HPC チャレンジでの SX システムの性能評価

小林 広明[1)] , 滝沢 寛之[1)2)], 小久保 達信[3)], 岡部 公起[1)],
伊藤 英一[1)] , 小林 義昭[3)], 浅見 暁[4)], 小林 一夫[4)]
後藤 記一[4)], 片海 健亮[5)], 深田 大輔[5)]

1) 東北大学情報シナジーセンター, 2) 東北大学大学院情報科学研究科,
3) 日本電気株式会社, 4) NEC情報システムズ, 5) NECソフト

1. はじめに

HPC (High Performance Computing) チャレンジは、総合的な HPC システム性能評価の試みとして、米国 DARPA (Defense Advanced Research Projects Agency) の HPCS (High-Productivity Computing Systems) プロジェクトの支援を受けて[1]、Tennessee 大学の J. Dongarra 博士 により、2003 年 11 月 SC2003(2003 年 Supercomputing Conference)において提唱された HPC システムのベンチマーク(BM)セットです[2]。Linpack HPC の単一性能指標のみで評価する Top500 を補完するものとして、より総合的で Linpack(HPL)も含んだ 7 セットの BM コードの集まりとなっています。この BM セットでは、これまでスーパーコンピュータの性能評価で重要視されてきた総演算性能の評価に加えて、アプリケーション実行におけるスーパーコンピュータの実効性能を引き出す上で重要なメモリアクセス性能とネットワーク性能の評価と、多くのアプリケーションで頻繁に使用されるカーネルコードを用いた性能評価が可能になっています。これにより、従来のノード数に頼るスーパーコンピュータの数量的な評価ばかりでなく、ノードの「質」を評価することが可能になります。本報告では、NEC と東北大学情報シナジーセンターが共同で行った HPC チャレンジベンチマークを使った SX-7 の評価結果について述べ、28 評価項目の中、16 項目で最高性能を出したベクトル型スーパーコンピュータの HPC 分野における優位性を明らかにします。

2. HPC チャレンジとは

テストの概要は、HPC チャレンジの Web ページを参照するのが一番です。

http://icl.cs.utk.edu/hpcc/index.html

この Web ページの情報をもとに説明します。HPC チャレンジは 7 つのカテゴリーの BM セットとなっています。まずは、評価方法について説明し、次にそれぞれのテストついて説明します。

2.1 評価方法

評価方法は、全ノード総合性能を評価するテスト(G: Global system performance)と、ノード単体のシングル環境でのテスト(SN: Single Environment)、多重負荷環境でのテスト(EP: Embarrassingly Parallel)の 3 つがあります。この 3 つのテストは、計算機システムが複数の CPU

から構成される並列計算機となっていることで、その特徴を評価することを目的としています。最近の並列計算機の構成方法は、SMP によるメモリ共有型計算機、MPI による分散メモリ型計算機、そしてその組み合わせによるハイブリッド型並列計算機があります。特に後者の組み合わせの並列計算機では、メモリが共有される単位をノードとして定義することが多く、そのノードをネットワークで繋ぎ一つのシステムとなっています。このように計算機システムが複雑な構成となった場合、全体と部分の評価が必要で、上記 3 つのテストで対応します。

全ノード総合性能テストでは、計算機システム全体を使ってのテスト(G)となり、一つのシステム全体でどの程度性能があるかを評価するものとなります。更に全ノード総合性能だけではなく、各ノードの性能についても評価するものが次の 2 つです。すなわち、ノード内ではメモリは共有されているため、複数プロセスを同時に使ってテストをするとメモリ利用の奪い合いが起こります。これを避ける一番簡単な評価方法が、プロセスを 1 つとしてテストすることで、これがシングル環境でのテスト(SN)になります。この場合、単体ノードの性能が最大限に発揮されます。逆に、複数のプロセスを同時に使ったテストの場合、メモリ利用の奪い合いが起こり、シングル環境のテストよりも性能が劣化することになります。その性能劣化具合を調べるのが多重負荷時のテスト(EP)になります。これらのテストで、計算機システムの各ノードの性能評価を行えます。

2．2 HPCC ベンチマークプログラム

2．2．1 HPL：連立1次方程式（LU 分解と後退代入）の計算プログラム

いわゆる Linpack の MPI で記述された分散並列版で、ノード全体の演算性能を評価します。オリジナルコードは、Netlib の HPL です。全ノードのトータルな演算性能が大きいほど高い性能となり、全ノード総合性能に依存します。性能の単位は Tflop/s（Tera floating-point operations per second:一秒間に浮動小数点演算を何回行うかを、10 の 12 乗の単位で表す）で表示されます。テスト項目は 1 項目で、全ノードを使った総合性能(G)のテストを行います。

2．2．2 DGEMM：実数行列 A,B の行列積 C=AB を計算するカーネルプログラム

DGEMM は Netlib の BLAS(Basic Linear Algebra Subprograms)の一つの機能として提供され、さまざまな数値計算に出てきます。例えば連立 1 次方程式 LU 分解計算（Linpack）では、演算の主要部分が DGEMM となり、性能を左右する一番重要な要素となります。本 BM による評価結果は、全ノード総合性能には依存せず、ノード単体の演算性能に依存します。性能の単位は Gflop/s（Giga floating point operations per second:一秒間に浮動小数点演算を何回行うかを、10 の 9 乗の単位で表す）で表示されます。テストの項目は 2 項目で、ノード単体のシングル環境(SN)、多重負荷環境(EP)のテストを行います。

2．2．3 STREAM：メモリバンド幅の評価プログラム

複写(Copy)、定数倍(Scale)、総和(Add)、積和(Triad)の 4 つの計算プログラムからなっており、ノード単体のメモリ性能を測定します。オリジナルコードは、J. D. McCalpin 博士の STREAM memory bandwidth benchmark です。全ノード総合性能には依存せず、ノード単体のメモリ性能に依存します。性能の単位は GB/s（Giga Bytes per second:一秒間に転送するメモリサイズ（バイト）を 10 の 9 乗の単位で表す）で表示されます。テストの項目は 8 項目で、シングル環境(SN)および多重負荷環境(EP)での、各ノード単体の 4 項目（複写、定数倍、総和、積和）のテストを行います。

2．2．4 PTRANS：行列の転置 $A=A+B^T$ を行うプログラム

PTRANS (Parallel matrix TRANSpose)は、全ネットワーク転送性能を行列の転置で評価します。オリジナルコードは Netlib の PARKBENCH (PARallel Kernels and BENCHmarks)です。全ネットワーク転送性能が大きいほど高い性能となり、全ネットワーク総合転送性能に依存します。性能の単位は GB/s で表示されます。テスト項目は 1 項目で、全ノードを使った総合性能(G)のテストを行います。

2．2．5 RandomAccess：整数データの間接参照（インダイレクトアクセス）性能を評価するプログラム

オリジナルコードは、DARPA HPCS Discrete Math Benchmarks です。ノード単体のメモリ間接参照のテストと、全ノードの MPI 通信での参照テストとなっています。ノード単体の性能に依存する項目と、全ノード総合性能に依存する項目の両方があります。性能の単位は Gup/s（Giga updates per second:一秒間に更新する要素数を 10 の 9 乗の単位で表す）で表示されます。テストの項目は 3 項目で、ノード単体のシングル環境(SN)、多重負荷環境(EP)のテスト、および全ノードを使った総合性能(G)のテストを行います。

2．2．6 FFTE ：離散フーリエ変換の性能評価を行うカーネルプログラム

一次元離散フーリエ変換を高速フーリエ変換で計算するカーネルプログラムです。オリジナルコードは、筑波大学の高橋大介博士が開発した FFTE です。ノード単体の FFT のテストと、全ノードを使った FFT のテストとなっています。ノード単体の性能に依存する項目と、全ノード総合性能に依存する項目の両方があります。性能の単位は Gflop/s で表示されます。テストの項目は 3 項目で、ノード単体のシングル環境(SN)、多重負荷環境(EP)のテスト、および全ノードを使った全環境(G)テストを行います。

2．2．7 Communication bandwidth and latency：データ転送能力を評価するプログラム

オリジナルコードは、HLRS(ドイツ High Performance Computing Center in Stuttgart)が開発の b_eff (effective bandwidth benchmark)です。

一般に、データ転送時間 T は、データ転送の立ち上がり時間（スタートアップ時間, Start-up_Time）とデータ転送速度(バンド幅 BW: Bandwidth)を用いて次式で与えられます。

T=Start-up_Time + (Data Size)/BW

転送するデータ量(Data Size)が小さい場合は、データ転送時間 T においてスタートアップ時間が支配的になるために、スタートアップ時間が短いほど優れた性能を示します。一方、転送データ量が大きい場合は、データ転送時間におけるセットアップ時間の割合は相対的に小さくなるために、バンド幅が高いほどデータ転送能力が高くなります。したがって、セットアップ時間は小さく、バンド幅は大きいほど、データサイズに関係なくデータ転送能力が優れていることになります。実際の測定では、スタートアップ時間は最小単位のデータの転送時間（レイテンシ:latency）を用いて、バンド幅は MB オーダのデータ転送時間を用いて、それぞれ近似的に評価します。

データ転送能力を評価するため、ネットワークの転送スキームは、Ping-Pong, Ring(Naturally

ordered, Randomly ordered)が用意されています。ネットワーク全体のポイント間の性能を評価するため、ノード間通信が多いほど性能が落ちることもあります。性能の単位はレイテンシがマイクロ秒、バンド幅が GB/s で表示されます。テストの項目は、バンド幅とレイテンシそれぞれ 5 項目ありますが、2004 年 12 月現在、ブラウザから詳細表示される項目をあげるとレイテンシが 2 項目、バンド幅が 3 項目となっています。測定項目は全部で 10 項目ですので、今後表示される項目が変更される可能性があります。

3. 実行ルール

HPC チャレンジには、ベースラインランとオプティマイズランの 2 つ実行ルールがあります。2004 年 12 月現在、全登録された結果は 49 件あるうち、ベースラインランは 45 件、オプティマイズランが 4 件となっており、多くの公表された値は、ベースラインランとなっています。

3.1 コードの変更が許されないベースラインラン

基本実行ルールであるベースラインランは、コンパイラによる高度な最適化と、高性能の BLAS ライブラリ、MPI ライブラリを使うことで HPC チャレンジベンチマークを評価します。HPC チャレンジの評価結果には、使用したコンパイラおよびライブラリのバージョン、そしてコンパイル時のオプションが公開されます。

3.2 コードの最適化が許されるオプティマイズラン

コードの修正を伴う最適化は、2 つのレベルが許可されています。一つは、限られた部分（特定のサブルーチン単位）のコード最適化であり、もう一つは、アルゴリズムの見直しを含んだ根本的な最適化です。しかし、後者は HPCC プロジェクト主催グループとの協議が必要で、また、その結果は公開されることになっています。2004 年 12 月現在のオプティマイズランの中で、前者の限られた部分の最適化の方しか登録されていません。

4. ベンチマークコードの入手方法

ベンチマークコードは HPCC のサイトで公開され、誰でもダウンロード可能です。

http://icl.cs.utk.edu/hpcc/software/index.html

最初のバージョン 0.3α は 2003 年 11 月 5 日に公表されています。この時点では、大テスト項目は 5 項目でした。0.6α が 2004 年 5 月 31 日に更新され、この時点でテスト 2 項目（DGEMM と FFTE）が追加され、2004 年 12 月時点と同じ大テスト項目が 7 項目となりました。最新のバージョンは 0.8β で 2004 年 10 月 19 日に更新されています。0.6α から見て大きな変更は無く、出力の変更などマイナーチェンジとなっています。

5. 各テスト項目の詳細分析とその評価結果

HPC チャレンジは、C 言語で書かれており、make(コンパイルとリンク)すると一つの実行形式（ロードモジュール）ができます。これを実行すると全項目のテストがまとめて行われ、結果が表

示されます。入力データのサイズやパラメータを制御するのが、「hpccinf.txt」という入力ファイルです。この入力ファイルの内容は次のようになっています（分かりやすいように、行番号をつけてあります）。

```
 1     HPLinpack benchmark input file
 2     Innovative Computing Laboratory, University of Tennessee
 3     HPL.out      output file name (if any)
 4     8            device out (6=stdout,7=stderr,file)
 5     1            # of problems sizes (N)
 6     30000        Ns
 7     1            # of NBs
 8     64           NBs
 9     1            PMAP process mapping (0=Row-,1=Column-major)
10     1            # of process grids (P x Q)
11     1            Ps
12     32           Qs
13     16.0         threshold
14     1            # of panel fact
15     2            PFACTs (0=left, 1=Crout, 2=Right)
16     1            # of recursive stopping criterium
17     44           NBMINs (>= 1)
18     1            # of panels in recursion
19     3            NDIVs
20     1            # of recursive panel fact.
21     2            RFACTs (0=left, 1=Crout, 2=Right)
22     1            # of broadcast
23     0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
24     1            # of lookahead depth
25     1            DEPTHs (>=0)
26     2            SWAP (0=bin-exch,1=long,2=mix)
27     64           swapping threshold
28     0            L1 in (0=transposed,1=no-transposed) form
29     0            U  in (0=transposed,1=no-transposed) form
30     1            Equilibration (0=no,1=yes)
31     16           memory alignment in double (> 0)
32     ##### This line (no. 32) is ignored (it serves as a separator). ######
33     0                       Number of additional problem sizes for PTRANS
34     1200 10000 30000                values of N
35     2                       number of additional blocking sizes for PTRANS
36     134   471                       values of NB
```

1〜31 行目は HPL に関連したパラメータとなっています。33〜36 行目が PTRANS に関連したパラメータです。以降、各テスト項目について説明しますが、HPL に関してはパラメータの決め方をより詳しく説明します。

今回、我々はスーパーコンピュータの「質」を評価するという観点から、SX-7 の 1 ノード（32 CPU）を用いて、その性能を 2 つ並列処理形態で評価しました。一つはノード内 32CPU の各 CPU をそれぞれ独立のノードとして見立てて、1CPU に 1MPI プロセスを割り当て、性能評価を行います。この場合、MPI による 32CPU の分散並列処理となります。もう一つは、SX-7 の共有並列を最大限に活かし、32CPU を 2 つの 16CPU グループに分け、それぞれに MPI プロセスを割り当てると共に、1 つの MPI プロセス内では 16CPU による SMP 共有並列処理を行うハイブリ

ッド型の並列処理形態です。HPC チャレンジは、プロセス間の通信などの評価を行うため、最低限 MPI の 2 プロセスのテストが必要となります。

また、評価の途中では、SMP 共有並列と MPI 分散並列の組み合わせの割合を変化させて、さまざまなパターンでの評価も同時に行いました。

5．1 HPL

- SMP 並列処理の導入効果

HPL のオリジナルコードは MPI で分散並列化されたものとなっていますが、SMP 並列化された BLAS とコンパイラによる自動 SMP 並列を使用することで、HPL を SMP 共有並列と MPI 分散並列を組み合わせたハイブリッドの並列化が可能となります。コードの変更が許されないベースラインランでは、一部のコードで SMP 共有並列のオーバヘッドに比べて処理時間が短い部分があり、コンパイラの判断により自動での並列化が行われないため、MPI 分散並列だけの結果と比較するとハイブリッド並列は多少並列性能が劣化しています。

- 実効効率(ピーク性能比)

ベクトル型スーパーコンピュータが高性能を発揮しており、SX-7 の MPI 分散並列では実効効率が 90.3%、SMP と MPI の共有分散ハイブリッド並列では実効効率が 76.9%となっています。スカラ型スーパーコンピュータでは SGI Altix がやや高性能で、実効効率が 67.7%となっています。 ただし HPC チャレンジは、実効効率が評価の対象とはなっていません。

-性能評価結果

HPL を実行するには、入力データのサイズ、解法のパラメータ、ブロックサイズのパラメータを決める必要があります。まずは、HPL のサブルーチン毎の計算時間の内訳(今後、このような内訳を「コスト分布」と呼びます)について説明します。表 1 は、入力データサイズを N=30,000、解法パラメータを Right Looking、ブロックサイズパラメータを NB=64 とし P=1、Q=32 として測定した結果です。HPL 全体では、データ生成と結果検証もありますが、性能測定に関係する部分のコストのみを抜き出します。実行は MPI 32 プロセスで、単位は秒です。

表 1　HPL のコスト分布

HPL　(Target portion for measurement)			
94.3　(sec)			
HPL_dgemm calculation	HPL_bcast_1ring communication	HPL_dtrsm calculation	HPL_pdupdateTT calculation
71.8	19.9	1.6	0.9

一番計算時間が長い、すなわち計算コストの一番大きいのは HPL_dgemm で、これは行列積となります。行列積の計算は、BLAS ライブラリの DGEMM を使っています。この時の測定では DGEMM の実効効率は 97%と高い性能が発揮されています。また、HPL_bcast_1ring はデータ転送の部分で、MPI の SEND と RECV の関数を使って転送が行われています。HPL のコストのほとんどがこれら 2 つの部分で占められます。LU 分解のオーダ評価を行うと、演算部分は N の 3 乗に比例しており、転送部分は N の 2 乗に比例しています。したがって、N が大きくなるとほとんどのコストが DGEMM になり、ベースラインランでの HPL の実効効率は 90%程度になると予想されます。

次に、ブロックパラメータ NB, P, Q, NBMIN が HPL 実行結果にどのように影響を及ぼすかについて検討します。まずは、NB=64, NBMIN=64 に固定して評価します。この時、実行時間を節約するため、小さなサイズのデータ N=20,000 で評価した結果が表 2 となります。P と Q が行列データの分散方法を決めるパラメータで、P と Q の積が MPI のプロセス数になります。なお、項目 T/V は HPL の解法を表します。HPL は LU 分解の解法に、3 つの選択肢(Left looking, Crout, Right looking)がありますが、本実験では一番性能が良かった Right Looking(外積型ガウスの消去法)を採用しました。表 2 中 T/V の項目の R の次の数字が NBMIN のサイズとなっています。

表 2　ブロックパラメータ(P, Q)の HPL 性能に対する影響

T/V	N	NB	P	Q	Time	Gflop/s
W10R3R64	20000	64	1	24	33.80	1.578e+02(82.1%)
W10R3R64	20000	64	2	12	33.91	1.573e+02(81.9%)
W10R3R64	20000	64	3	8	35.37	1.508e+02(78.5%)
W10R3R64	20000	64	3	6	46.73	1.141e+02(79.2%)
W10R3R64	20000	64	4	4	51.79	1.030e+02(81.1%)
W10R3R64	20000	64	6	3	48.70	1.095e+02(76.0%)
W10R3R64	20000	64	8	2	55.01	9.696e+01(67.3%)
W10R3R64	20000	64	12	1	78.96	6.756e+01(70.3%)

この結果を見ると、P=1 に固定するのが一番良い性能となることが分かります。以降 P=1 に固定して評価を続けます。

次に、入力データサイズを N=10,000 に縮小し、プロセス数を 48 に固定(P=1, Q=48)して、NBMIN と NB のパラメータ依存性を調べた結果が表 3 となります（注：このテストは SX-6 6 ノードを使って評価しています）。NBMIN の組み合わせは NBMIN=32、44、64、144 で行い、NB について NB=64、128、256 の±1 前後のパラメータで評価しました。

表 3 プロックパラメータ(NBMIN, NB)の HPL 性能に対する影響

T/V(Change:NBMIN)	N	NB	P	Q	Time	Gflop/s
WC10R3R32	10000	63	1	48	3.16	2.11E+02
WC10R3R44	10000	63	1	48	3.13	2.13E+02
WC10R3R64	10000	63	1	48	3.14	2.12E+02
WC10R3R144	10000	63	1	48	3.75	1.78E+02
WC10R3R32	10000	64	1	48	3.14	2.12E+02
WC10R3R44	10000	64	1	48	3.12	2.14E+02
WC10R3R64	10000	64	1	48	3.1	2.15E+02
WC10R3R144	10000	64	1	48	3.08	2.17E+02
WC10R3R32	10000	65	1	48	3.22	2.07E+02
WC10R3R44	10000	65	1	48	3.37	1.98E+02
WC10R3R64	10000	65	1	48	3.18	2.10E+02
WC10R3R144	10000	65	1	48	3.16	2.11E+02
WC10R3R32	10000	127	1	48	3.86	1.73E+02
WC10R3R44	10000	127	1	48	3.89	1.72E+02
WC10R3R64	10000	127	1	48	4.06	1.64E+02
WC10R3R144	10000	127	1	48	4.2	1.59E+02
WC10R3R32	10000	128	1	48	3.91	1.71E+02
WC10R3R44	10000	128	1	48	3.61	1.85E+02
WC10R3R64	10000	128	1	48	3.98	1.67E+02
WC10R3R144	10000	128	1	48	4.17	1.60E+02
WC10R3R32	10000	129	1	48	4.06	1.64E+02
WC10R3R44	10000	129	1	48	4.04	1.65E+02
WC10R3R64	10000	129	1	48	4.16	1.60E+02
WC10R3R144	10000	129	1	48	4.28	1.56E+02
WC10R3R32	10000	255	1	48	5.97	1.12E+02
WC10R3R44	10000	255	1	48	6.05	1.10E+02
WC10R3R64	10000	255	1	48	6.16	1.08E+02
WC10R3R144	10000	255	1	48	6.55	1.02E+02
WC10R3R32	10000	256	1	48	5.92	1.13E+02
WC10R3R44	10000	256	1	48	6.22	1.07E+02
WC10R3R64	10000	256	1	48	6.19	1.08E+02
WC10R3R144	10000	256	1	48	6.5	1.03E+02
WC10R3R32	10000	257	1	48	6.13	1.09E+02
WC10R3R44	10000	257	1	48	6.13	1.09E+02
WC10R3R64	10000	257	1	48	6.14	1.09E+02
WC10R3R144	10000	257	1	48	6.56	1.02E+02

結果を見ると、表 3 から NBMIN に関してはパラメータ依存性がはっきりしませんが、NBMIN=64、144 は性能劣化していることが確認されました。NB に関しては、サイズを大きくすると性能劣化していること分かりました。以上の結果から HPL のブロックパラメータは、NBMIN=32 または 44、NB=64 が望ましいことが分かり、入力データを大きくする時には、このパ

ラメータを採用することにしました。

最後に、2004 年 12 月現在登録されている HPL の値を図 1 に示します。全体ノードが大きいほど高性能となるため、シングルノードの SX-7 の順位は 49 位中 MPI 並列版が 31 位、SMP +MPI 並列版が 39 位となっています。トップは 252CPU 構成の Cray X1 であり、24 ノード(192CPU)構成の SX-6 が、Cray X1 に続くグループとなっています。

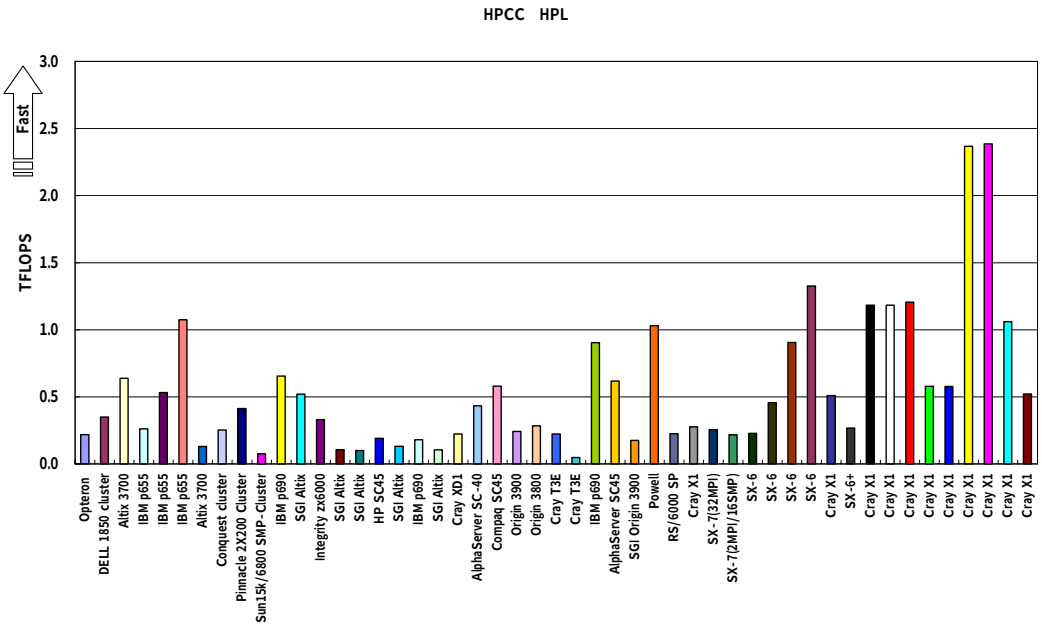

図 1　HPL の結果

5．2 DGEMM

- データサイズ

DGEMM のデータサイズは、HPL のデータサイズ N と MPI のプロセス数から次の式で決められます。

DGEMM データサイズ　= HPL N／2／MPI プロセス数の平方根

- SMP 並列処理の導入効果

十分に SMP 並列化された BLAS を利用することで、DGEMM のテストは高度に並列化された結果が得られます。

- 実効効率

シングル環境(SN)での DGEMM の実効効率はピーク性能に近い 99%以上の性能を発揮します。また多重負荷環境(EP)においても実効効率は 93%以上となっています。

- 性能評価結果

2004 年 12 月現在登録されているシングル環境(SN)の結果を図 2 に示します。データのないプラットフォームもありますが、これは古いバージョンの HPCC の結果で、DGEMM の項目が入っていない時期のものとなっています。SX-7 は、高性能な 16CPU を使った SMP 共有並列処理により、他のシステムと比較して圧倒的に優れた演算性能を示しています。

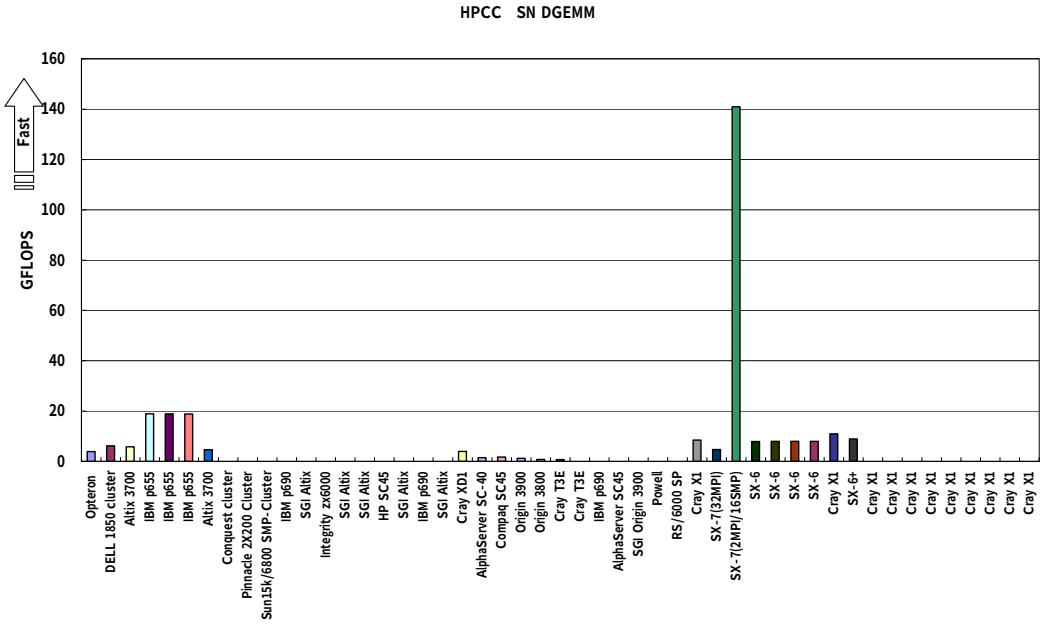

図 2　DGEMM の結果（シングル環境）

5．3 STREAM

- データサイズ

STREAM の配列サイズは、HPL のデータサイズ N と MPI のプロセス数から次の式で決められます。

STREAM 配列サイズ = HPL N^2／MPI プロセス数／3

- SMP 並列処理の導入効果

オリジナルコードに OpenMP による SMP 並列化の指示行が入っており、並列化されます。実行するとノード内において並列化された結果が得られます。表 4、5 は全 32CPU 中、SMP 並列に 16CPU、多重負荷として 2 プロセス使用して実行した結果です。STREAM 配列サイズ =620,166,666 としました。多重負荷環境(EP_STREAM)は、シングル環境(SN_STREAM)での実行に比べてメモリ負荷がかかるため、性能が劣化していることが分かります。

表 4　多重負荷環境での STREAM 性能

EP_STREAM_Copy	389.791 GB/s
EP_STREAM_Scale	348.593 GB/s
EP_STREAM_Add	428.084 GB/s
EP_STREAM_Triad	492.161 GB/s

表 5　シングル環境での STREAM 性能

SN_STREAM_Copy	537.486 GB/s
SN_STREAM_Scale	379.734 GB/s
SN_STREAM_Add	437.240 GB/s
SN_STREAM_Triad	556.609 GB/s

-性能評価結果

2004 年 12 月現在登録されているシングル環境の積和の結果(SN_STREAM_Triad)と多重負荷環境の積和の結果(EP_STREAM_Triad)を図 3、4 に示します。ベクトルロードストアユニットを有するベクトル機のメモリ性能の良さが際立っています。特に SX-7 は共有並列を最大限に活かされた結果となっており、16CPU を使った SMP 共有並列の高性能が発揮されて、圧倒的に優れたメモリ性能が示されています。

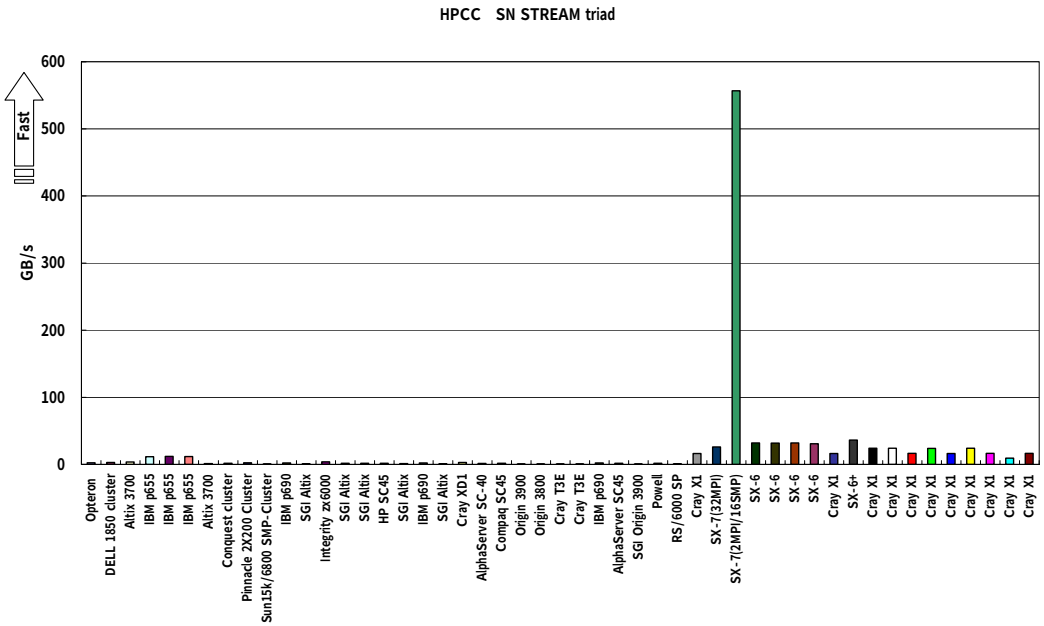

図 3 シングル環境 SN_STREAM_triad の結果

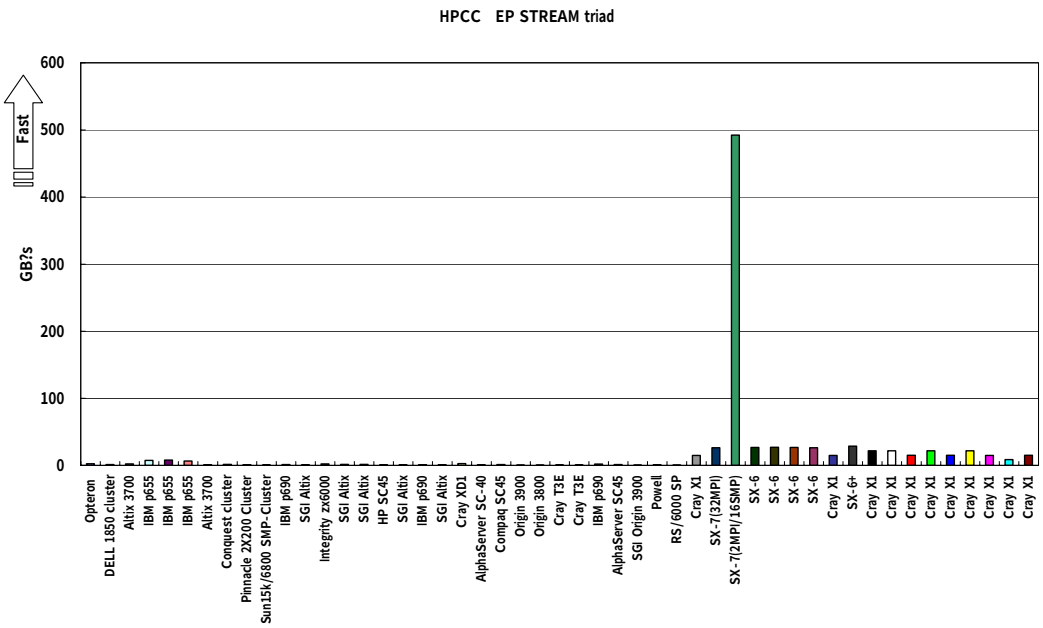

図 4　多重負荷環境 EP_STREAM_triad の結果

5．4 PTRANS

- データサイズ

PTRANS のサイズは、HPL のデータサイズと同じ N、または個別に PTRANS のサイズ N を決めることができます。転置される配列のサイズは、N／2 となっています。

-性能評価結果

PTRANS は MPI のデータ転送による通信部分(sendrecv)が、コストの大部分となっています。測定区間の転送処理全体のコスト分布(実行時間の内訳)は表 6 のようになります。この時使用したパラメータは、N=90,000、NB=471、P=1、Q=32 としました。

表 6　PTRANS のコスト分布

ptr_trans (Target portion for measurement)				
0.651　(sec)				
intrans-1,2	dtr2mx	dtr2b	sendrecv	intrans-4
0.000	0.006	0.038	0.541	0.066

まずは、通信について説明します。行列の転置を行うため、次の図 5 に示されているような転送方法に最適化されています。すなわち、データ転送は、(自分のランク番号－1)のランクから始まって、ランク番号を減少させていく方向に、それぞれのプロセスがデータを転送します。更にランク 0 のプロセスの後に、最後のランクから始まってランク番号を減らす方向に同じ方法でデータ転送が行われます。これに対応して、データの受信は、(自分のランク番号+1)のランクから始まってランク番号が増えていく方向に行われます。このようなスケジューリングに基づいて、

データの送受信の競合を避けるように最適化が行われています。

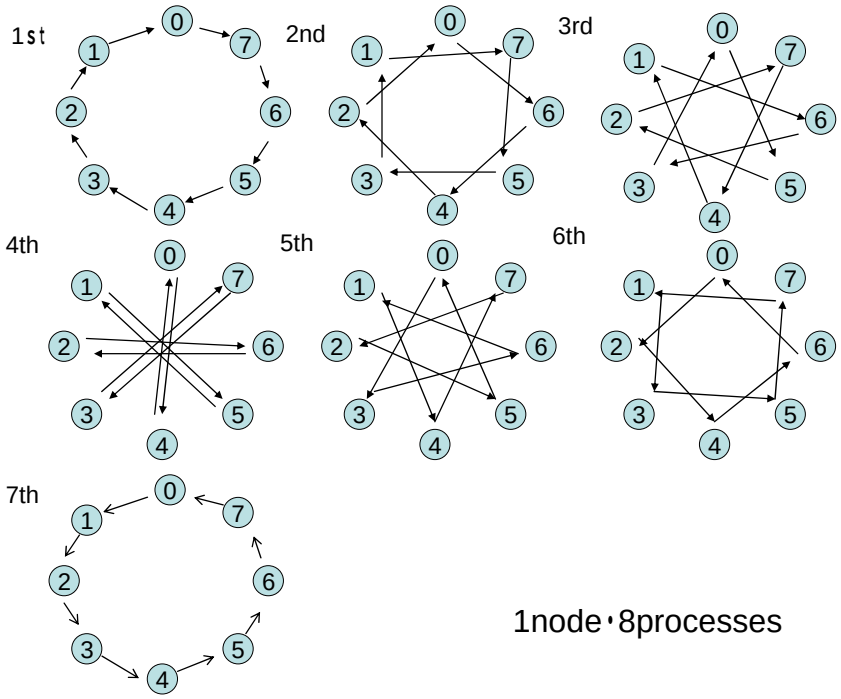

図 5 最適な転置方法

PTRANS の性能に関して、通信コストが最も大きな要素でありますが、通信以外の処理コストの時間も BM 計測区間の大きな要素になっています。このコストを調べるため、N=10,000 に固定して NB を色々な組み合わせで評価しました。まずは、評価区間内の性能(Performance)と転送時間(transmission)、バンク競合時間(bank)を調査しました。表 7 に結果を示します。

表 7 PTRANS コスト分布

N	Performance (GB/s)	transmission (sec)	bank (sec)	Number of processes
10000	7.423	0.024	0.0062	4(NB= 90)
10000	7.871	0.023	0.0066	4(NB=100)
10000	7.035	0.026	0.0111	4(NB=104)
10000	8.543	0.022	0.0064	4(NB=105)
10000	11.231	0.014	0.0011	8(NB=105)
10000	7.091	0.026	0.0112	4(NB=106)
10000	6.496	0.029	0.0118	4(NB=110)
10000	6.424	0.029	0.0119	4(NB=120)
10000	5.972	0.031	0.0131	4(NB=200)

測定区間に含まれるバンク競合時間が、転送時間と同じ程度の時間になるため、性能全体に影響していることがわかります。また、コスト分布から、非通信部分のコストは 17%も占められていることがわかり、この部分を高速化することにより、全体の性能を改善させることが出来ます。これを調べるため、NB、P、Q の様々な組み合わせを用いて測定した結果が以下の表 8 となります。一番右側の RESID が検証結果で、0.00 が正しい結果であることを示しています。

表 8　PTRANS ブロックパラメータの評価

TIME	M	N	MB	NB	P	Q	TIME	CHECK	GB/s	RESID
WALL	45,000	45,000	150	150	1	8	0.94	PASSED	17.266	0.00
WALL	45,000	45,000	187	187	1	8	0.84	PASSED	19.237	0.00
WALL	45,000	45,000	241	241	1	8	1.16	PASSED	13.952	0.00
WALL	45,000	45,000	255	255	1	8	0.82	PASSED	19.828	0.00
WALL	45,000	45,000	255	255	2	4	4.15	PASSED	3.901	0.00
WALL	45,000	45,000	471	471	1	8	0.93	PASSED	17.356	0.00
WALL	45,000	45,000	105	105	1	16	1.09	PASSED	14.889	0.00
WALL	45,000	45,000	143	143	1	16	1.05	PASSED	15.427	0.00
WALL	40,000	40,000	147	147	1	16	0.89	PASSED	14.336	0.00
WALL	45,000	45,000	150	150	1	16	1.01	PASSED	15.991	0.00
WALL	45,000	45,000	187	187	1	16	1.45	PASSED	11.166	0.00
WALL	45,000	45,000	200	200	1	16	1.15	PASSED	14.050	0.00
WALL	45,000	45,000	241	241	1	16	1.01	PASSED	15.992	0.00
WALL	45,000	45,000	255	255	1	16	1.13	PASSED	14.386	0.00
WALL	45,000	45,000	300	300	1	16	1.21	PASSED	13.437	0.00
WALL	45,000	45,000	450	450	1	16	1.12	PASSED	14.491	0.00
WALL	45,000	45,000	471	471	1	16	0.93	PASSED	17.353	0.00
WALL	45,000	45,000	150	150	1	32	0.76	PASSED	21.208	0.00
WALL	45,000	45,000	187	187	1	32	0.81	PASSED	19.944	0.00
WALL	45,000	45,000	241	241	1	32	0.71	PASSED	22.757	0.00
WALL	45,000	45,000	255	255	1	32	0.77	PASSED	20.995	0.00
WALL	45,000	45,000	471	471	1	32	0.66	PASSED	24.569	0.00

表 8 を見ると、最も性能を支配しているパラメータは NB で、適切な NB の値を選ぶことが、PTRANS の性能を決める重要な要素となっていることがわかります。コードを解析すると、通信と通信以外の両方にパラメータ NB が影響しています。NB 値は、通信以外の処理ではベクトル長になっており、この NB 値によってループのストライドが変化するために、バンク競合が発生するコードになっています。また、バンクコンフリクトを回避するため、NB を奇数にすることが考えられますが、通信処理の影響で必ずしも奇数とすることが最良の結果を得ることにならないことも分かりました。

2004 年 12 月現在登録されている PTRANS の値を図 6 に示します。ノード数が大きいほど高性能となるため、シングルノードの SX-7 の順位は低くなっています。トップは 252CPU 構成の Cray X1 であり、24 ノード(192CPU)構成の SX-6 が、それに続きます。

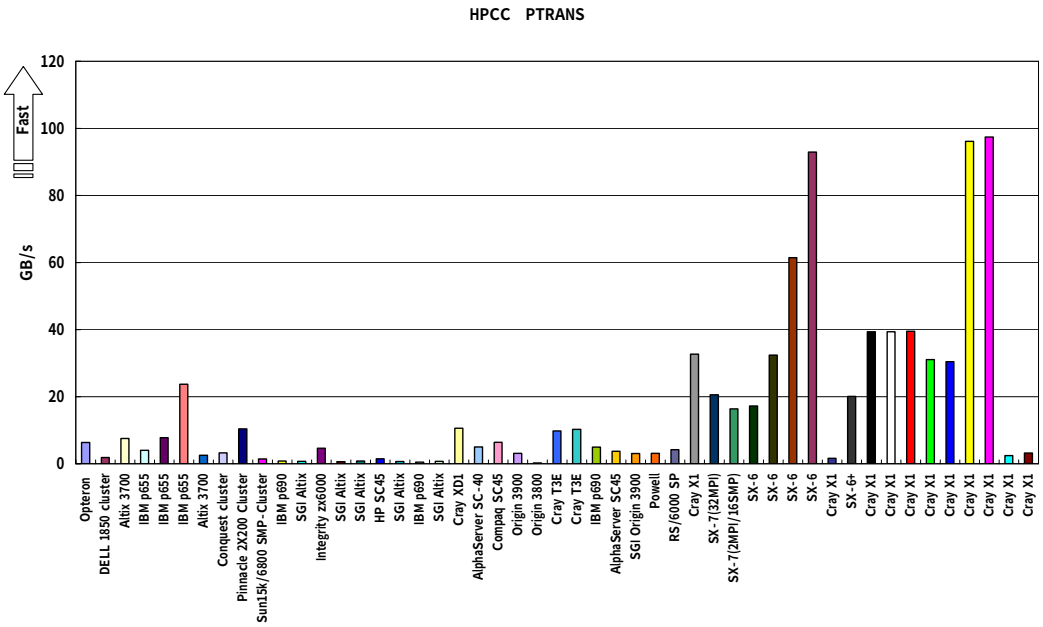

図 6 PTRANS の結果

5．5 RandomAccess

- データサイズ

ノード単体テストの RandomAccess のテーブルサイズ（rand の範囲）と更新領域サイズ（配列 Table の大きさ）は、HPL のデータサイズ N と MPI のプロセス数から次の式で決められます。

テーブルサイズ= HPL N^2／MPI プロセス数

更新領域サイズ＝テーブルサイズ×4

- メモリアクセス方法

ノード単体テストでは、ノード内のメモリのランダムアクセス（インダイレクトアクセス）性能を評価します。図 7 で示すように、乱数で作成されたテーブル rand(i)に基づき、配列データを順次格納して、メモリのランダムアクセス性能を評価します。

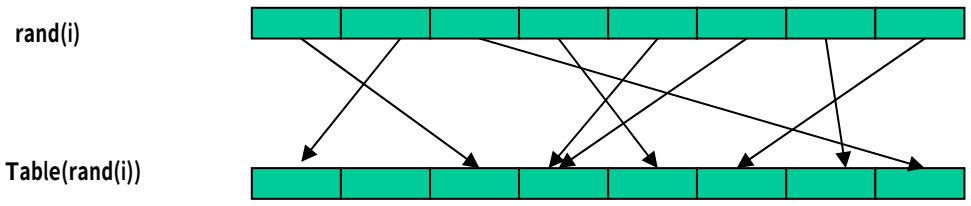

図 7 ランダムアクセスの方法

- 性能評価結果

シングル環境(SN)と多重負荷環境(EP)の両方において、ベクトル機はメモリランダムアクセス性能が高いため、SX の性能は高い順位に位置しています。なお、ランダムアクセスのループ長は 128 と短いループ長に固定されているため SMP 並列化が困難となっています。また全ノードテストでは、ノード間の MPI 転送性能の評価を目的としており、同時データアクセス性が評価され、トータル転送パスが多いほど高性能となっています。2004 年 12 月現在登録されているシングル環境の結果を図 8 に示します。ノード単位テストでは、SX、Cray X1 などのベクトル機がトップグループになっており、ベクトル型スーパーコンピュータのメモリアクセス性能の良さが評価されています。

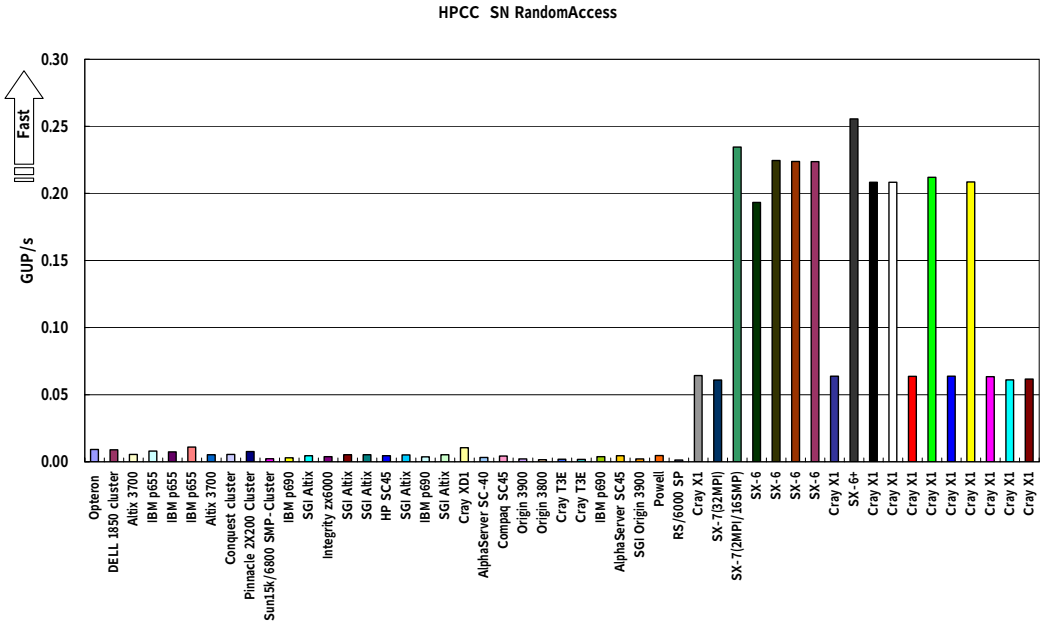

図 8　シングル環境の RandomAccess の結果

また、図 9 は MPI による全ノードテストの結果です。全ノードテストは、ネットワーク全体の総合性能が評価されるため、ノード（CPU）数の少ない SX は高い性能とはなっていません。

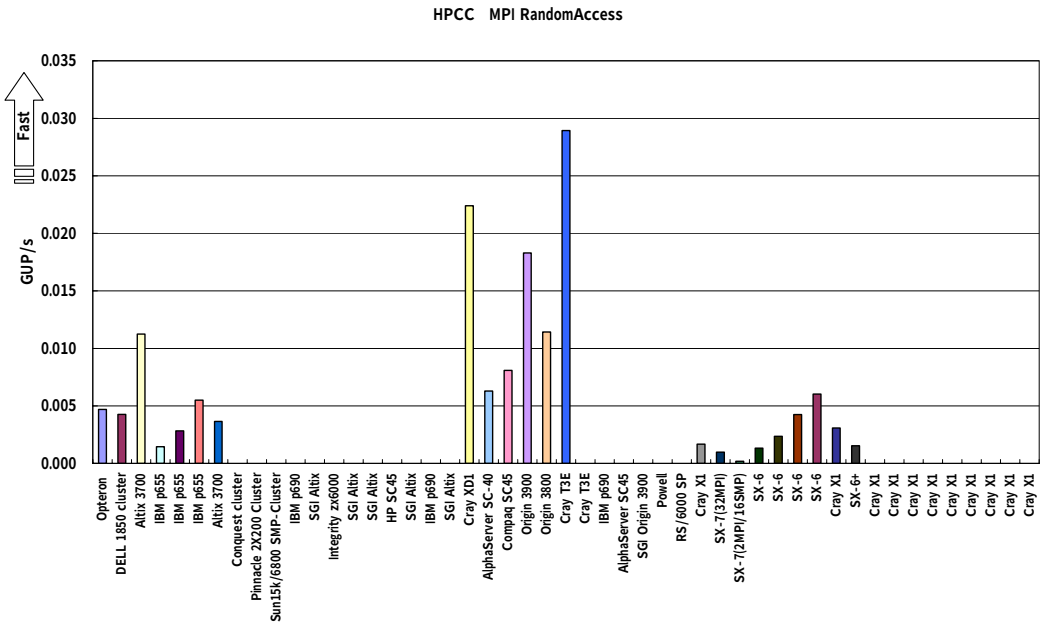

図 9　全ノードテストの RandomAccess の結果

5．6 FFTE

- データサイズ

FFT のサイズは、HPL のデータサイズ N と MPI のプロセス数から、次の計算式の値を超えない最大の 2 のべき乗の値となっています。

N^2／プロセス数／2／(struct fftw_complex のサイズ)　(SN_FFTE, EP_FFTE)

N^2／プロセス数／3／(struct fftw_complex のサイズ)　(G_FFTE)

-性能評価結果

FFT のコードの中で、L2SIZE というパラメータがあり、キャッシュサイズを指定するのに使われています。L2SIZE の値は使用したコンピュータハードウェアに合わせて変更する必要がありますが、ベースラインランの実行ルールでは、このパラメータの変更は許されておらず、固定のサイズとなっています。このように現時点では、FFT のコードは、十分最適化されておらず、登録された各マシンの評価結果は不十分な値となっています。

2004 年 12 月現在登録されているシングル環境の結果を図 10 に示します。シングル環境では、SX-7 がトップとなっています。また、図 11 に示す全ノードテストでは、SX システムがトップグループとなっています。

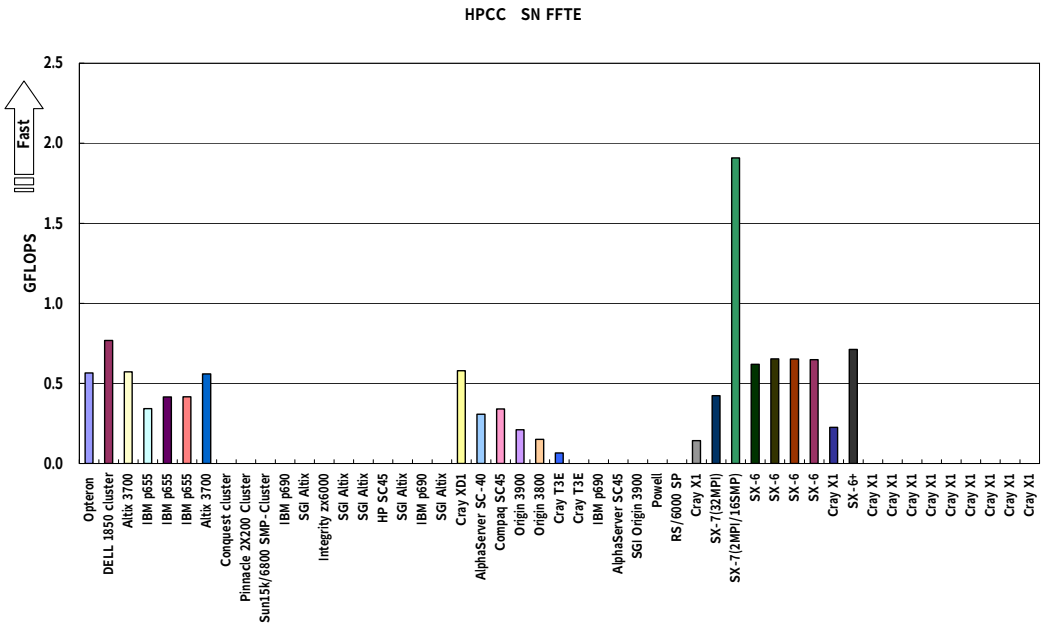

図 10　シングル環境の SN_FFTE の結果

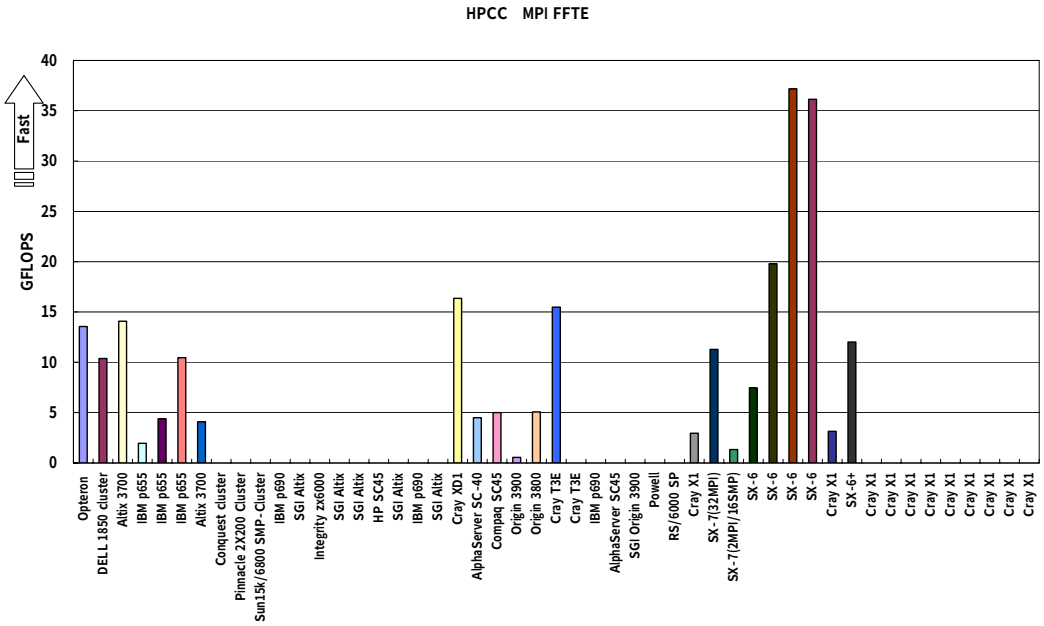

図 11　全ノードテストの G_FFTE の結果

5．7 Communication bandwidth and latency

- 転送スキーム

バンド幅およびレイテンシの評価のスキームは Ping-Pong 転送スキームと Ring 転送スキームの 2 つあります。このうち、Ring 転送スキームはデータを MPI のランク順に転送する方法と、ランダムに転送する方法があります。この 2 つのスキームを使って、バンド幅テストでは 2M バイトのデータを転送し、レイテンシーテストでは 8 バイトのデータを転送してデータ転送性能の評価を行います。図 12 は、これら 2 つの転送スキームを簡単に説明したものです。

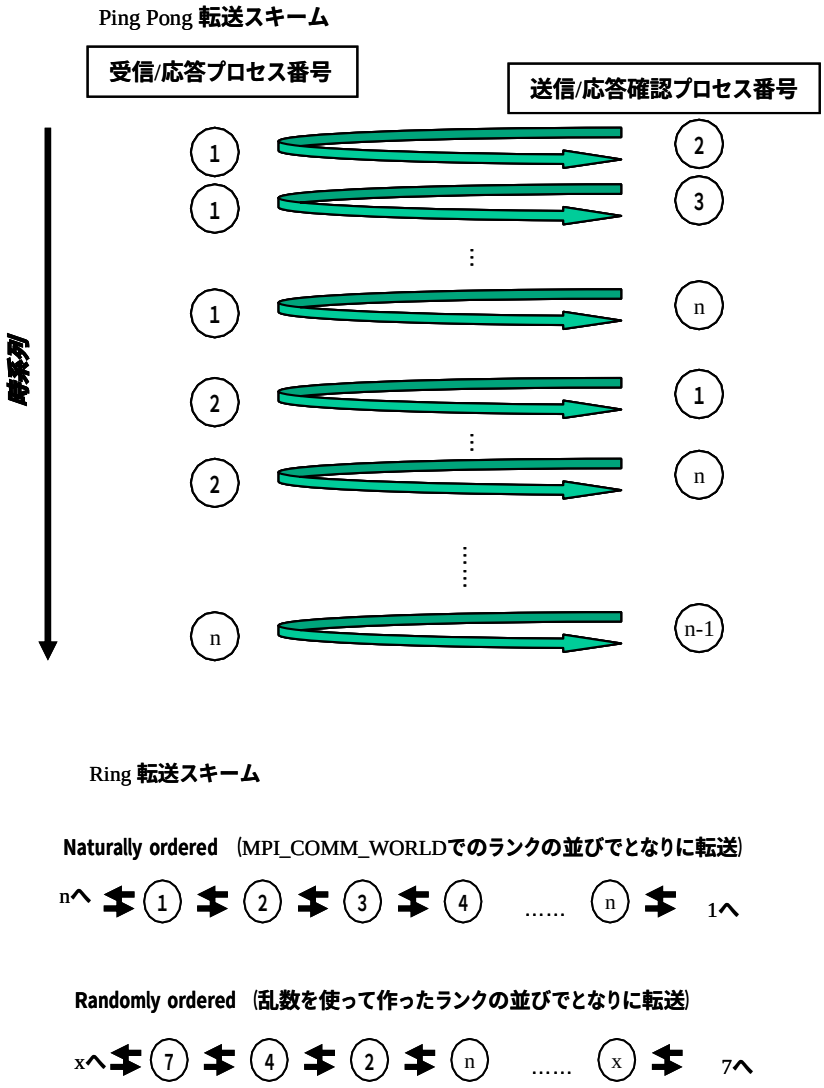

図 12 転送スキーム

-性能評価結果

Communication bandwidth and latency の内、2004 年 12 月現在登録されているレイテンシ 5 項目の内、Randomly ordered Ring のレイテンシの性能を図 13 に示します。SX-7 はノード内の転送のため、比較的高性能の部類に属します。最速は、Cray XD1 になっています。

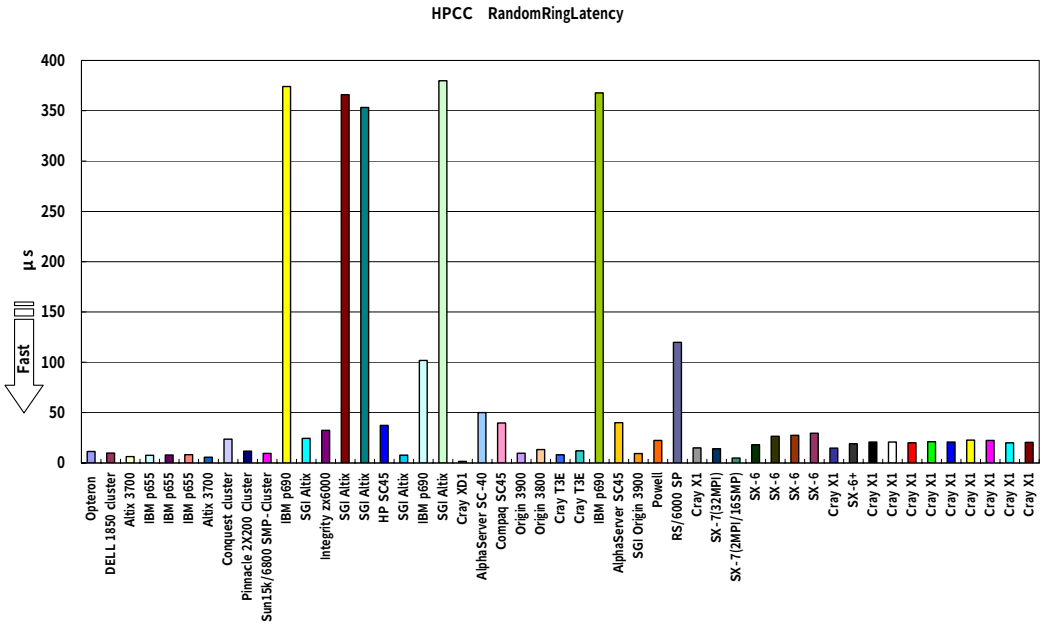

図 13　Randomly ordered Ring のレイテンシの性能

バンド幅の転送性能を見ると、Naturally Ordered Ring と Randomly Ordered Ring の性能に大きな差があります。この差はノード間の転送データ量の合計の差によるものであり、Naturally Ordered Ring の場合、ノード間転送を最小にするための最適化を簡単に行えますが、Randomly Ordered Ring は通信パターンが複雑なため最適化が困難になっています。2004 年 12 月現在登録されているバンド幅 5 項目の内、Naturally Ordered Ring のバンド幅の結果を図 14 に示します。SX はトップグループになっていることが分かります。

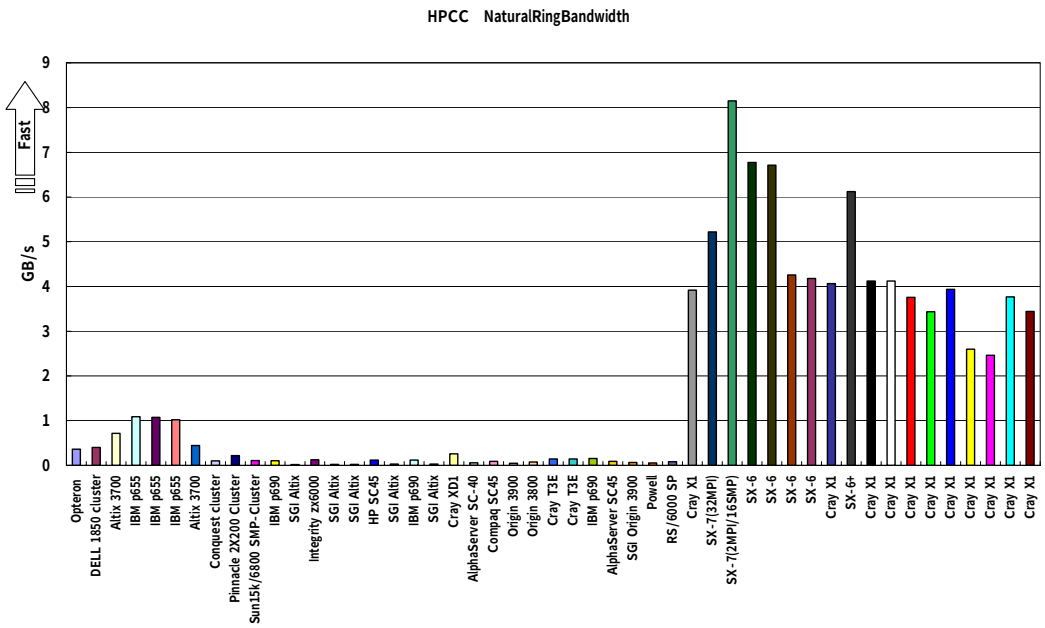

図 14　Naturally Ordered Ring のバンド幅の性能

6．考察

6．1　HPCC のアプローチに対して評価できる点

1)Linpack 以外の本格的な HPC 領域の BM プログラムとなっている点があげられます。性能評価として、Linpack のような演算に限定された測定指標と異なり、メモリバンド幅性能、ネットワーク性能、基本カーネル（まだ不十分ながら行列 DGEMM、FFT）が含まれており、より総合評価となる測定指標へのアプローチとなっています。

2)主要な HPC ユーザおよびベンダが参画している点をあげられます。具体的には、CRAY X1、IBM Power5、SGI Altix など、ベクトル型およびスカラ型の主要なプラットフォームの性能が登録済みとなっています。また、SX の結果も、東北大からだけでなく、HLRS（独）、DKRZ（独）からも登録されています。

3)運営・評価体制がそれなりに確立されている点があげられます。HPC チャレンジプログラムは DoD/DARPA の援助を受け実施されており、委員も J. Dongarra 博士(Linpack ベンチマーク運営)、McCalpin 博士（STREAM ベンチマークの運営者）、と HPC ベンチマークの中心メンバーが参加しています。

6．2　HPCC のアプローチにおける課題

1)総合指標の確立

複数の評価指標の集合体であり、総合指標が存在せず一義的な解釈困難となっている点があげられます。従って、総合指標の作成には、なお試行を要すると思います。これに対する 1 つ

の答えとして、我々は、図 15 に示すような評価方法を考えました。登録されている全結果の項目の順位をレーダーチャートで表し、外側が第 1 位として順位で正規化しています。外側にあるほど高順位であることを示しています。SX-7 の登録された結果（SMP 実行の方）で、SX-7 単体は 32CPU 構成であるため、HPL では順序が 49 位中 39 位としながら、メモリバンド幅や、ネットワーク性能、行列積、FFT で 1 位であることがわかります。このような方法で総合評価するのも一つの方法だと思います。

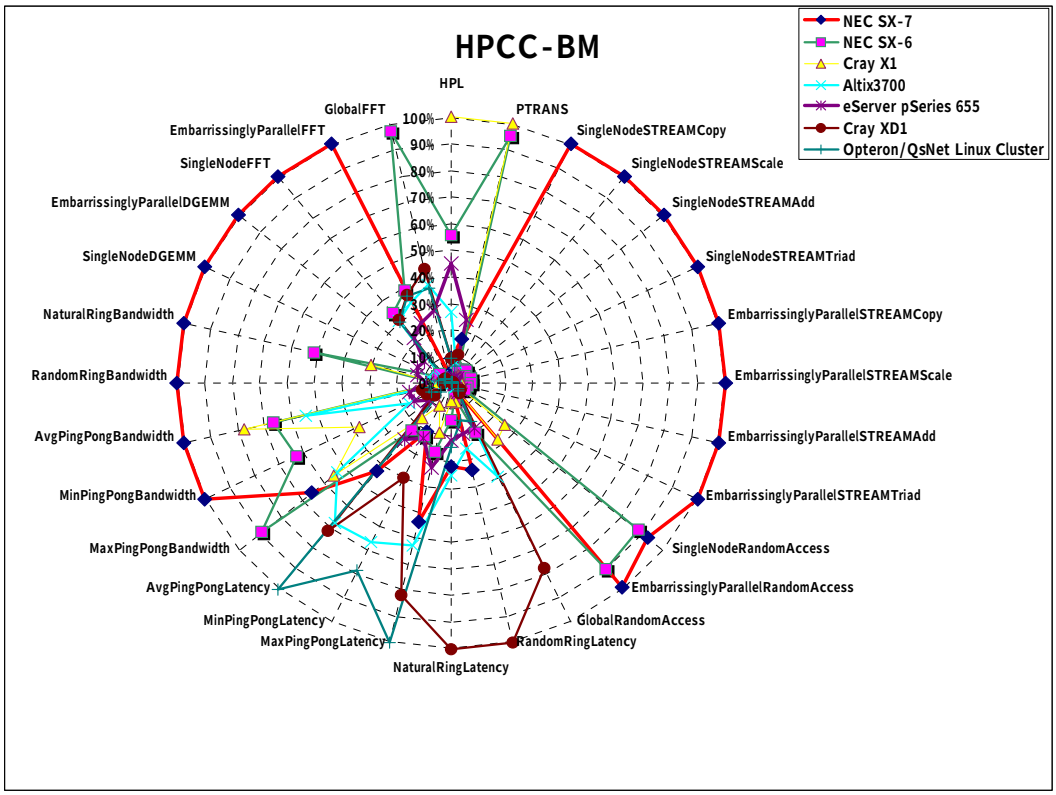

図 15　HPC チャレンジ順位のレーダーチャート

2)評価項目間での一貫性、公平性の確保

計測対象機の最大性能（演算、メモリ、ネットワーク）を「量」で評価する項目があげられます。Top500 の Linpack HPC のように、「量」のみで大規模なシステムを評価することも一つの見方ですが、この「量」でのみの性能評価とすると、Top500 と同じように今後単一ボリュームの大きなシステムしか性能評価のレースに参加できないという問題もでてきます。実際の計算では、ボリュームが大きくなると効率が下がる傾向にありますので、これを評価する指標として量による総性能評価に加えてそのときの稼働率、すなわち実効効率を評価することが大切だと考えます。必要な計算結果を効率よく得ることも一つの評価項目になるとなりますので、最大性能を評価する場合、「量」とその「実効効率」も評価の対象となることを期待しています。

3)簡明性の向上

指標の解釈には、内容の理解が必要なため、評価結果がわかりにくいという問題点があります。今後 HPC コミュニティやメディアを通したより一層の広報･普及活動を推進する必要があると思います。

7． おわりに

以上、HPCチャレンジによるSX-7の評価について述べました。様々な角度からスパコンを評価するHPCチャレンジベンチマークにおいて、情報シナジーセンターのSX-7は、28評価項目中16項目で最高性能を得ました。ベクトル型スーパーコンピュータのメモリ性能の高さに加え、SX-7では、SMP並列で32までと大きな共有並列化することができることから、HPC分野で高い潜在能力を持っていることを明らかにしました。情報シナジーセンターのSX-7のHPCチャレンジベンチマーク評価結果の登録[3]に対し、J. Dongarra 博士からは以下のようなコメントを受けています。” We are impressed with the continuing high performance of the SX family of processors. The SX-7 lives up to the expectations.”

情報シナジーセンターの SX システムは、8 ノード(240CPU)からなる総性能 2.1Tflop/s のシステムですが、24 時間フル稼働の現在、常に 85%以上の CPU 利用率（2003～2004 年の実績で毎年度 6 月以降 90%以上、11 月以降 95%以上、実行待ちジョブ毎日 70～80 件）で動作しており、学内外の多くの研究者に活用されております。このような高性能なスーパーコンピュータシステムを利用して、学会論文ばかりでなく、新聞紙上を賑わすような数多くの研究成果が生み出されております[4][5][6]。

HPCチャレンジベンチマークによるスーパーコンピュータの評価の試みはまだ始まったばかりで、今後、ベンダやユーザからのフィードバックを得ながら様々な改良が加えられ、スーパーコンピュータの総合的な評価指標として確立されていくと思われます。現在、HPCを支えるスーパーコンピュータシステムとして、ベクトル型スーパーコンピュータのようなカスタム設計によるもの、スカラ並列スーパーコンピュータのCOTS (Commercial Off-the-Shelf, 商用量産品)ベースのもの、そしてPCクラスタやGridなど様々なものがありますが、米国では、市場性重視でのHPCシステムの研究開発の危うさを2004年6月にHPC特別委員会報告書で指摘し[7]、米国システムが中心のTop500リスト中の296システムを占める高性能クラスタデザインによるスーパーコンピュータでは、国家安全保障の要求水準を満たすには不十分といった議論がなされています。また、2004年11月に米国ピッツバーグで開催されたSC2004では、ベクトル型スーパーコンピュータ（地球シミュレータ, CRAY-X1）とスカラ並列型スーパーコンピュータ（SGI Altix, IBM Power3/4）の実用的なアプリケーションを用いた性能比較の報告が米国Lawrence Berkeley National Laboratoryの研究グループからあり、運用開始後3年近くたった今でもベクトル並列型である地球シミュレータの実効性能の高さが示されました[8]。そのような背景の中、スーパーコンピュータの新しい評価ベンチマークの研究開発プロジェクトであるHPCチャレンジベンチマークが重要視されております。加えて、同年11月には「高性能計算再生法」が可決され、大統領署名をもって今後3年間に総額1億6600万ドルの予算がDOE(エネルギー省)が中心となって、HECS(High-End Computing Systems)の研究開発に投入されることが決まりました[9]。米国では、IBM BlueGene/Lが2004年11月のtop500ランキングで1位になった現在においても積極的、かつ継続的にHECS/HPCS研究開発計画が推進されています。日本においても、日本の先進科学技術分野における国際競争力を失わないために、実効性能に優れたHECS/HPCSの研究

開発を国策として継続的に支援するとともに、産学官の精力的な取り組みが必要不可欠と思います。

謝辞

今回の実験でご協力いただいた日本電気株式会社第一官庁システム開発事業部の撫佐昭裕氏、神山　典氏、金野浩伸氏に深く感謝いたします。

参考文献

[1] DARPA HPCS Program, http://www.highproductivity.org/.
[2] HPC Challenge, http://icl.cs.utk.edu/hpcc/index.html.
[3] HPCC 評価結果の全登録ﾘｽﾄ, http://icl.cs.utk.edu/hpcc/hpcc_results_all.cgi.
[4] 東北大学大型計算機センター年報（昭和 54 年度～平成 10 年度）.
[5] 東北大学大型計算機センター1999 年度～2000 年度の歩み.
[6] 東北大学情報シナジーセンター年報（平成 13 年度～平成 15 年度）.
[7] Federal Plan for High-End Computing, Report of the High-End Computing Revitalization Task Force (HECRTF), May 10, 2004.
[8] Leonid Oliker, et al., “Scientific Computations on Modern Parallel Vector Systems,” Proceedings of SC 2004, CD-ROM, 2004.
[9] Department of Energy High-End Computing Revitalization Act of 2004, http://thomas.loc.gov/, 2004.