InfiniBand クラスタを用いた疎行列線形代数演算ライブラリの効率的な実装技術

西田 晃[†,††]

大規模疎行列に関する線形代数演算を効率的に処理するためには，高性能な相互結合網を備えた並列計算環境を低いコストで構築する必要がある．本研究では，大規模疎行列を対象とした反復解法のスケーラブルな並列実装を実現するため，PCI Express と InfiniBand を組み合わせたクラスタ環境を構築し，性能を評価するとともに，コモディティインターコネクト技術の可能性と実装上の問題点について検討した．評価結果から，疎行列線形演算のスケーラブルな実装を実現するためには，主にノード間の通信帯域幅の確保に重点を置いた設計が必要となることが明らかとなった．

Efficient Implementation of Sparse Linear Algebra Operations on InfiniBand Cluster

Akira Nishida[†,††]

Construction of scalable and low cost parallel computing environment is indispensable for the efficient solution of linear systems with large sparse matrices. In this study, we build a PC cluster based on PCI Express and InfiniBand technology for the scalable implementation of parallel iterative linear solvers, and evaluate its performance. The potential ability and the problems to be solved for commodity interconnect technologies are also discussed. The results of our evaluation show that the communication bandwidth of the intranode interconnects is the most critical factor for the scalable implementation of parallel sparse matrix computations.

1. はじめに

コモディティプロセッサ技術の進展に伴い，高速の PC をネットワークで結合したクラスタ技術が，大規模科学技術計算において実用的な選択肢のひとつとなっている[3),7)]．しかしながら，高速な専用ネットワークを使用した共有メモリ型並列計算機等と比較して，PC クラスタ上でのノード間通信では，通信帯域幅やレイテンシに関するハードウェア上の制約から，十分なスケーラビリティが得られない場合も多い．

大規模疎行列を扱う反復解法において，間接参照を伴うベクトル間演算は計算量の大部分を占める重要な処理である．図 1 に例として共役勾配法のアルゴリズムを示す．疎行列反復解法においては，大半の処理が図 2に示すような内積を含むベクトル間演算，疎行列 - ベクトル間演算から構成されている．したがって，並列化に際しては，これらのベクトル演算が効率的に実装されなくてはならないが，このような処理にはメモリ帯域幅と共に，大域的な通信を処理するための高性能な相互結合網が必要となる．本研究では，これらの問題について詳細に調べるため，コモディティ技術を用いて構成した高性能な PC クラスタ環境を構築し，大規模疎行列を対象とする反復解法を実装する上で障害となるボトルネックについて考察した．

1. Choose x_0
2. $p_0 = r_0 = b - Ax$
   $k = 0$
3. $\alpha_k = (r_k, p_k)/(p_k, Ap_k)$
4. $x_{k+1} = x_k + \alpha_k p_k$
5. $r_{k+1} = r_k - \alpha_k Ap_k$
6. $\beta_k = (r_{k+1}, r_{k+1})/(r_k, r_k)$
7. $p_{k+1} = r_{k+1} + \beta_k p_k$

Increment k and if not convergent goto 3

図 1 共役勾配法のアルゴリズム
Fig. 1 Algorithm of Conjugate Gradient Method.

2. 背景

ワークステーションやサーバを高速ネットワークで相互に接続するための試みとして，様々な技術が提案されている．学術用途には Myrinet や Quadrics，ギガビットイーサネット等が多く用いられてきたが，近年，Intel，富士通等により提案された InfiniBand 技術の開発が進展している．また，PCI バスと互換性を持つ次世代イン

† 東京大学大学院情報理工学系研究科コンピュータ科学専攻
Department of Computer Science, the University of Tokyo
†† 科学技術振興事業団 CREST
CREST, JST

図 2　反復解法における演算処理の例.
Fig. 2　Vector operations for iterative solvers.

タフェース規格として, PCI Express が昨年より実用化され, 広い帯域幅を必要とする高速なネットワーク環境の構築が容易になってきている. 本研究では, これらの技術を組み合わせることにより, 大規模疎行列演算を効率的に処理するためのスケーラブルなクラスタ環境を構築するとともに, その評価を行った. 以下ではこれらの技術的背景について述べる.

PCI Express

PCI Express は従来の PCI バス技術と互換性を持つ次世代のシリアル転送インタフェース規格であり, Intel, NEC 等により 2004 年より実用化されている. PCI バスが 1GB/s の帯域幅を上限とする共有バス方式であったのに対して, PCI Express ではデバイス間を直接接続することができ, また一方向 2.5Gb/s の帯域幅を持つレーンを最大 32 本まで束ねることことにより, 双方向で最大 16GB/s の実効帯域幅を実現する (図 3 参照). グラフィックカード用のバス規格である AGP (Accelerated Graphic Port) と統合されたことから, 安価なマザーボードを用いて高性能な相互結合網を持つクラスタ環境を構築することが可能となっている.

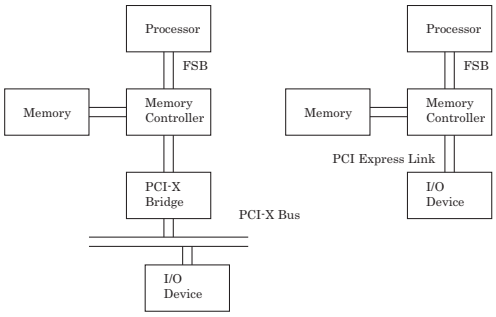

図 3　PCI-X, PCI Express の比較
Fig. 3　Comparing PCI-X with PCI Express.

InfiniBand

PCI Express に現時点で対応している高速ネットワーク技術として, InfiniBand を挙げることができる. Mellanox Technologies 社の開発する InfiniHost ホストチャネルアダプタ (HCA) は, 1 レーン当たり 2.5Gbps の帯域幅を持つ InfiniBand を 4 本束ねた 2 個のポートを持ち (図 4 参照), 8 レーンの PCI Express スロットを利用することにより, 双方向で 40Gbps の帯域幅を実現している ☆.

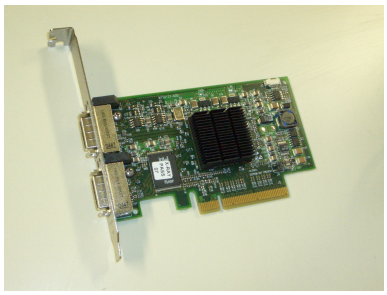

図 4　Mellanox Technologies 社の PCI Express 対応 InfiniHost HCA MHEA28-XT.
Fig. 4　InfiniHCA HCA MHEA28-XT for PCI Express bus from Mellanox Technologies, Inc.

なお, InfiniBand の帯域幅は今後さらに拡張される予定であり, Mellanox Technologies 社では 1 ポート当たり片方向で 20Gb/s の帯域幅を持つ通信チップを既に発表している.

プロセッサ

Opteron プロセッサは AMD 社の開発するマルチプロセッシング対応の 64bit プロセッサであり, チップ内にメモリコントローラを内蔵している. メモリレイテンシが小さく, メモリ帯域幅がプロセッサ数に比例して増大する点に特徴がある.

メモリアフィニティ

Opteron プロセッサは cc-NUMA 技術に対応していることから, 十分なメモリ帯域幅を確保するためには, メモリアフィニティ機能が実装されたオペレーティングシステムを用いることが必要である[6]. 本研究では, 実験に用いたすべてのノードについて, カーネル 2.6.4 を採用した SuSE Linux 9.1 を用いた. 2.6 以降の Linux カーネルはメモリアフィニティを実装しており, 複数の Opteron プロセッサのメモリ帯域幅を引き出すことができる.

以上の技術的背景から, 本研究では相互結合網として Mellanox Technologies 社の PCI Express 対応 dual port InfiniHost HCA を用いてクラスタ環境を構築することとし, 128MB の DDR メモリを搭載した MHEL-CF128-T, サーバの主記憶を通信用に利用して低コスト化を実現した MHEA28-XT の 2 種類の HCA を用いて評価を行った. なおスイッチには 24 ポートの InfiniBand スイッチである MTS2400 を使用した.

まず, 予備評価として, 早期に PCI Express に対応した Intel 社の 64bit Xeon サーバと AMD 社の Athlon64 マザーボードを用いて InfiniBand ネット

☆ ただし冗長性を確保するため 8B/10B データ符号化方式を採用しており, 実際の帯域幅はこの 80% の 4GB/s となっている.

ワークの通信性能を評価した．前者には Dell 社の PowerEdge SC 1420, チップセットには Intel E7520 を, 後者には Asus A8N-SLI Deluxe, チップセットには NVIDIA nForce4 SLI を用いた

これらに MHEL-CF128-T を搭載し, 通信性能を測定した結果, それぞれ 1.8GHz, 2.2 GHz の Athlon64 プロセッサを搭載したサーバ間の通信帯域幅は約 970MB/s, また 2.8HGz Xeon プロセッサ搭載サーバ (1 スレッドのみを使用) の結果は約 800MB/s であった．MPI レイテンシに関してはいずれも 4us 弱であり, Mellanox 社の公称値である 3.7us とほぼ等しい．

これらの結果を踏まえ, 本研究ではクラスタ構築に Opteron プロセッサを使用し, PCI Express に対応した NVIDIA 社のチップセットである nForce Professional 2200 と組み合わせることとした．nForce Professional を使用したマザーボードには, 2005 年 3 月に発表された Rioworks 社の 2-way 構成用のマザーボード HDAM Express を, またプロセッサには 2GHz, 1MB キャッシュ搭載の Opteron 246 を用いて, 8 ノード 16 プロセッサからなるクラスタを自作し, 実験に使用した．なおメモリには 512MB PC3200 DDR SDRAM (ECC Registered) 4 枚を使用した．

図 5 InfiniBand スイッチ MTS2400 で接続された Opteron クラスタ
Fig. 5 Opteron Cluster connected with InfiniBand switch MTS2400.

3. 性能評価

InfiniBand に対応した MPI ライブラリとしては, MVAPICH[4] や LAM MPI[2], MPICH/SCore[8] などを挙げることができる．MVAPICH には通信をモニタして必要に応じて分割し, 複数のポートに分配するストライピング機能が実装されており[5], 実験に用いた PCI Express 対応 InfiniHost HCA には 2 個の 4x ポートが搭載されているため, 以下では MVAPICH の最新版である 0.9.5 を使用した．比較のため, 1 個のポートのみを用いた場合, ch_p4 MPICH 1.2.7 を用いて GbE ネットワーク上で計測した場合についても調べた．なお GbE については 32bit, 66MHz PCI 対応の RTL8169 チップ搭載カードを PCI-X スロット上で使用した．また, 専用ネットワークとの比較のため, SGI 社の分散共有メモリ型並列計算機である Altix 3700 上でも MPI の性能を測定した．Altix では, 独自の高速ネットワークである NUMAflex を用いて Intel Itanium2 プロセッサを接続している．1 ノード内の 2 個のプロセッサが 6.4GB/s の帯域幅で接続され, さらに 4 個のノードが 3.2GB/s の帯域幅でルータに接続される構造となっている．ここでは, 32KB L1 キャッシュ (16KB data), 256KB L2 キャッシュ, 及び 3MB L3 キャッシュを備え, プロセッサ当たり 1GB の主記憶を持つ 1.3GHz Itanium2 プロセッサ 32 個を搭載した Altix 3700 を使用し, 隣接する 16 プロセッサを番号順に割り当てて用いて実験を行った．

図 6-8 に MHEL-CF128-T, MHEA28-XT の MPI レイテンシと帯域幅を示す．GbE 上での MPI レイテンシが最小で 28.71us, 双方向帯域幅は最大で 58MB/s であったのに対し, InfiniHost HCA の MPI レイテンシは 3.99us, 双方向帯域幅は 2907MB/s (MHEL-CF128-T) であった．ストライピング機能はメッセージサイズが小さい場合にはレイテンシに特に影響を与えていないが, サイズが大きくなるとレイテンシを小さくする方向に働く．

なお, ノード内での MPI レイテンシはノード間に比べて低くなっているが, 帯域幅が小さくなっており, 現状では注意が必要である．また, MHEA28-XT の帯域幅の特性は MHEL-CF128-T とほぼ同じであり, 4MB までのサイズでの性能の低下は見られなかったが, 双方向帯域幅に関しては図 8 に示すように, MHEL-CF128-T と比較して, 若干小さなサイズで性能低下が起きている．これらの原因はドライバの実装に問題が残っているためであると思われるが, 実験ではノード 0-3 に MHEL-CF128-T を残すこととし, ノード 4-7 に対して MHEA28-XT を使用した．

PCI-X バスを用いた場合については, レイテンシが約 6us, 双方向帯域幅が 1877MB/s であることが報告されている[5] ので, PCI Express を用いることにより, 大幅に性能が向上していることがわかる．また, MVAPICH の配布元である Ohio 州立大学の測定結果☆によれば, 3.4GHz の EM64T 対応 Xeon サーバ上での帯域幅の最高値は 2704MB/s であり, この結果と比較しても 7% 程度高い性能が得られている．クロックの異なる Athlon64 ノード間での通信性能が Opteron クラスタでの実験結果とほぼ同じであることから, 通信性能はチップセットの性能によってほぼ決定されると推測されるので, 本研究で構築したクラスタは, 現時点で最も高い通信帯域幅を持っているものと考えられる．

比較のため, Altix 3700 の結果を図 9-10に示す．InfiniBand と比較すると, レイテンシに関しては依然良い

☆ http://nowlab.cis.ohio-state.edu/projects/mpi-iba/

性能を示しているものの, 帯域幅は相対的に小さい.

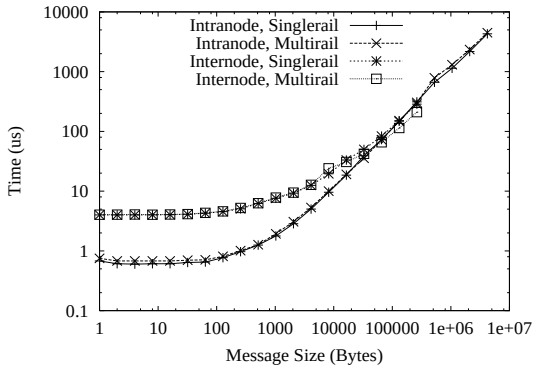

図 6 InfiniHost HCA (MHEL-CF128-T) の MPI レイテンシ
Fig. 6 MPI Latency of InfiniHost HCA (MHEL-CF128-T) with different number of communication ports.

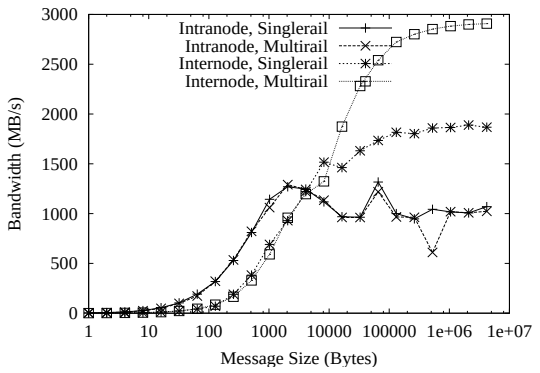

図 7 InfiniHost HCA (MHEL-CF128-T) の双方向帯域幅.
Fig. 7 MPI Bidirectional Bandwidth of InfiniHost HCA (MHEL-CF128-T) with different number of communication ports.

3.1 STREAM benchmark

次に, ベクトル演算で必要となる局所的なメモリ帯域幅について調べるため, STREAM benchmark[1] を用いて, Opteron クラスタと SGI Altix 3700 の双方について, メモリ帯域幅を測定した. このベンチマークプログラムでは, 倍精度浮動小数配列に対して以下の演算を行い, 実測値をもとに計算機の実効帯域幅を評価する.

ここでは MPI 版の並列プログラム `stream_mpi.f` を用い, 複数のネットワーク構成で評価した. 図 11 に STREAM benchmark の評価結果を示す. 問題サイズはプロセス当たり 4,000,000, すなわち約 91.6MB である. 結果を平均すると 1 プロセス当たり約 2.5GB/s のメモリ帯域幅が得られており, SGI Altix 3700 上での OpenMP 版 STREAM Benchmark の実行結果 (図

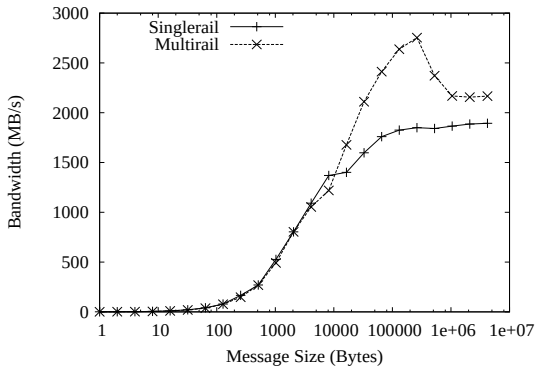

図 8 InfiniHost HCA (MHEA28-XT) の双方向帯域幅.
Fig. 8 MPI Bidirectional Bandwidth of InfiniHost HCA (MHEA28-XT) with different number of communication ports.

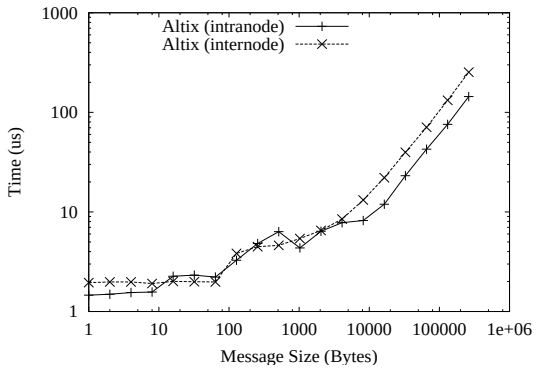

図 9 SGI Altix 3700 上での MPI レイテンシ
Fig. 9 MPI Latency on SGI Altix 3700.

表 1 STREAM benchmark の構成
Table 1 STREAM benchmark types.

Benchmark	Operation	Bytes per iteration
Copy	a[i] = b[i]	16
Scale	a[i] = q * b[i]	16
Add	a[i] = b[i] + c[i]	24
Triad	a[i] = b[i] + q * c[i]	24

12) と比較して, ベクトル演算に関してより十分な性能が確保できているといえる.

4. NAS Parallel Benchmark

以上の準備のもとに, 反復解法とクラスタの通信性能との関係を調べるため, NAS Parallel Benchmark 3.2 のうち, MPI 版 CG の Class S から C までを用いて評価を行った. CG では対称正定値疎行列の最小固有値を逆反復法と共役勾配法により計算する. NAS Parallel Benchmark はクラスタの性能を調べるために用いられ

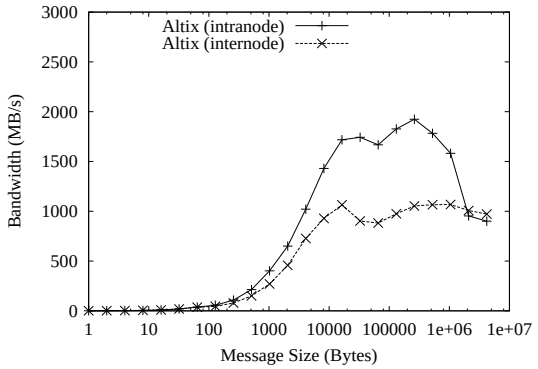

図 10 SGI Altix 3700 上での双方向帯域幅.
Fig. 10 MPI Bidirectional Bandwidth on SGI Altix 3700.

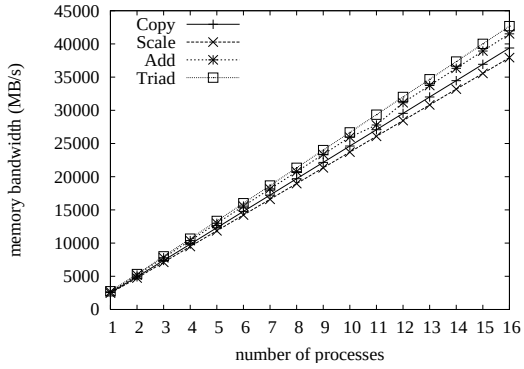

図 11 ノード当たり 2MPI プロセスで実行した場合の STREAM benchmark の台数効果
Fig. 11 Speedup of STREAM benckmark results with two processes per node.

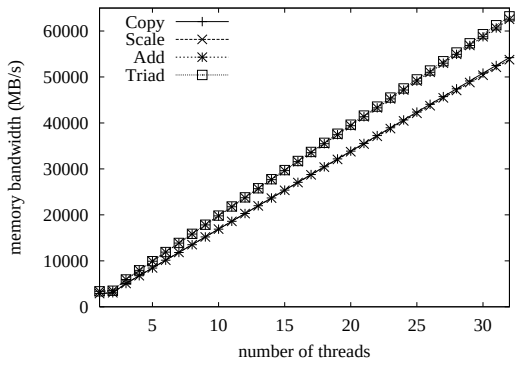

図 12 SGI Altix 上 (配列サイズ 80,000,000) での STREAM benchmark の性能 (MB/s)
Fig. 12 STREAM benchmark performance in MB/s with array size 80,000,000 on SGI Altix with memory affinity

ることが多いが, 図 13 に Altix での例を示すように, CG に関しては通常は十分なスケーラビリティが得られないのが普通である.

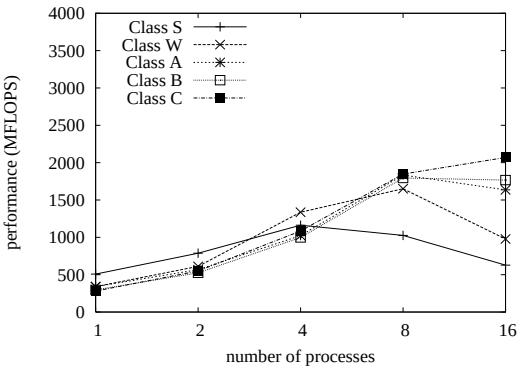

図 13 SGI Altix 3700 上での NPB CG の演算性能
Fig. 13 Speedup of NPB CG on SGI Altix 3700.

これには様々な原因が考えられるが, ここでは帯域幅に関する制約が原因であると仮定し, これに関する評価を行った. GbE ネットワーク, InfiniHost HCA 上の 1 ポートのみを使用した場合, 2 ポートを使用した場合の結果を以下に示す.

GbE 上での結果から, 同一の計算性能を持つクラスタ環境においても, 帯域幅が性能に大きな影響を与えていること, また, InfiniHost HCA 上での結果から, 同一のレイテンシのネットワークにおいても, 帯域幅が小さい場合にはスケーラビリティに限界があり, 性能低下が見られることが分かる. このことは, CG においては (レイテンシよりも) 帯域幅が性能を制約していること, また, ポート数を増やして帯域幅を向上させることによりスケーラビリティが得られていることからも, (CG の並列化効率自体には問題がなく, むしろ) ネットワーク性能によって演算性能が律速されることが分かる.

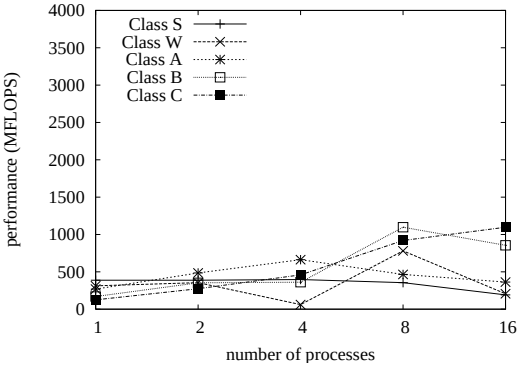

図 14 GbE 1 ポートを使用した場合の NPB CG の演算性能
Fig. 14 Speedup of NPB CG with one port of GbE.

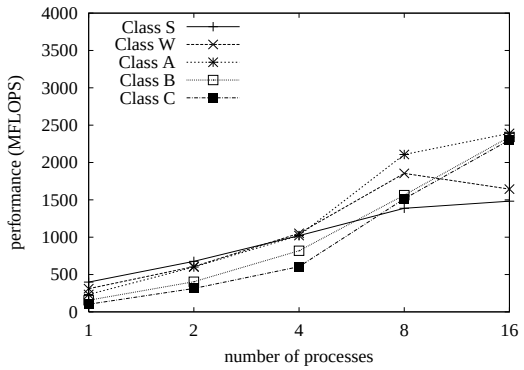

図 15 InfiniHost HCA 上の 1 ポートのみを使用した場合の NPB CG の演算性能
Fig. 15 Speedup of NPB CG with one port of InfiniHost HCA.

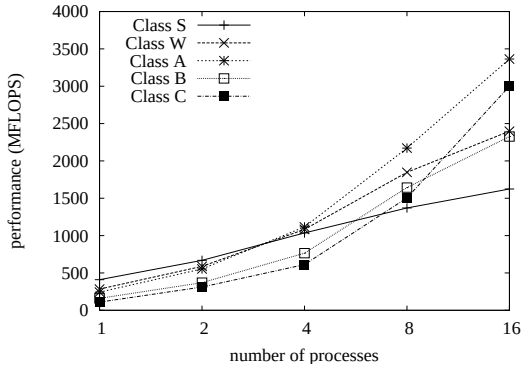

図 16 InfiniHost HCA 上の 2 ポートを使用した場合の NPB CG の演算性能
Fig. 16 Speedup of NPB CG with two ports of InfiniHost HCA.

5. 考　　察

ここまで見てきたように, アプリケーションの特性には違いがあり, 特に今回取り上げた疎行列線形演算においては, ノード間の通信帯域幅によって性能が律速されていることから, 実装に際しては適切にアーキテクチャを選択する必要があることが分かる. 特に反復解法を多用する機会の多い流体力学等の科学技術計算分野においては, 相互結合網の選択は重要な意味を持っており, 今回の評価では比較的顕著に性能の違いが現われている点で, 興味深い結果であるといえる.

6. ま と め

本稿では, コモディティハードウェアによる計算環境として PCI Express 対応の InfiniBand 技術を用いたクラスタ環境を構築し, 比較対象として SGI Altix 3700 を用いてその性能を評価するとともに, コモディティインターコネクト技術の可能性と実装上の問題点について検討した. その結果, 疎行列線形演算のスケーラブルな実装を実現するためには, 通常のアプリケーションとは異なった観点, すなわち, ノード間の通信帯域幅の確保に重点を置いたアーキテクチャを構築する必要があることを示した.

謝辞　本研究を進めるに当たり, ご議論を頂いた小柳義夫教授, 小柳研究室諸氏に感謝の意を表します. また, InfiniBand の性能評価にあたっては, Mellanox Technologies, Inc., 株式会社アルティマより多くのサポートを頂きました. なお, 本研究の一部は, 科学研究費補助金特定領域研究 16016225, 及び科学技術振興事業団戦略的創造研究推進事業の支援によるものである.

参 考 文 献

1) *STREAM: Sustainable Memory Bandwidth in High Performance Computers*, http://www.cs.virginia.edu/stream/.
2) G. Burns, R. Daoud, and J. Vaigl, *LAM: An Open Cluster Environment for MPI*, in Proceedings of Supercomputing Symposium, 1994, pp. 379–386.
3) J. I. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach, Third Edition*, Morgan Kaufmann, 2003.
4) J. Liu, B. Chandrasekaran, W. J. J. Wu, S. P. Kini, W. Yu, D. Buntinas, P. Wyckoff, and D. K. Panda, *Performance Comparison of MPI implementations over Infiniband, Myrinet and Quadrics*, in Proceedings of the International Conference on Supercomputing '03, 2003.
5) J.Liu, A.Vishnu, and D.K. Panda, *Building Multirail InfiniBand Clusters: MPI-Level Design and Performance Evaluation*, in Proceedings of the International Conference on Supercomputing '04, 2004.
6) A. Nishida and Y. Oyanagi, *Performance Evaluation of Low Level Multithreaded BLAS Kernels on Intel Processor based cc-NUMA Systems*, LNCS, 2858 (2003), pp. 500–510.
7) G. F. Pfister, *In Search of Clusters*, Prentice-Hall, second ed., 1998.
8) S. Sumimoto, A. Naruse, K. Kumon, K. Hosoe, and T. Shimizu, *PM/InfiniBand-FJ: a high performance communication facility using InfiniBand for large scale PC clusters.*