Vol. 46 No. SIG 12(ACS 11) 情報処理学会論文誌：コンピューティングシステム Aug. 2005

HPC Challengeベンチマークを用いたSX-7システムの性能評価

滝沢 寛之[†1,†2] 小久保 達信[†3]
片海 健亮[†4] 小林 広明[†2,†1]

HPC Challenge（以下HPCCとする）ベンチマークは，高性能計算（High-Performance Computing，以下HPC）システムの総合的な性能評価のために提唱されているベンチマーク集である．現在までに広く用いられている浮動小数点演算性能に加えて，メモリアクセスやネットワーク通信の性能等，複数の観点から多角的にHPCシステムを評価することにより，HPCCベンチマークは実用的な科学技術計算に対する実効性能を適切に評価する指標として期待されている．本論文では，東北大学情報シナジーセンターで運用しているNEC SX-7システムの性能をHPCCベンチマークを用いて評価した結果について述べる．28の評価項目のうち16項目において著しく高い評価が得られた結果に基づいて，HPC分野におけるベクトル型アーキテクチャの優位性について議論する．

Performance Evaluation of the SX-7 System Using the HPC Challenge Benchmark

Hiroyuki Takizawa,[†1,†2] Tatsunobu Kokubo,[†3]
Kenryo Kataumi[†4] and Hiroaki Kobayashi[†2,†1]

The HPC challenge benchmark (HPCC) is a benchmark suite developed for comprehensive performance evaluation of high-performance computing (HPC) systems. HPCC is promising to appropriately evaluate the effective performance of HPC systems for practical scientific computing, due to its multilateral evaluation from several viewpoints, such as memory access and networking performances, along with the floating-point operation rate widely used until now. In this paper, we report the performance evaluation results of an NEC SX-7 system of Information Synergy Center, Tohoku University, using the HPCC benchmark. Based on the results that the system can get excellent scores in 16 of 28 tests in the benchmark, we discuss the superiority of its vector architecture in the field of HPC.

1. はじめに

コンピュータシミュレーションは理論，実験に続く第3の手法として先進科学技術の推進に重要な役割を果たしている．その大規模高度シミュレーションの高効率，高速な実行に必要不可欠なスーパコンピュータの性能の指標として，Top500プロジェクトは1993年から年に2回，世界最高性能の500台のスーパコンピュータを集計し，Top500リストとして公開してきた[1]．同プロジェクトは，LINPACKベンチマーク[2]実行時の最大浮動小数点演算性能（R_{max}値）に基づいてHPCシステムの性能を評価し，順位付けしている．

2004年11月に公開されたリストによると，10位以内にベクトル型プロセッサを用いたシステム（ベクトル並列型スーパコンピュータ）は地球シミュレータのみであり，同プロジェクト開始時（1993年6月）にはリストの66.8%を占めていたベクトル並列型スーパコンピュータが，2004年11月にはわずか4.2%を占めるのみとなっている．このように，Top500リストではスカラ型プロセッサによる大規模並列システム（スカラ並列型スーパコンピュータ）が優位である一方で，近年，ベクトル並列型スーパコンピュータの実効性能の高さが再び見直されつつある．たとえば，Olikerらは4つの実用的な科学技術計算の実行結果からベクトル並列型スーパコンピュータである地球シミュレータおよびCray X1と，スカラ並列型スーパコンピュータであるSGI AltixおよびIBM Power3/4ベースのシステムとを性能比較し，ベクトル並列型スーパコン

†1 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
†2 東北大学情報シナジーセンター
Information Synergy Center, Tohoku University
†3 日本電気株式会社
NEC Corporation
†4 NECソフト株式会社
NEC Soft, Ltd.

ピュータの実効性能の高さを実証している[3]．

Top500 プロジェクトで採用されている LINPACK ベンチマークのような単一指標だけでは，HPC システムの性能のほんの一面しか評価できず，実際のアプリケーションにおいて高い性能を発揮できる HPC システムの研究開発のためには不十分との指摘もある[4]．実際，Oliker らの報告からも明らかなとおり，LINPACK の評価結果は HPC システムに求められる現実的な科学技術計算に対する実効性能とは必ずしも一致しない．このような背景から，現実的な科学技術計算に対する実効性能をより適切に評価できる新しいベンチマークが強く求められており，LINPACK ベンチマークを補完する複合的なベンチマーク集である HPCC ベンチマークが開発されている[5]．

本論文では，東北大学情報シナジーセンターで運用しているベクトル並列型スーパコンピュータ NEC SX-7 システムを，HPCC ベンチマークを用いて性能評価した結果について述べる．従来の LINPACK ベンチマークでは適切に評価されてこなかった性能差が，HPCC ベンチマークでは評価結果に顕著に反映されることを示す．その結果から，ベクトル並列型スーパコンピュータの HPC 分野における優位性について考察する．

2. HPCC ベンチマーク

HPCC ベンチマークは HPC システムの性能を多様な観点から計測し，現実的なアプリケーションにおける演算性能をより適切に評価することを目的として，Dongarra らが米国 DARPA の支援を受けて開発しているベンチマーク集である[5]．2004 年 10 月 19 日付けで公開されている最新☆のバージョン 0.8β は，システム性能を多角的に評価するために従来の LINPACK (High-Performance Linpack，HPL) を含む 7 つのベンチマークから構成されている．従来より重要視されてきた浮動小数点演算性能に加えて，高い実効性能を達成するうえで重要なメモリアクセス性能やネットワークを介したデータ転送速度，様々なアプリケーションで頻繁に利用されるカーネルコードに対する性能等を測定し，特定のアプリケーションに偏らない性能評価の実現を目指している．

HPCC ベンチマークのソースコードのコンパイルにより，7 つのベンチマークすべてを行う実行ファイルが 1 つだけ生成される．設定可能なパラメータは従来の HPL とほぼ同じであり，HPL のパラメータに基づいて他のベンチマークのパラメータが算出される☆☆．したがって，ベンチマークごとに最適にパラメータをチューニングすることは許されておらず，1 組のパラメータ設定で全項目の評価を一括して行うことが義務づけられている．システム性能計測結果として，合計 28 の数値が得られる．

近年の大規模 HPC システムでは，SMP 等によるメモリ共有型並列計算機をノードとして定義し，複数ノードを高性能ネットワークで接続することで分散メモリ並列計算による大規模化を実現する構築例が多い．このような複雑な構成となっている場合，システム全体性能とノード単体性能を区別して評価することは重要である．HPCC ベンチマークによる HPC システムの評価方法には，全ノードを使ったシステム総合性能評価 (Global，G) とノード単体の性能評価があり，後者には単一プロセスでの評価 (Single Node，SN) と多重負荷時の評価 (Embarrassingly Parallel，EP) の 2 種類が用意されている．SMP ノード内ではメモリが共有されているため，複数のプロセスを実行する場合にはプロセス間でメモリ競合が発生し，性能が低下する．評価方法 SN は単一のプロセスだけを実行することでメモリの競合を回避し，ノード単体の性能を計測する．一方，評価方法 EP は複数のプロセスを実行することによってメモリ競合を意図的に発生させ，その性能への影響を明らかにする．

さらに，プログラムの実行ルールについてもプログラムコードをいっさい変更しないベースライン実行 (baseline runs) と，コードの修正をともなう最適化を許可するオプティマイズ実行 (optimized runs) の 2 種類が用意されている．2004 年 12 月現在，HPCC のサイト[5] で公開されている評価結果のほとんどがベースライン実行によるものであり，本論文でもベースライン実行のみを扱う．

7 つのベンチマークそれぞれによる評価項目と 3 種類の評価方法 (G，SN，EP) との関係を以下で説明する．

2.1 High-Performance Linpack (HPL)

HPL は LU 分解による連立一次方程式の解法プログラムである．MPI (Message Passing Interface) に基づく分散メモリ型並列計算システムの全ノード利用時の浮動小数点演算性能 (Tflop/s) を，評価方法 G によって実測する．

HPL による評価結果に大きな影響を与えるパラメー

☆ 2004 年 12 月現在．

☆☆ ただし，後述の PTRANS だけは行列サイズ等のパラメータを HPL とは別に設定することも可能．

タとして，問題サイズ N，LU 分解におけるブロックサイズ NB，およびプロセス格子 (P,Q) がある．問題サイズ N に対して，HPL の計算時間は $O(N^3)$，通信時間は $O(N^2)$ に従って増加する．すなわち N を大きくすることで計算時間が実行時間全体に占める割合を大きくし，通信のオーバヘッドを相対的に軽減させることが可能である．また，HPL の計算時間の大半を占めるのが BLAS（Basic Linear Algebra Subroutines）中の DGEMM による行列積の処理時間であり，DGEMM の最適化によって HPL の実効効率（ピーク–実効性能比）を大きく向上させることが期待できる．ブロックサイズ NB は，その設定値の減少にともなって負荷バランスが向上する一方で通信回数が増加する．逆に，NB を増加させることによって通信回数が減る一方で負荷バランスが悪化する．すなわち NB の最適値は，演算性能とプロセス間のデータ転送性能との比に大きく依存しており，実験的に求める必要がある．プロセス格子 (P,Q) は，行列をどのようにして分割してプロセスに割り当てるのかを指示する設定項目である．

また，LU 分解のアルゴリズムも外積法（right-looking），内積法（left-looking），およびクラウト（Crout）法の 3 種類[6] のいずれかを選択可能である．さらに，LU 分解中に発生するブロードキャスト通信方法も，設定項目 $BCAST$ によって指示可能である．

2.2 DGEMM

DGEMM は BLAS 中の倍精度実数行列の積を計算するサブルーチンであり，ノード単体の浮動小数点演算性能（Gflop/s）を評価方法 SN および EP によって測定する．評価方法の違いにより，2 つの測定結果が得られる．

2.3 STREAM

STREAM は Copy，Scale，Sum，Triad の 4 つのテストから構成されており，主に持続可能なメモリ帯域幅（GB/s）を測定するベンチマークである．ノード単体性能を評価方法 SN と EP により評価する．実行するテストと測定方法の違いにより，合計 8 つの測定結果が得られる．

2.4 Parallel matrix TRANSpose (PTRANS)

PTRANS は CPU 間での 1 対 1 同時通信を多用することにより転置行列を計算するプログラムであり，評価方法 G によりネットワークの総データ通信容量（GB/s）を評価する．

2.5 RandomAccess

1 秒間に Read-Modify-Write 処理できる 64 ビットワード数を，ランダムメモリアクセス性能測度 Updates Per Second とする．RandomAccess は Giga Updates Per Second（GUP/s）値を計測するためのプログラムであり，MPI 通信によるノード間のデータアクセス性能を評価方法 G で測定し，さらにノード内でのメモリへのランダムアクセス性能を評価方法 SN および EP で測定する．評価方法の違いにより，3 つの測定結果が得られる．

2.6 FFTE: A Fast Fourier Transform Package

1 次元離散フーリエ変換のカーネルプログラムであり，評価方法 G，SN および EP によりシステム全体およびノード単体での浮動小数点演算性能（Gflop/s）を測定する．評価方法の違いにより，3 つの測定結果が得られる．

2.7 Communication Bandwidth and Latency

プロセス間のデータ転送の帯域幅と遅延を評価するためのプログラムである．2 M バイトのデータを転送して，帯域幅を実測する．また，8 バイトのデータ転送に要した時間を計測し，転送遅延とする．データ転送手法として PingPong 転送と Ring 転送が用意されており，さらに Ring 転送には `MPI_COMM_WORLD` でのランクの順番に並べられた Natural-Ordered Ring と乱数により並べられた Random-Ordered Ring がある．測定項目として 10 項目用意されているが，HPCC プロジェクトのサイト[5] では登録データの転送遅延に関する 3 項目（PingPong Min.，Random Ring および Natural Ring）と帯域幅に関する 2 項目（PingPong Max. と Random Ring）の結果のみ表示される．

3. HPCC を用いた性能評価

3.1 評価環境

HPCC ベンチマークを用いてベクトル型スーパコンピュータの性能評価を行い，LINPACK ベンチマーク単独での評価との相違を明確にする．評価実験には，東北大学情報シナジーセンターに設置されたベクトル型スーパコンピュータ NEC SX-7（以下，単に SX-7 とする）を用いる[7]．SX-7 システムは，256 G バイトのメモリを共有する 32 台の CPU を搭載した SMP ノードを構成要素としている．CPU 単体あたりの理論最大演算性能は 8.83 Gflop/s であり，SMP ノードあたりに換算すると 282.56 Gflop/s の理論最大演算性能を有する．CPU あたりのメモリ帯域幅の理論最大値は 35.3 GB/s である．

本論文では，HPCC ベンチマーク用いて SX-7 の

SMP ノードの実効性能を評価する．SMP ノードあたり 8 台の CPU の NEC SX-6[8] や，SMP ノードあたり 4 台の CPU の Cray X1[9] と比較すると，32 台の CPU による大きな SMP 共有並列を実現できることが SX-7 の長所の 1 つである．HPCC ベンチマークでは MPI 通信の性能を評価するために必ず複数の MPI プロセスが必要なため，本論文では SMP ノードの利用形態として以下の 2 つを考え，最大 32 台もの大規模 SMP 共有を利用できる SX-7 の利点を評価する．

(1) MPI 並列のみ
32 台のすべての CPU に MPI プロセスを 1 つずつ割り当て，MPI による 32 並列計算を実現する．

(2) SMP+MPI 並列
32 台の CPU を 16 台ずつの 2 つのグループに分け，各グループに対して MPI プロセスを 1 つずつ割り当てる．MPI プロセスはそれ自身がコンパイラによって自動並列化されており，高度に SMP 並列化された BLAS ライブラリもリンクされている．このため，各プロセスは 16 CPU による SMP 共有並列処理により効率良く実行される．

HPCC ベンチマークのコンパイルには以下のオプションやライブラリを用いた．

- BLAS：NEC MathKeisan 1.3.0
- MPI：NEC MPI/SX 6.7.8
- コンパイラ：NEC C++/SX Rev.0.61
- コンパイルオプション
  -Popenmp -Pauto -size_t64 -xint -Caopt
  -O fullmsg -pvctl,fullmsg,vwork=stack
  -Orestrict=arg -pi,auto

3.2 結果と考察

本論文で述べる SX-7 の評価結果はすでに HPCC プロジェクトのサイトに登録されており，それぞれの性能測度から他の HPC システムとの性能比較が可能である☆．MPI 並列の評価に用いた HPCC ベンチマークの設定ファイル `hpccinf.txt` を**表 1** に示す．SMP+MPI 並列の場合には $Q = 2$ と変更し，その他は表 1 と同じパラメータを用いて実行した．

以下，各項目での評価結果について簡単に述べ，3.2.8 項でそれらの結果について考察する．

3.2.1 HPL

まず，HPL のパラメータ設定について事前に検討した結果を述べる．

☆ 2004 年 12 月現在，IBM，Sun，SGI 等，45 のシステムの結果が公開されている．

表 1 HPCC ベンチマークのパラメータ設定
Table 1 Parameters for the HPCC benchmark.

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
8            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
61000        Ns
1            # of NBs
64           NBs
1            PMAP process mapping (0=Row-,1=Column-major)
1            # of process grids (P x Q)
1            Ps
32           Qs
16.0         threshold
1            # of panel fact
2            PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
44           NBMINs (>= 1)
1            # of panels in recursion
3            NDIVs
1            # of recursive panel fact.
2            RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
1            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
16           memory alignment in double (> 0)
##### This line (no. 32) is ignored (it serves as a separator). ######
0                        Number of additional problem sizes for PTRANS
1200 10000 30000          values of N
2                         number of additional blocking sizes for PTRANS
255  471                          values of NB
```

NB は，HPL による評価結果に大きな影響を与える重要なパラメータである．この NB を小さくすることで各プロセス間の負荷の偏りが小さくなることが期待できるが，通信回数が多くなるために性能劣化の要因となる．一方，大きくすることによって負荷の偏りが大きくなるだけでなく，ブロック化処理のオーバヘッドも発生して性能劣化の要因となる．予備実験により，本実験条件下では $NB = 64$ が最適であることが分かった．

プロセス格子 (P, Q) については，$P > 1$ の場合，行と列の両方向に分割されることによってベクトル長が $1/P$ だけ短くなる．したがって，ベクトル演算器の処理の効率の観点から考えると，ベクトル長をできる限り長くするために $P = 1$ としてプロセスを 1 次元格子に設定し，ブロックを短冊状に配置することが望ましい．一方，通信時間に着目すると $P \simeq Q$ かつ $P \leq Q$ の 2 次元格子に設定することが望ましい[10]．しかし，本実験の機器構成ではクロスバスイッチで接続された共有メモリを介した高速な通信が可能であるため，後者の影響はほとんど顕在化せず，前者による処理の高速化の効果が大きい．このため本実験では，列優先のプロセスマッピングで $P = 1$ と設定した．したがって，データ分散化は NB 格子幅の Q プロセスのブロックサイクリックとなる．

$NDIV$ は再帰的にブロック化するときの段数であ

表 2 HPL による性能評価結果
Table 2 The results on the HPL test.

	Tflop/s
SMP+MPI	0.2174
MPI	0.2553

表 3 DGEMM による性能評価結果
Table 3 The results on the DGEMM test.

	Gflop/s	
	SN	EP
SMP+MPI	140.974	140.636
MPI(1)	4.736	8.239
MPI(2)	8.656	8.343

る．段数が大きい場合には小さな部分もブロック化されるため，ブロック化のオーバヘッドで性能が劣化する．一方，段数が小さい場合にはブロックを作ること自体の処理が大きくなるため，処理全体での性能が劣化する．ただし，予備実験を行ったところ，2 から 5 段での N が小さなサイズの場合，パラメータによる有意な差はなかった．このため本実験では，段数を 3 段として評価した．同様に，本実験環境下では LU 分解のアルゴリズムや $BCAST$，$NBMIN$ も評価結果にほとんど影響を与えないことが予備実験により明らかになった．

本実験では，以上の検討結果をふまえて HPCC ベンチマークのパラメータ設定を行った（表 1）．

MPI 並列および SMP+MPI 並列の浮動小数点演算性能を評価した結果を**表 2** に示す．表 2 より，MPI 並列の場合は理論性能比 90.3%の高い実効効率を達成できていることが分かる．また，SMP+MPI 並列の場合は 76.9%の実効効率を達成している．コードの改変が認められていないベースライン実行では，コンパイラの自動最適化の性能も評価に大きく影響する．本実験環境では，コンパイラの判断によりコードの一部が SMP 並列化されないため，SMP+MPI 並列と比較して MPI 並列の方が優れた結果となった．HPCC のサイトに登録されている他のスカラ並列型スーパコンピュータの実効効率と比較すると，SX-7 は SMP+MPI 並列の場合でも依然として高い実効効率を維持できている．

3.2.2 DGEMM

DGEMM の実行時間の内訳を調査した結果，その大半は BLAS 内のサブルーチンによって費やされていることが明らかになった．すなわち，CPU や SMP ノード単体の演算性能に加えて，利用可能な BLAS の品質が評価結果に大きく影響する．

DGEMM による評価結果を**表 3** に示す．表から分かるように，16 CPU を 1 つの SMP ノードとして評価する SMP+MPI 並列の場合，ノード単体性能としてはきわめて高い評価を得ることができる．SX-7 ではさらに 32 CPU までを SMP ノードとして扱うことができるため，SMP ノード単体の演算性能が求められる科学技術計算に関しては非常に高い実効性能を期待できることが分かった．また，今回採用した BLAS ライブラリは SX-7 用に高度に最適化されており，理論性能比で考えると表中に MPI(1) で示された MPI 並列は 93.3%，SMP+MPI 並列は 99%の非常に高い実効効率を達成できた．

ただし，MPI(1) 並列では評価方法 SN での評価の方が EP よりも低い値となっている．これは SN において，1 プロセスで DGEMM 計算を行っている以外は，待ちの状態になっており，その同期待ちの影響で性能が劣化しているからである．一方，EP ではすべてのプロセスが DGEMM の計算を行っており，メモリ帯域を使いきることが性能劣化の主要因となる．本実験では SN の方がより性能劣化の影響が現れた結果になった．

同期のオーバヘッドを小さくするために同期待ちのチェックをより緩く監視するパラメータ SUSPENDCOUNT を 10000 に変更した結果を，表 3 中の MPI(2) に示す．表 3 において，MPI(1) は MPISUSPEND を OFF にしたときの結果であり，HPCC のサイトにはこれらの結果が登録されている．また，MPI(2) は SUSPENDCOUNT を 10000 に設定して計測した結果であり，MPI 通信のパラメータを適切に設定することによってさらに性能が向上することが明らかになった．

3.2.3 STREAM

STREAM のコードには OpenMP による指示行が含まれているため，コンパイラによる自動 SMP 並列化が有効に機能する．また，今回用いたコンパイラでは，高いベクトル化率も達成可能であった．

STREAM による評価方法 SN での評価結果を**表 4** に示す．また，評価方法 EP での評価結果を**表 5** に示す．一般的に，ベクトルロードストアユニットを有するベクトル型スーパコンピュータは高いメモリ帯域幅を達成可能である．さらに，SMP+MPI 並列の場合には SX-7 の SMP 共有並列を最大限に活かした結果となっており，MPI 並列と比較しても非常に高いメモリアクセス性能を実現している．

この STREAM においても，MPI 通信のパラメータを調整することによって大幅な性能向上が得られる．表 4 および 5 中の MPI(2) は，DGEMM のときと同

表 4 STREAM による評価方法 SN での性能評価結果
Table 4 The results on the SN STREAM test.

	GB/s			
	Copy	Scale	Add	Triad
SMP+MPI	537.486	379.734	437.240	556.609
MPI(1)	26.452	26.191	28.852	26.008
MPI(2)	35.104	35.037	35.359	35.359

表 5 STREAM による評価方法 EP での性能評価結果
Table 5 The results on the EP STREAM test.

	GB/s			
	Copy	Scale	Add	Triad
SMP+MPI	389.791	348.593	428.084	492.161
MPI(1)	26.456	26.129	28.831	26.154
MPI(2)	26.441	26.117	28.884	26.336

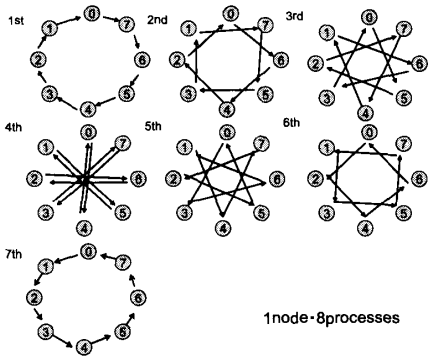


図 1 PTRANS でのデータ送受信手順
Fig. 1 The data transfer scheme for the PTRANS test.

様にパラメータ SUSPENDCOUNT を 10000 に設定した場合の結果である．表 4 から分かるとおり，通信パラメータの適切な設定により，実効メモリ帯域幅を平均 31.2% 向上させることが可能であった．

3.2.4 PTRANS

PTRANS の実行時間の内訳を調査すると，`MPI_Send()` と `MPI_Recv()` による 1 対 1 通信に多くの時間が費やされている．PTRANS では，**図 1** に示す手順で 1 対 1 通信を繰り返すことで全対全通信を行い，行列の転置を実現している．この図は 8 プロセスの場合の手順であり，データ送受信は次のようになる．まず，各プロセスは（自分のランク -1）のランクを持つプロセスから始まって，ランクが徐々に減少する順番で送信相手を変化させる．逆に受信処理は（自分のランク $+1$）のランクを持つプロセスから始まって，ランクが徐々に増加する順番で受信相手を変化させる．

PTRANS による評価結果を**表 6** に示す．PTRANS の実行時間はパラメータ設定によって大きく変化する．特にパラメータ NB がデータ転送とその前後の処理で使われ，そのサイズでのメモリコピーが発生する．また，通信以外の処理では NB の値はベクトル長となっており，この値によってループのストライドが変化してバンク競合が発生する．**表 7** に，問題サイズを $N = 10000$ に固定し，プロセス数 4 として NB を変化させて測定区間内の性能を評価した結果を示す．この結果から，通信時間およびバンク競合時間ともに NB に依存していることが分かる．バンク競合を抑えるために NB を奇数とすることが考えられるが，その場合には通信性能が低下するために必ずしも良い結果とはならなかった．

表 6 PTRANS による性能評価結果
Table 6 The results on the PTRANS test.

	GB/s
SMP+MPI	16.340
MPI	20.546

表 7 NB の PTRANS への影響
Table 7 The effects of NB on the PTRANS performance.

GB/s	comm.(sec)	bank(sec)	NB
7.423	0.024	0.0062	90
7.871	0.023	0.0066	100
7.035	0.026	0.0111	104
8.543	0.022	0.0064	105
7.091	0.026	0.0112	106
6.496	0.029	0.0118	110
6.424	0.029	0.0119	120
5.972	0.031	0.0131	200

表 8 RandomAccess による性能評価結果
Table 8 The results on the RandomAccess test.

	GUP/s		
	SN	EP	G
SMP+MPI	0.23450	0.23262	0.000178
MPI	0.06094	0.20800	0.000964

3.2.5 RandomAccess

RandomAccess による評価結果を**表 8** に示す．評価方法 SN と EP の場合，RandomAccess はノード内のメモリへのランダムアクセス性能を評価している．3.2.3 項でも述べたとおり，一般的にベクトル型スーパコンピュータは高いメモリ帯域幅を有するため，高い評価が期待できる．逆にスカラ型プロセッサの場合にはキャッシュミス時の遅延が顕著に現れるため，低い評価になることが予想される．評価方法 G の場合には，ネットワーク全体の総合性能が評価される．この評価では，ノード数が大きい大規模なシステムの方が有利である．

3.2.6 FFTE

FFTE による評価結果を**表 9** に示す．FFTE では，

表 9 FFTE による性能評価結果
Table 9 The results on the FFTE test.

	Gflop/s		
	SN	EP	G
SMP+MPI	1.9084	1.5698	1.337
MPI	0.4244	0.6808	11.288

表 10 帯域幅の性能評価結果
Table 10 The results on the communication bandwidth test.

	Bandwith [GB/s]				
	P.Min.	P.Ave.	P.Max.	R.Ring	N.Ring
SMP+MPI	10.9	10.9	10.9	8.1	8.1
MPI	3.1	3.3	10.4	5.0	5.2

表 11 転送遅延の性能評価結果
Table 11 The results on the latency test.

	Latency [usec]				
	P.Min.	P.Ave.	P.Max.	R.Ring	N.Ring
SMP+MPI	3.6	3.6	3.7	4.9	4.9
MPI	3.9	7.1	9.7	14.2	14.4

そのコードの中の **L2SIZE** という定義を，HPC システムの構成に合わせて適切な値に調整する必要がある．しかしベースライン実行ではコードの改変が許されていないため，その値を調整できない．その結果，FFTE による性能評価では低い評価しか得られなかった．HPCC サイトに登録されている他の HPC システムについても，同様の理由で低い評価となっていると考えられる．

3.2.7 Communication Bandwidth and Latency

データ転送の帯域幅と遅延を計測した結果を，**表 10** および **表 11** に示す．本実験では SX-7 の SMP ノード単体を用いており，ノード内通信を評価していることから，これらの評価項目では比較的高い評価となった．複数の SMP ノード間のデータ転送も含めた SX-7 のデータ転送性能については，今後評価する予定である．

3.2.8 考　　察

SX-7 は，同じベクトル型スーパコンピュータと比較しても，特にメモリ帯域幅に関する項目でその性能がきわめて高く評価されている．また，2004 月 12 月現在登録されている HPC システムの中では，SX-7 が 28 の評価項目中 16 の項目において最も高い評価となっている[5]．

プラズマ物理学分野で用いられる LBMHD では，データのワードアクセスごとに約 1.5 回の浮動小数点演算が行われる[3]．このようにメモリアクセス頻度に対して計算負荷が低い場合，ピーク演算性能に対して低いメモリ帯域幅の HPC システムでは演算器へのデータ供給が間に合わなくなるため，高い実効性能の維持は困難である．しかし，SX-7 の場合，CPU 単体あたりの理論最大演算性能（Gflop/s）と理論メモリ帯域幅（GB/s）の比は 1 : 4 であり，DGEMM と STREAM の計測結果から，実効性能でも理論性能と同等の比を達成できていることが分かる．同じベクトル型スーパコンピュータの Cray X1 では，それらの理論性能の比は 1 : 2 である[3]．スカラ型スーパコンピュータの場合には，高いメモリ帯域幅を持つ SGI Altix ですらその理論性能どうしの比で 1 : 1 となっており[3]，実測値ではメモリアクセス性能が演算性能をつねに下回る結果となっている[5]．このことから，SX-7 が他の HPC システムと比較して高いデータ供給能力を有しており，LBMHD のようにデータアクセスあたりの演算負荷が低い科学技術計算に対しても高い実効性能を維持できることが示唆された．

従来の単一指標である LINPACK ベンチマークでは，上記に示した演算性能とメモリ帯域幅との実測値の比のような，性能のバランスの良し悪しを評価できなかった．このことから，複合的な評価である HPCC ベンチマークは高性能，高効率なスーパコンピュータの研究開発に必要不可欠な性能評価ツールになっていくと思われる．

4. おわりに

本論文では，次世代の標準ベンチマークとして期待されている HPCC ベンチマークを用いて，ベクトル型スーパコンピュータである NEC SX-7 の性能を評価した．HPCC ベンチマークは，LINPACK ベンチマークのような演算に限定された測定指標と異なり，メモリ帯域幅性能，ネットワーク性能，基本カーネルが含まれており，HPC システムの性能を複数の観点から多角的に評価できる．

評価実験の結果，実用的な科学技術計算において高い実効性能を達成するためには重要であるにもかかわらず LINPACK ベンチマークでは評価できなかった性能差についても，HPCC ベンチマークでは結果に顕著な差として現れることが分かった．メモリアクセス性能の観点から，SX-7 のようなベクトル並列型スーパコンピュータはスカラ並列型スーパコンピュータの性能を凌駕している．また，特に SX-7 の SMP 並列の場合には，最大 32 個の CPU 間での大きなメモリ共有並列が実現可能であり，MPI 等と比較してデータ共有のためのオーバヘッドを大幅に軽減させることが可能である．LINPACK ベンチマークのようにキャッ

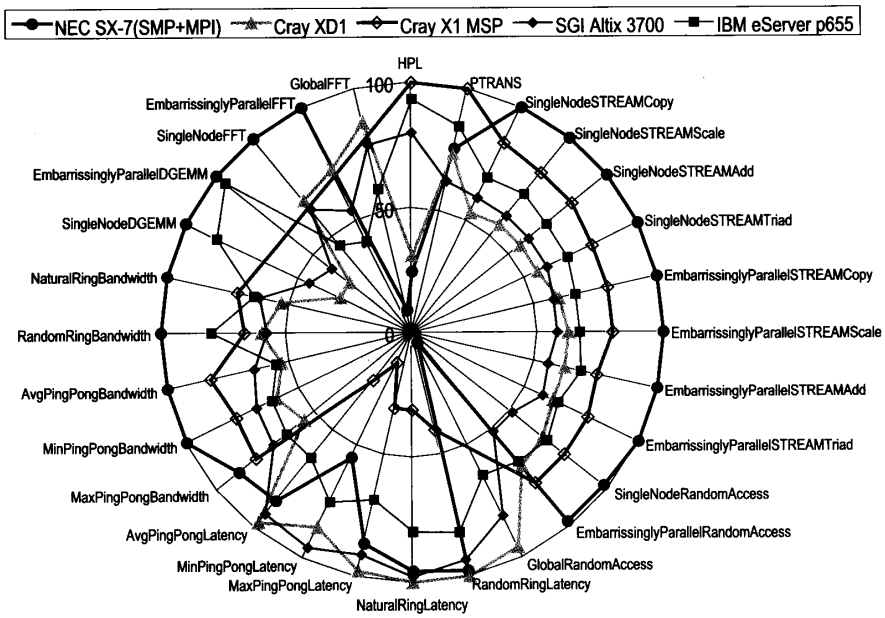


図 2 システム性能特性のレーダーチャート
Fig. 2 The rader chart of the overall results.

シュが極端に有効に機能するアプリケーションを除いて，これらの性能差は一般的な科学技術計算の実効性能に著しく影響することから，ベクトル並列型スーパコンピュータの高い潜在能力が明らかになった．大学を中心とする学術機関の研究者のベクトル並列型スーパコンピュータに対する需要が依然として高いことが，この HPCC ベンチマークによる評価の妥当性を裏付けている．利用者プログラムに対する高速化を中心とした利用技術支援を積極的に行った結果，現在東北大学情報シナジーセンターで 24 時間運転で運用中の 8 ノード（240 CPU）の SX-7 システムは，年平均 85%以上の CPU 利用率（毎年 11〜2 月は 95%以上，実行待ちジョブ毎日 80 件程度）で活用されている．

HPCC ベンチマークを用いることで，演算性能とメモリ帯域幅の比のようなシステム性能のバランスや，HPC システムが得意とする計算等を検討できるようになる．今後この HPCC ベンチマークがより普及することで，個々の科学技術計算について高い実効性能を期待できる HPC システムを選択可能になり，高価な HPC システムを適材適所でより有効に活用できるようになると考えられる．

ただし，複数の評価指標の集合体であり，総合指標が存在しない HPCC ベンチマークでは，HPC システムの性能の一義的な解釈が困難である．HPCC ベンチマークの各項目での評価内容を十分に理解し，すべての計測結果を解析することによって，評価結果が暗示するシステム性能の特性を読みとることが求められる．

システム性能の特性を読みとるための手段の 1 つとして，レーダチャートで各評価結果を図示することが考えられる．**図 2** は，HPCC のサイトに登録されている結果☆の中で，SX-7 の SMP+MPI 並列に対する評価結果の順位を正規化し，各軸の値としたレーダチャートである．一番外側の 100%は，その項目において SX-7 の SMP+MPI 並列が第 1 位であることを示している．本実験では SX-7 の単独ノードを用いたため，CPU 数はわずか 32 台であり，CPU 数に結果が依存する HPL や PTRANS では低い評価結果となっている．一方で，CPU 数に依存しない，メモリ帯域幅（STREAM 等）やノード単体性能（DGEMM 等）を評価する項目の大半では非常に高い評価が得られている．HPCC サイトの登録数が少ないためこの解析が必ずしも適切であるとはいえないが，他の HPC システムでも同様のレーダチャートを作成することで，その性能の相対的な特性をある程度把握できると考えられる．同様に，Lazou はオリンピック大会のメダル方式を提唱し，HPCC ベンチマークの 8 評価項目について 1 位，2 位，および 3 位のシステムをそれぞれ金メダル，銀メダルおよび銅メダルとして評価し，獲得メダル数からベクトル並列型スーパコンピュータの実効性能の高さを述べている[11]．今後，多様な構成での各種 HPC システムの評価結果が HPCC のサイトに数多く登録されることを期待するとともに，より適切で効果的な解析手法の確立が今後の課題としてあげ

☆ 2004 年 12 月現在．

られる．

謝辞 本実験に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，HPC 販売推進本部の小林義昭氏，株式会社 NEC 情報システムズの小林一夫氏，浅見暁氏，後藤記一氏，NEC ソフト株式会社の深田大輔氏，および東北大学情報シナジーセンターの岡部公起氏，伊藤英一氏をはじめとする関係者各位に深謝します．

参 考 文 献

1) Meuer, H., et al.: Top500 Project Page. http://www.top500.org
2) Dongarra, J.: Performance of Various Computers Using Standard Linear Equations Software, Computer Science Technical Report CS-89-85, University of Tennessee (1989). an updated version at http://www.netlib.org.
3) Oliker, L., et al.: Scientific Computations on Modern Parallel Vector Systems, *IEEE/ACM SC2004 Conference*, Pittsburgh, PA (2004).
4) The High-End Computing Revitalization Task Force: Federal Plan for High-End Computing, Technical report (2004).
5) Dongarra, J., et al.: HPC Challenge Benchmark Project Page. http://icl.cs.utk.edu/hpcc
6) Dongarra, J., et al.：コンピュータによる連立一次方程式の解法—ベクトル計算機と並列計算機，丸善 (1993). 小国 力（訳）.
7) 東北大学情報シナジーセンター：*SENAC*, Vol.35, No.3 (2002).
8) 日本電気株式会社：NEC 技報，Vol.55, No.9 (2002).
9) Cray Inc.: Cray X1 Supercomputer product page. http://www.cray.com/products/x1/index.html
10) 笹生 健，松岡 聡：HPL のパラメータチューニングの解析，並列/分散/協調処理に関する湯布院サマー・ワークショップ (SWoPP2002) (2002).
11) Lazou, C.: HPC Benchmarks: Going for Gold in a Computer Olympiad?, *HPC wire*, Vol.14, No.3 (2005).

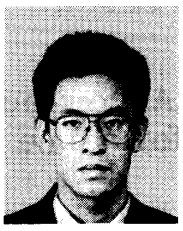
(平成 17 年 1 月 24 日受付)
(平成 17 年 4 月 18 日採録)

滝沢 寛之（正会員）

平成 11 年東北大学大学院情報科学研究科博士課程修了．同年 10 月新潟大学助手．平成 15 年東北大学助手．平成 16 年東北大学講師．高性能計算システムやその応用としてデータクラスタリングやニューラルネットワーク等の研究に従事．博士（情報科学）．IEEE，電子情報通信学会各会員．

小久保達信

平成 2 年東北大学大学院理学研究科博士課程後期 3 年の課程修了（化学第二専攻）．同年日本電気株式会社入社．数学ライブラリの開発に従事．特に乱数，固有値，FFT 等の開発を行う．現在，同社 HPC 販売推進本部 HPC ソリューションマネージャー．理学博士．



片海 健亮

平成 10 年千葉大学大学院自然科学研究科博士課程前期 2 年の課程修了（生命・地球科学専攻）．同年日本電気ソフトウェア株式会社（現 NEC ソフト株式会社）入社．HPC 分野のプログラム高速化チューニングに従事．現在，同社製造ソリューション事業部所属．修士（理学）．

小林 広明（正会員）

昭和 63 年東北大学大学院工学研究科博士課程修了．同年東北大学工学部助手．平成 3 年東北大学講師．平成 5 年東北大学助教授．平成 13 年 10 月東北大学教授（情報シナジーセンター副センター長兼任）．コンピュータアーキテクチャ，並列処理システムとその応用に関する研究に従事．工学博士．IEEE Senior member，ACM，電子情報通信学会各会員．