムを評価するにあたり，以下に示す 4 種類のアプリケーションを使用した．

- **qsort** … 代表的なソートアルゴリズム．メモリアクセス量に対し，計算時間が大きいという特徴がある．ソート要素数は 150M 個 (600MB) とし，GNU C Library 2.3.3 の qsort (_quicksort) 関数にかかる時間を性能値とした．
- **NAS Parallel Benchmark** [8] … 代表的な HPC ベンチマーク集．以後 NPB と略す．バージョン 3.2-SER を使用．NPB に含まれる 11 種類のベンチマークのうち，表 2 に示す 6 種類を用いる．これらはディスク I/O を行わず，入力ファイルで実行時間が変更できることを基準に選択した．なお，mg 以外は測定時間が長いためループ回数を削減する変更を行った．起動から終了までにかかる時間を性能値とした．
- **姫野ベンチマーク** [4] … メモリ読込み負荷が高い HPC ベンチマーク．行列サイズは $(i, j, k) = (177, 177, 353)$ とする (591MB メモリを消費)．C 版を使用．主計算ループ 1 回毎にメモリのほぼ全体を必ず 1 度はアクセスする．主計算ルーチンの MFLOPS 値を性能値とした．
- **Gaussian** [2] … 分子軌道計算を行う商用実アプリケーション．ターゲットデータとしてメモリ量を 449MB 消費するよう変更を加えた test087 を採用した．起動から終了までにかかる時間を性能値とした．

5.4.2　アプリケーション性能測定結果

前節で述べたアプリケーションを Nuzura システム上で性能評価した．クライアント物理メモリサイズを，全てのアプリケーションがメモリに載るオンメモリ (900MB)，490MB，360MB，230MB，100MB と減らしていくことで，アプリケーションが使用する遠隔スワップメモリ量を変化させた．

結果を図 5 に示す．横軸に性能採取時のクライアント物理メモリサイズを，縦軸にオンメモリを 1 とする相対実行時間をとる．クライアント物理メモリサイズを減らすにつれて，各アプリケーションの要求メモリサイズは変らないので遠隔スワップメモリへのスワップ I/O 量が増え，相対実行時間も増えていく．

図 5 を元に，横軸にメモリ拡大率 =（アプリケーションが動作に必要なメモリサイズ÷ローカルメモリサイズ）を，縦軸にオンメモリを 1 とする相対実行時間をとったグラフを図 6 に示す．メモリ拡大率は，例えば 1 の場合にアプリケーションは全てクライアント物理メモリ上で動作するが，3 になった場合は遠隔スワップメモリに 2 ＋クライアント物理メモリに 1 の割合でアプリケーションのメモリが格納されている状態を表す．

図 6 において，qsort の相対実行時間は他と比べて緩かに推移した．qsort で得られた性能値の一部を表 3 に示す．また，ハードディスクをスワップデバイスとして使用した場合の性能も示

す．メモリ拡大率が 1.67 の時に，遠隔スワップメモリの相対実行時間は 1.16 倍だが，ディスクスワップは 41.8 倍になった．

qsort の性能劣化が少ない理由はその計算の特殊性にある．qsort は分割統治法であるため，ソートのワーキングセットは計算が進むにつれて小さくなっていく．やがて，ある時点からソート対象となる全ての要素がクライアント物理メモリに載った状態となる．この様子を表したのが図 7 上である．図 7 上はクライアント物理メモリサイズ 230MB の環境で qsort 計測中に発行されたスワップ I/O 量を示す．全要素 600MB のうち，分割統治されるワーキングセットのサイズが 230MB 以下になるとクライアント物理メモリ上でのみ計算が行われる．そのため，スワップへの I/O 回数が数秒に 1 回しか発生せずオンメモリとの性能差は 1.38 倍となった．

これに対し，図 6 において qsort 以外のアプリケーションではメモリ拡大率が増えると相対実行時間は数倍以上に増えた．これらはいずれも遠隔スワップメモリへの I/O 量が増えたことによる．例としてクライアント物理メモリサイズが 230MB と 100MB 時の Gaussian スワップ I/O 量を図 7 中・下に示す．クライアント物理メモリサイズが 230MB から 100MB に減ると，遠隔スワップメモリへの I/O 量は合計 19.4GB から 89.2GB へ 4.6 倍に増え，計算時間も 173 秒から 725 秒へ 4.2 倍長くなった．図 6 の Gaussian ではメモリ拡大率が 3 から 4 になるときに相対実行時間低下の傾きが大きくなっている．これは Gaussian で主要な計算時間を占めるワーキングセットのサイズが全体の 1/3〜1/4 であると考えられ，遠隔スワップメモリを使用するコストがそれまでに比べて高いことを示す．

図 6 の lu-hp では，メモリ拡大率が増えるに従って加速度的に性能が低下している．lu-hp は主計算ループにおいて行列内をとびとびに細かくアクセスする性質を持つアプリケーションであり，メモリ拡大率 6.28 の時に 5.2TB の総スワップ I/O 量が発生し，相対実行時間は 244 倍を示した．

図 6 において，姫野ベンチマークはメモリ拡大率が 1 から 2 にかけて最も傾きが大きいアプリケーションである．姫野ベンチマークのメモリの多くは，各計算ループ 1 回毎に 1 度しか利用されないという性質を持つ．クライアント物理メモリが不足する状況では，計算に必要なページをスワップインすると，その分しばらく参照されていないページはスワップアウトされる．しかし，次の計算ループが始まると先ほどスワップインしたページは再びスワップアウトされることを繰り返す．従って，クライアント物理メモリがアプリケーションメモリサイズよりも小さくなると，スワップ I/O 量はアプリケーションメモリサイズ分読み書きされ，相対実行時間は変化しないことが期待される．しかし，実際にはクライアント物理メモリサイズが 230MB から 100MB に減ると，総スワップ I/O 量は主計算ループ 1 回あたり 2.1GB から 2.9GB へ増え，相対実行時間も 1.65 倍に増えていた．

以上から，遠隔スワップメモリの実用性は，メモリ拡大率とアプリケーションのメモリアクセスパターンやワーキングセットのサイズによって変化する．qsort のような遠隔スワップメモリへのアクセスが少ないアプリケーションは，メモリ拡大率を増やしても性能低下しにくい．反対に，Gaussian や姫野ベンチマークのようにメモリ拡大率が増えると遠隔スワップメモリへ発生する I/O 量も qsort と比較して増えるアプリケーションでは，その分相対実行時間も増加する．

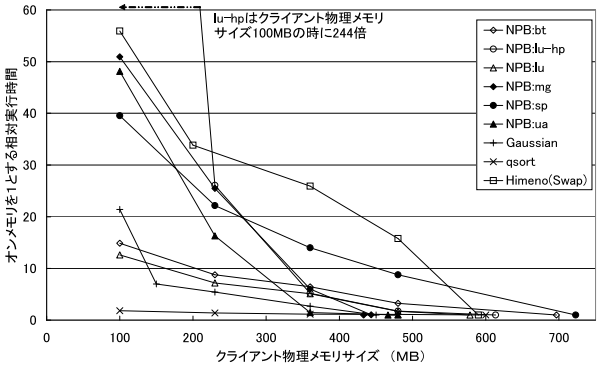

図 5　ベンチマーク性能（横軸: クライアント物理メモリサイズ）
Fig. 5　Benchmark result (x-axis: client physical memory size)

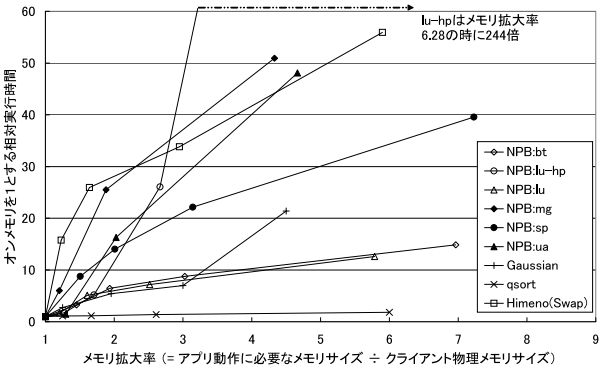

図 6　ベンチマーク性能（横軸: メモリ拡大率）
Fig. 6　Benchmark result (x-axis: memory expansion rate)

表 3　qsort の性能測定結果
Table 3　Result of qsort performance

テスト方法	メモリ拡大率 (比率)	性能 (秒)	オンメモリ性能差 (倍)
オンメモリ	1.00	82.4	1.00
遠隔スワップメモリ	1.67	95.5	1.16
遠隔スワップメモリ	6.00	149.6	1.82
ディスクスワップ	1.67	3446.7	41.82

6.　遠隔スワップメモリ性能の改善検討

前節で，姫野ベンチマークではメモリ拡大率が増えるに従って相対実行時間が増加することを述べた．姫野ベンチマークでは，その主計算ループ中に使用する行列は参照主体と更新主体の 2 種類に分けられる．参照主体の行列はアプリケーションメモリ全体の約 6/7 を占め，初期化時に更新された後，主計算ループ中は参照のみ行われる．残りのメモリ領域を占める更新主体の行列は，主計算中に読み書き両方を行う．そして，どちらも主計算ループ 1 回毎に先頭から一度は全体が参照または更新される．ここで，前節で述べたように主計算ループが 1 回進む毎にスワップインされた行列要素は途中でスワップアウトされている．例えばクライアント物理メモリ 360MB の環境では

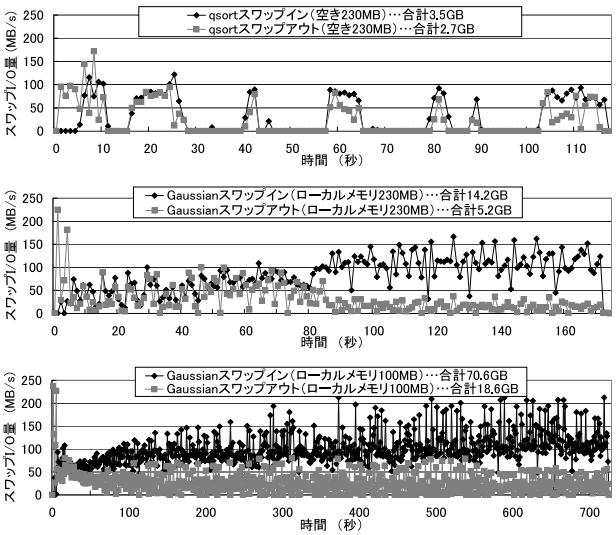

図 7　スワップ I/O 量: qsort(上), Gaussian 230MB(中) 100MB(下)
Fig. 7　Swap I/O volume: qsort (top), Gaussian 230MB (middle) 100MB (bottom)

主計算ループ 1 回あたり 377MB の Write I/O が発生していた．

そこで，アプリケーションのメモリアクセス特性を VM サブシステムが解析できる，またはアプリケーションから通知できると仮定し，ページの使用種別に応じてスワップ I/O を制御することで Linux カーネルのページ置換方式を改良し性能向上する効果が得られないかを検討した．その結果，姫野ベンチマークに以下 2 点の変更を加えることによって性能向上が可能であることが分かった．

- **ファイルマップ** … スワップを用いる代わりにファイルへ mmap したファイルマップ領域に全行列を割当てる．ファイルマップは更新が行われなかったページは書き出さずにメモリから破棄する．このため参照主体の行列を書き戻す I/O 量を削減する効果が期待できる．
- **メモリロック** … 更新主体の行列に対し，メモリロック (mlock) をかける．これにより更新主体の行列に関してスワップアウトやファイルへの書込みを抑えるよう VM サブシステムへ設定することが可能になる．

以上の変更を加えた姫野ベンチマークをクライアントメモリ 360MB の Nuzura システム上で実行した．また，ファイルマップの場合はスワップの代りに RNBD 上へ ext2 ファイルシステムを作成し mmap ファイルを配置した．計測は主計算ループ 1 回分の性能を採取した．姫野ベンチマークは 591MB メモリを使用する設定のため，アプリケーションメモリ全体の 2/5 が常時遠隔スワップメモリ上にある状態となる．

結果を表 4 に示す．表 4 で最良値と平均値の両方を示したのは，姫野ベンチマークでは遠隔スワップメモリを使用した場合，実行毎に I/O によって性能が変動したためである．また，表 4 には主計算ループ中に発生したスワップの Read または Write I/O 量を示す．結果から，スワップ (mlock なし) からファイルマップ (mlock あり) へ変更することで最大 4 倍近く性能が向上した．また，ファイルマップ (mlock あり) では Write I/O 量をほぼ 0 に，Read I/O 量を参照主体の行列サイズに近い 483MB まで抑えることができた．

表 4　ファイルマップとメモリロックを適用した性能値
Table 4　The performance using file map and memory lock

	性能 (MFLOPS)		I/O 量 (MB)	
遠隔メモリ方式	最良値	平均値	Read	Write
(参考:オンメモリ)	953.9	950.3	0	0
スワップ: mlock なし	49.9	44.7	1025.0	377.3
スワップ: mlock あり	70.7	55.8	745.5	381.8
ファイルマップ: mlock なし	143.1	126.1	519.3	61.3
ファイルマップ: mlock あり	194.7	154.0	483.0	0.4

以上から，姫野ベンチマークのメモリアクセスパターンを考慮したページ置換方式を使用することで，スワップ I/O 量を削減して性能を向上させ，遠隔スワップメモリの実用性をさらに高めることが可能であることを示した．

7. おわりに

本論文では遠隔スワップメモリシステム Nuzura の実装について述べた．そして，クライアント物理メモリの最大 7 倍近いメモリ拡大率まで 4 種類のアプリケーションを実行し，相対実行時間が 1.1〜244 倍になることを示した．また，アプリケーションのメモリアクセスパターンやクライアント物理メモリの比率に応じて性能の変化が異なることを示した．この結果から，我々はアプリケーションと実行環境次第で高速インタコネクトを使った遠隔スワップメモリの利用価値はあると考えている．

また，ページ置換方式を変更した姫野ベンチマーク性能を測定した．結果から，アプリケーションのメモリアクセスパターンを考慮することでスワップ I/O 量を削減し，性能向上が可能であることを示した．今後はページ置換方式の改善により，Nuzura システムの性能をさらに高めることを考えている．

文　献

[1] Michail Flouris and Evangelos P. Markatos. “The Network RamDisk: using remote memory on heterogeneous NOWs.” Cluster Computing: The Journal on Networks, Software, and Applications, 2(4), pp. 281-293, 1999.
[2] M. J. Frisch, et al. “Gaussian” http://www.gaussian.com/.
[3] John L. Hennessy and David A. Patterson. “Computer Architecture A Quantitative Approach, 3rd Edition” Morgan Kaufmann Publishers, 2003.
[4] 姫野龍太郎.「姫野ベンチマーク」 http://accc.riken.jp/HPC/HimenoBMT/.
[5] Shuang Liang, Ranjit Noronha, Dhabaleswar K. Panda. “Swapping to Remote Memory over InfiniBand: An Approach using a High Performance Network Block Device” IEEE International Conference on Cluster Computing (Cluster 2005), 2005.
[6] Pavel Machek, et al. “Network Block Device” http://nbd.sourceforge.net/.
[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 10Gb Ethernet インタフェースカードの設計」先進的計算基盤システムシンポジウム SACSIS2006, pp.127-134, 2006.
[8] “NAS Parallel Benchmarks” http://www.nas.nasa.gov/Resources/Software/npb.html .
[9] Tia Newhall, Sean Finney, Kuzman Ganchev, Michael Spiegel. “Nswap: A Network Swapping Module for Linux Clusters” Proceedings of Euro-Par’03 International Conference on Parallel and Distributed Computing, 2003.