The Science and Engineering Doshisha University, Vol.44, No.3 October 2003

# Evaluaion of Linux-based High Performance Computing Cluster using LINPACK Benchmark

Tomoyuki Hiroyasu* Mitsunori Miki* and Yoshikazu Kugii**

(Received July 29, 2003)

Due to recent improve performance of widely available personal computers, PC cluster systems attract attention than high performance vector supercomputer. PC cluster systems consist of many PCs connected by network and are used for parallel or distributed computations. HPC cluster systems are attractive to the scientist and engineers who are need of high-performance computations. At Intelligent Systems Design Laboratory, we install an HPC cluster system called Xenia which is a 64-node cluster running Linux Operating System. We evaluated the cluster system using HPL, which is an implementation of LINPACK benchmark, which is widely used for evaluating performance and is introduced in the TOP500 Supercomputer Sites. It is possible to get better performances through fine-tuning of parameters, but it is difficult to determine the best, because HPL has a wide range of parameter selections. In this paper, we present the combination of parameters we used when Xenia got the 195th place on 20th Top500 Supercomputer Sites, and show how to tune the parameters. We also mention the knowledges obtained through these experiments about parameter tuning for better HPL performance.

**Key words** : PC cluster, TOP500 Supercomputer Sites, LINPACK Benchmark, Linux

**キーワード** : PC クラスタ，TOP500 スーパーコンピュータサイト，リンパックベンチマーク，Linux

# LINPACK Benchmarkによるハイパフォーマンス Linuxクラスタの性能評価

廣 安 知 之・三 木 光 範・釘 井 睦 和

## 1. はじめに

近年，超並列計算機に代わってパーソナルコンピュータやワークステーションなど一般に利用されているコンピュータをネットワークで繋ぎ，1つの計算機として利用できる PC クラスタシステム[1–4](以下クラスタ)が注目されている．中には，Myricom 社[5] の Myrinet のようにスループットが 1Gbps を超え，従来の超並列計算機と同等の低レイテンシを実現するネットワークを利用して，大規模なクラスタも構築されている．クラスタ最大の利点は，同程度の性能を持つスーパーコンピュータと比較して非常に高いコストパフォーマンスを有することである．そのため，一部の研究機関や企業しか所有することができなかった高性能計算サーバを研究室やグループ単位でも所有することが可能となった．世界の高性能コンピュータの上位 500 位をリストアップしている Top500 Supercomputer Sites[6] においても，クラスタシステムはスーパーコンピュー

* Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6930, Fax:+81-774-65-6796, E-mail:mmiki@mail.doshisha.ac.jp , tomo@is.doshisha.ac.jp
** Graduate Student, Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6716, Fax:+81-774-65-6716, E-mail:ykugii@mikilab.doshisha.ac.jp

タにひけをとらない性能を見せている．Top500 にランクインすることは，所有機関にとって高性能の計算サーバを有することを世界にアピールする最大の機会である．そのため，ハイパフォーマンスコンピュータのユーザやベンダ，大規模計算機センターにとって大きな興味の対象であり，Top500 はこの種のリストの中で最大のものとなっている．

我々は TOP500 に挑戦できるハイエンドクラスタ Xenia Cluster System を構築した．本稿ではこの Xenia の性能評価を LINPACK Benchmark の実装の 1 つである HPL[7] (High-Performance LINPACK Benchmark) を用いて行い，そのパラメータ設定やコンパイラ比較，通信を行うネットワークのインターコネクタ比較を行った．これらを通して，高性能クラスタの構築，および HPL のパラメータ設定の方針に関する知見を得ることができた．

## 2. PC クラスタ

PC クラスタは，単一で稼働するコンピュータの集まりで，1 つの計算資源として使用可能な並列もしくは分散システムである．クラスタを構成する各ノードはコンピュータの最小構成である CPU，メモリ，OS などを有している．一般的に PC クラスタという表現は次のように分類することができる．

- HPC(High Performance Computing) クラスタ

  1990 年代中頃からパーソナルコンピュータの性能向上を背景に，NASA の Beowulf プロジェクト[8–10] が提唱するクラスタシステムが発展してきた．主に科学技術計算で利用される並列アプリケーションの実行を目的としたクラスタである．Beowulf クラスタはスーパーコンピュータと同等の処理能力を低コストで実現することが可能である．Beowulf クラスタは典型的な実例である．Beowulf クラスタは既存の低価格なハードウェアとオープンソースな Linux や FreeBSD などの OS を用いて構築し，並列処理プログラミング (MPI[11,12]，PVM[13] 等) や並列アプリケーションを実行できる．

  クラスタリングソフトウェア SCore[14,15] を用いた SCore クラスタも Beowulf プロジェクトとほぼ同時期に RWCP(RealWorld Computing Partnership：新情報処理開発機構) において開発が始まった．Beowulf と異なり，SCore は当初からクラスタコンピューティングのためのシステムソフトウェア環境の提供を目的にしている．現在，SCore は Linux ベースのオープンソースとして PC クラスタコンソーシアムにより提供されている．クラスタとしての特徴は，独自の PM 通信機構による高速化，SCore-D による TSS(Time Sharing System) 環境，ジョブスケジューリングの機能が挙げられる．

- HA(High Availability) クラスタ

  ミッション・クリティカルなアプリケーションを実行するためのクラスタであり，冗長化構成により障害発生時には切替え作業で対処する．フォールト・トレラント・コンピュータの適用分野でシステムをより低コストに実現できる．システムが停止しては困るようなデータベースなどの基幹業務をはじめ，アプリケーションサーバ，ファイルサーバのほか近年ではダウンタイムが致命傷となるインターネット上のサーバ (ファイアウォールサーバやメールサーバなど) に利用されている．

## 3. システム構成の検討

本研究では，高性能な PC クラスタを構築する．構築したクラスタの性能評価には LINPACK ベンチマークを用い，TOP500 のランクインを目標とした．LINPACK ベンチマークは実行時に不確実性の少ない数値計算アルゴリズムであるため，それをもとにしてハードウェアからある程度の性能予測が可能である．
LINPACK ベンチマークの実装の一つである分散メモリ型並列計算機ベンチマーク，HPL(High-Performance LINPACK Benchmark) の演算量は式 (1) で与えられることが既知である．実行性能値からインターコネクタの違いにより理論ピーク性能値，メモリなどを決定することができる．N は HPL における問題サイズ，T は演算時間 [秒] である．HPL については 6 節で詳しく述べる．N に関しては式 (2) より設定することが可能であり，メモリ量の検討を行うことが可能である．

$$Performance[GFlops] = (\frac{2}{3}N^3 + \frac{3}{2}N^2) \times \frac{1}{T} \times 10^{-9} \quad (1)$$

$$N = \sqrt{(\text{計算機の全メモリ容量 } [byte]) \times x[\%]/8} \quad (2)$$

Table 1. TOP500 リスト (2002 年 6 月) における Myrinet2000 を用いたクラスタ (一部).

| ランク | Cluster 名 | Rmax [GFlops] | Rpeak [GFlops] | Ratio [%] | CPU の種類 | CPU 数 |
|---|---|---|---|---|---|---|
| 35 | HELICS | 825 | 1430 | 57.69 | Athlon 1.4GHz | 512 |
| 47 | Presto III | 716.1 | 1536 | 46.62 | Athlon 1.2GHz | 480 |
| 58 | Netfinity Cluster | 594 | 1024 | 58.01 | PentiumIII 1GHz | 1024 |
| 247 | CLIC | 221.6 | 424 | 52.26 | PentiumIII 800MHz | 530 |

Table 2. TOP500 リスト (2002 年 6 月) における Ethernet を用いたクラスタ (一部).

| ランク | Cluster 名 | Rmax [GFlops] | Rpeak [GFlops] | Ratio [%] | CPU の種類 | CPU 数 |
|---|---|---|---|---|---|---|
| 126 | Netfinity Cluster | 366 | 1790 | 20.45 | PentiumIII 1.4GHz | 1280 |
| 127 | Netfinity Cluster | 366 | 1144 | 31.99 | PentiumIII 1/1.26GHz | 1024 |
| 375 | Netfinity Cluster | 177 | 408 | 43.38 | PentiumIII 1GHz | 408 |
| 381 | Netfinity Cluster | 170 | 477 | 35.64 | PentiumIII 933MHz | 512 |

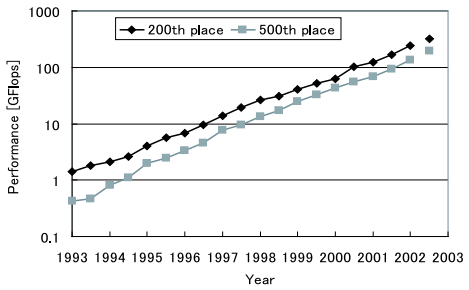


Fig. 1. TOP500 における 200 位，500 位の推移.

Fig. 1 に示すように，近年の TOP500 の傾向から，20th リスト (2002 年 11 月) ランクインに必要な最低限の Rmax(LINPACK 実行性能値) を 200GFlops 以上のシステムとして，200 位以内のランクインに必要な最低限の Rmax を 320GFlops 以上のシステムとして必要スペックの検討を行った.

Table 1 より，インターコネクタに Myrinet-2000 を使用する場合は Rpeak(理論ピーク性能値) の 50 %が Rmax になると仮定できる．Rmax が 200GFlops 以上のシステムを構築したい場合は，Rpeak が 400GFlops 以上のシステムが必要となる．また Table 2 より，インターコネクタに Ethernet を使用する場合は，Rpeak の 30 %が Rmax になると仮定できる．同様に Rmax が 200GFlops 以上のシステムを構築したい場合は，Rpeak が 700GFlops 以上のシステムが必要となる．Myrinet-2000 と Ethernet のそれぞれの場合に必要なプロセッサ数は Table 3，Table 4 のようになる.

Table 3. Rmax200GFlops 以上のシステム.

| | Myrinet-2000 | Ethernet |
|---|---|---|
| Xeon 2GHz | 100PE | 175PE |
| PentiumIII 1GHz | 400PE | 700PE |

Table 4. Rmax320GFlops 以上のシステム.

| | Myrinet-2000 | Ethernet |
|---|---|---|
| Xeon 2GHz | 160PE | 267PE |
| PentiumIII 1GHz | 640PE | 1067PE |

Table 3，Table 4 における Myrinet，Ethernet どちらの場合も式 (1)，式 (2) より，十分な問題サイズを与えられるようにシステム全体のメモリ量は 60GB 以上は必要であると考えられる.

## 4. Xenia Cluster System

前節で述べたように，目標となる LINPACK 実行性能値からシステムの構成を決定することができる．新たに導入したクラスタ (Fig. 2) では，前節のように性能を見積もり，必要最低限のスペックで導入を行った．導入したクラスタは Xenia と呼び，以降 Xenia と記す．IBM 社のサーバ用ワークステーション IntelliStation M Pro 6850 60J を 64 台用いたクラスタであり，インターコネクタに Myrinet-2000 と Ethernet を使用している．主なハードウェア構成，ネットワーク構成を Table 5 に示す.

Fig. 2.　Xenia クラスタ.

Table 5.　Xenia のハードウェア構成.

| | |
|---|---|
| ノード数 | 64 |
| CPU | Intel Xeon 2.4GHz × 2 |
| メモリ | 1GB × 64(計 64GB) |
| OS | Red Hat Linux 7.3 |
| 通信ライブラリ | MPICH-1.2.4 |
| 通信プロトコル | GM，TCP/IP |
| 通信媒体 | Myrinet-2000，FastEthernet |

Table 6.　Xeon プロセッサ概要.

| | |
|---|---|
| L1 キャッシュ | 8KB |
| L2 キャッシュ | 512KB |
| クロック当たりの命令発行数 | 6 |
| 整数パイプライン | 4 |
| 浮動小数点パイプライン | 2 |
| システムバス速度 | 400MHz |
| 3D 拡張命令 | SSE2 |

### 4.1　プロセッサ

Xenia では Intel Xeon プロセッサを採用している．Intel Xeon プロセッサの特徴を Table 6 に示す．Xeon プロセッサは Pentium4 プロセッサとアーキテクチャが類似しているが，デュアルプロセッサに対応しているという点で異なる．

Xeon プロセッサでは 8KB のデータキャッシュ以外に実行トレースキャッシュを備え， デコード済みのマイクロオペレーションをプログラムの実行順に最大 12KB 格納することができ，高速な演算が実現する．SSE2 (ストリーミング SIMD 拡張命令 2) 命令セットを用いることにより，クロック周波数の 2 倍の浮動小数点演算が可能となる．このことから Xenia のピーク性能値は 614.4GFlops となる．

### 4.2　コンパイラ

Xenia には gcc，Intel，pgi の 3 種類のコンパイラを用意した．それぞれの特徴について述べる．

- GCC[16)]
  GCC(GNU Compiler Collection) は C，C++，Objective C，Fortran などで書かれたプログラムをコンパイルすることが可能である．GNU プロジェクトに使用されているため，UNIX 系 OS では現在最も広く普及している．
- Intel[17)]
  Intel C++/Fortran Compiler は Intel が開発している C++/Fortran 用のコンパイラである．特徴としては，Pentium 系の CPU に最適化したバイナリを生成することがあげられる．
- PGI[18)]
  PGI Compiler は，Portland Group により開発されているコンパイラで，HPC(High Performance Computing) において，最適化能力が優れていると評されている．

### 4.3　ネットワーク

Xenia はメインのインターコネクタとして Myricom 社の Myrinet-2000，補助として Ethernet(100BASE-T) で接続されている．Myrinet-2000 の特徴を以下に述べる．

- 高速通信 (データレート 2Gbps(双方向では 4Gbps)，レイテンシ 9 μ sec 以下)
- ゼロコピー通信や TCP/IP，UDP を利用可能
- Myrinet スイッチによりあらゆるネットワークトポロジーを実現
- 高い信頼性 (Myrinet スイッチの MTBF は 100 万時間を超え，Myrinet インターフェースの MTBF は数 100 万時間を超える)
- Ethernet に比べて非常に高価

## 5.　LINPACK Benchmark

LINPACK は米国 Tennessee 大学の J.Dongarra 博士らによって開発された *LU* 分解に基づく連立一次方程式を解くための Fortran サブルーチンライブラリである[19)]．LINPACK は行列演算ライブラリである BLAS[20)] (Basic Linear Algebra Subprograms) 上に構

築される．現在では，非対称密行列を係数行列とする連立一次方程式を解く際の演算性能を評価するベンチマークテストを指すことが多い．

LINPACK ベンチマークテストには次の 3 種類のベンチマークテストがある．

- LINPACK Benchmark N=100
  N=100 に固定して，LU 分解と解ベクトルの計算にかかった時間を計測する．使用するルーチンは DGEFA，DGESL(単精度の場合は SGEFA，SGESL) の 2 種類で，それぞれ LU 分解，$x$ の求解を実行する．規定によりソースの改変ができないため，ハードウェアおよびコンパイラを対象としたベンチマークテストであるといえる．

- Toward Peak Performance
  係数行列は N=1000 で固定するが，ユーザがソースプログラムの改変をすることが認められている．このため計算機システムが発揮できる最大の演算性能を試すためのベンチマークテストであるといえる．

- Highly Parallel Computing
  このベンチマークは Top500 で採用されており，係数行列の次元数やブロックサイズなどユーザが設定できるので，マシンの最も良い性能を評価することが可能である．現在は HPL というパッケージで配布されている．HPL については 6 節で詳しく述べる．

LINPACK の演算量は次式で評価されることが規定されている．

$$\text{演算量} = \frac{2}{3}N^3 + O(N^2) \tag{3}$$

これは係数行列 $A$ を $LU$ 分解した後，前進・後退代入によって解ベクトル $x$ を求めるという直接解法を適用することを前提にした演算量である．具体的な $LU$ 分解の方法と演算量について 5.1 節で述べる．

### 5.1 LU 分解の直接解法アルゴリズム

$A$ を $n \times n$ 行列，$b$ を $n$ 次元ベクトルとするとき，行列方程式

$$Ax = b \tag{4}$$

を解く．行列 $A$ の $LU$ 分解とは、行列 $A$ を三角行列 $L$ と上三角行列 $U$ の積で

$$A = LU \tag{5}$$

と表すことである．ここで，$n \times n$ 行列 $L$ が下三角行列とは，行列 $L$ の 第 $ij$ 成分を $L_{ij}$ と書くとき，

$$L_{ij} = 0 \qquad (i > j \text{ のとき}) \tag{6}$$

が成立することである．また，$n \times n$ 行列 $U$ が上三角行列とは，$U$ の第 $ij$ 成分を $U_{ij}$ と書くとき，

$$U_{ij} = 0 \qquad (i < j \text{ のとき}) \tag{7}$$

が成立することである．行列 $A$ が $LU$ 分解されると式 (4) は次のように解かれる．

$$Ax = LUx = b \tag{8}$$

より，$y = Ux$ とおいて，まず前進代入，

$$Ly = b \tag{9}$$

を解く．三角行列を係数とする方程式 (9) は容易に解くことができる．実際，$y = (y_1, y_2 \cdots, y_n)$ とするとき，式 (9) を $y_n, y_{n-1} \cdots, y_1$ の順に解いて行けば良い．この計算は $O(n^2)$ 回の 浮動小数点数演算でできる．こうして，$y$ が求められたならば後退代入，

$$Ux = y \tag{10}$$

を解いて，$x$ を求める．この計算も $O(n^2)$ で解くことができる．

## 6. High-Performance LINPACK Benchmark

HPL(High-Performance LINPACK Benchmark) は，LINPACK の実装の一つである．分散メモリ型並列計算機用のベンチマークソフトウェアであり，ガウス消去法を用いた密行列連立一次方程式の求解にお

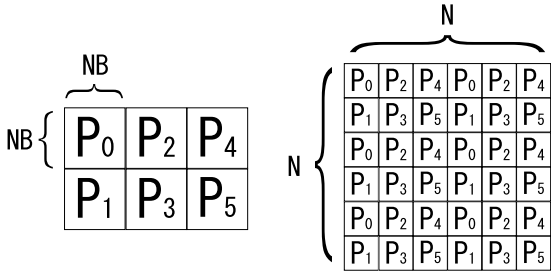


Fig. 3. ブロックサイクリック分割.

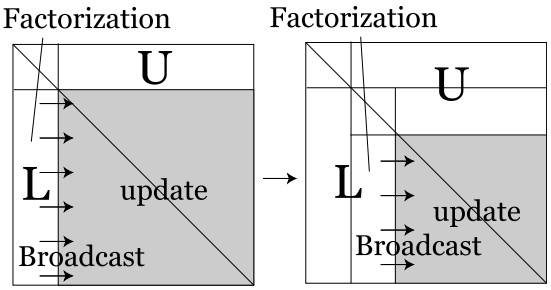


Fig. 4. 更新計算.

ける実行時間により性能を評価する．行列計算ライブラリに ATLAS[21] (Automatically Tuned Linear Algebra Software) を用いる．HPL は様々なパラメータを計算機の特性に合わせて設定でき，高度に最適化された行列演算カーネルを組み込むことで，より高い性能を得ることができるようになっている．

### 6.1 アルゴリズム

HPL では，まず Fig. 3 のようにプロセスをプロセスグリッドという 2 次元配列の格子状にブロックサイクリックに並べ，係数行列を複数の正方形に分解してプロセスグリッド上に割り当てる．LU 分解処理は Fig. 4 のように Panel Factorization，Panel Broadcast，Update，Backward Substitution というフェイズから構成される．それぞれにおいてパネル列 LU 分解，分解済みパネルの送信，未分解小行列の更新計算を行い，$L$ と $U$ を求めてから後退代入演算による求解を行う．

### 6.2 パラメータ

HPL では次の 16 項目についてのパラメータ[22] を設定できる．性能に大きく影響を与えるものは問題サイズ N，ブロックサイズ NB，プロセスグリッド (P，Q)，Broadcast のトポロジーなどである．

- 問題サイズ N
- ブロックサイズ NB
- プロセスグリッド (P，Q)
- 解のチェックにおける残差の境界値
- Panel Factorization のアルゴリズム
- 再帰的 Panel Factorization のアルゴリズム
- 再帰的 Factorization におけるサブパネル数
- 再帰的 Factorization におけるサブパネル幅の最小値
- Panel Broadcast のトポロジー
- Look-ahead の深さ
- Update における通信トポロジー
- long における U の平衡化処理の有無
- mix における行数の境界値
- L1 パネルの保持の仕方
- U パネルの保持の仕方
- メモリの alignment

## 7. 計測結果

### 7.1 HPL のパラメータの設定

LINPACK においてシステムの最大実行性能を得るためにはシステムの特性にあった最適なパラメータを設定する必要がある．そこで，HPL の最適なパラメータについて調査を行った．前節で述べた計測に大きく影響するパラメータそれぞれについて，結果から得られた Xenia の最適なパラメータについて述べる．

#### 7.1.1 問題サイズ N

問題サイズ N は HPL で解く問題の大きさである．つまり，HPL では N 次元連立方程式を解くことになる．N は HPL の結果に最も大きな影響をもたらす．N の値は，計算対象となる計算機の全メモリ容量の 80 %を使用するように設定する．これより，導き出された最適な N の値は 82897 となる．この値付近で，N を変動させ HPL の計測を行った結果を Fig. 5 に示す．用いたコンパイラは gcc-2.96，最適化オプションは -fomit-frame-pointer -O3 -funroll-loops で，N 以外の主なパラメータは NB:80，(P，Q):(8，16)，BCAST:1ringM である．この結果を見ると，問題サイズが 82000 の時に性能劣化が見られるが 83000 以降は問題サイズが大きくなるにつれて高いパフォーマンスを示し，86000 の時に最も良い値となっている．計算で求めた最適な N である 82897 よりも大きくなっているが，これはメモリの空き容量を増加させるために並列計算に必要のないいくつかのサービスを停止し，多くのメモリを使用可能としたためである．86000 ではメモリの 95 %以上が使用され，スワップが頻繁に起こっている．つまり，これ以上問題サイズを上げると極端に結果が悪くなる．

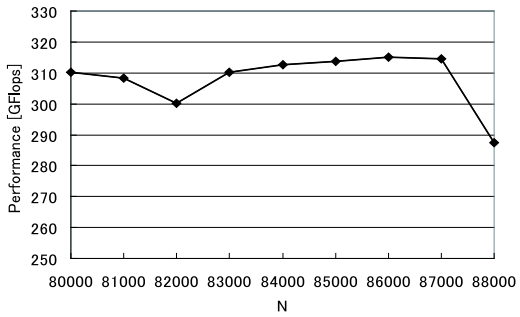


Fig. 5. N による比較.

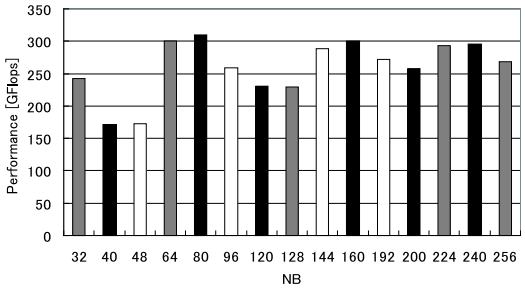


Fig. 6. NB による比較.

#### 7.1.2 ブロックサイズ NB

ブロックサイズ NB は問題をどのような大きさに分けるかを決める粒度のことである．NB が大きくなると，通信量が減るがロードバランスが悪くなり，NB が小さくなると，通信量が増えるがロードバランスが良くなる．また，良い NB の値があればその整数倍も良い結果をもたらすことがある．NB は通常，HPL のパラメータを決定する際に最も設定が難しいとされている[7]．最適な NB を求めるために ATLAS のインストールログから得られた 40 の倍数と 48 の倍数を，2 の累乗である 32 の倍数の合計 3 通りの NB を用いて計測を行った．結果を Fig. 6 に示す．用いたコンパイラは gcc-2.96，最適化オプションは -fomit-frame-pointer -O3 -funroll-loops で，NB 以外の主なパラメータは N:80000，(P，Q):(8，16)，BCAST:1ringM である．

この結果より，40 の倍数である 80 が最も良い NB であるといえる．この 40 という値は，ATLAS が CPU のキャッシュサイズを認識する際に導き出す NB の値である．

#### 7.1.3 プロセスグリッド (P，Q)

プロセスグリッド (P，Q) は問題の行列をそれぞれのプロセスにどのように分割するかを示す．必然的に P と Q の積が実行ノード数となる．P と Q は等しいか，P より Q が大きい方が良いとされている．Xenia

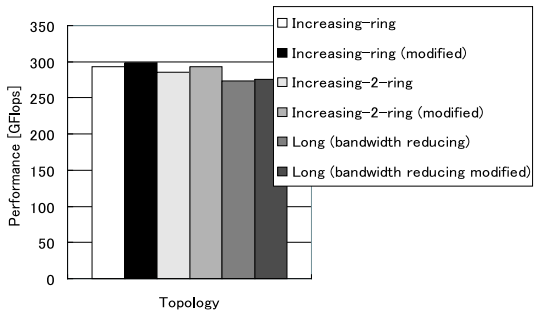


Fig. 7. Panel Broadcast による比較.

の実行プロセッサ数は 128 なので，この条件に合う (P，Q) は (8，16) となる．

#### 7.1.4 Panel Broadcast のトポロジー

Panel Broadcast のトポロジーには Increasing-1ring，Increasing-2ring，Bandwidth-reducing の 3 種類と，次の Panel Factorization を行うプロセスにメッセージ送信をさせない modified 版がそれぞれ 3 種類の計 6 種類が存在する．それぞれの方法に関して計測を行った結果を Fig. 7 に示す．用いたコンパイラは gcc-2.96，最適化オプションは -fomit-frame-pointer -O3 -funroll-loops で，BCAST 以外の主なパラメータは N:80000，NB:64，(P，Q):(8，16) である．

この結果より，Xenia における BCAST は Increasing-ring (modified) が最も良い性能を出すことが分かる．

### 7.2 コンパイラによる比較

gcc，Intel，pgi の 3 種類のコンパイラを用いて計測比較を行った．各コンパイラの最適化オプションは以下の通りである．

- GCC
  -fomit-frame-pointer -O3 -funroll-loops
- Intel
  -O3 -axKW
- PGI
  -fast -Mvect=sse

GCC では HPL がデフォルトで設定しているオプションを，Intel では Xeon プロセッサと類似したアーキテクチャをもつ Pentium4 に最適化するオプションを，PGI では SSE2 をサポートするオプションをそれぞれ用いている．用いたパラメータは 7.1 節で導き出した Table 7 の通りである．

計測結果は Fig. 8 のようになった．この結果より，HPL の計測に関しては PGI が最も良い値を示していることが分かる．

Table 7. Xenia における最適パラメータ.

| | |
|---|---|
| N | 86000 |
| NB | 80 |
| (P，Q) | (8，16) |
| Broadcast | Increasing-ring (modified) |

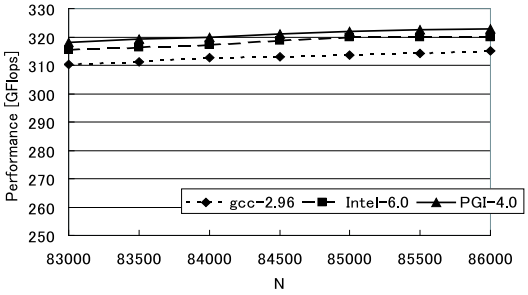


Fig. 8. コンパイラによる比較.

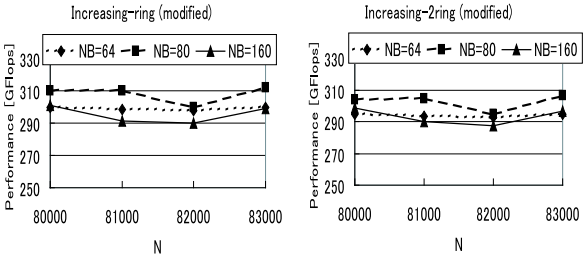


Fig. 9. 問題サイズ 82000 の検討 (P，Q)=(8，16).

### 7.3 問題サイズ 82000 の検討

Fig. 5 より問題サイズ N が 82000 の時に大きくパフォーマンスが劣化していることが分かる．この結果より，パラメータそれぞれに依存関係があるかを確認するために異なるパラメータを数種類用いて N が 82000 のときのパフォーマンスについて検討した．用いたパラメータは N を 80000 から 83000 まで 1000 ずつ増加させ，NB を 64，80，160 の 3 通り，BCAST を 1ringM と 2ringM の 2 通り，(P，Q) の値を 3 通りに変化させ計測を行った結果を Fig. 9，Fig. 10，Fig. 11 に示す.

これらの結果より，問題サイズが 82000 でもパフォーマンスが低下しない組合せが存在することが分かる．このことよりパラメータには依存関係が存在することが分かる．しかし，全体的なパフォーマンスで比較すると，Table 7 に示すパラメータが最も高いパフォーマンスを示していることが分かる.

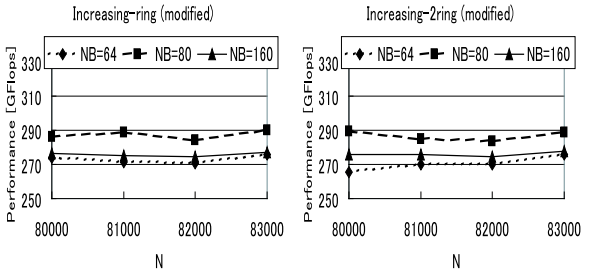


Fig. 10. 問題サイズ 82000 の検討 (P，Q)=(16，8).

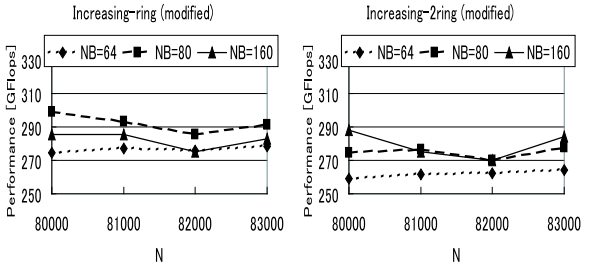


Fig. 11. 問題サイズ 82000 の検討 (P，Q)=(4，32).

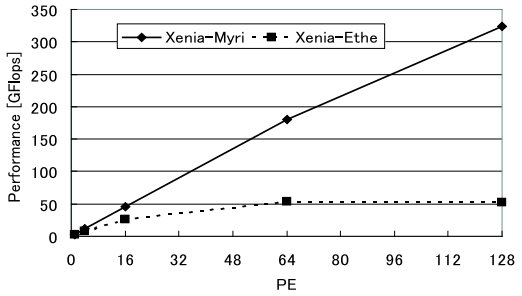


Fig. 12. Myrinet と Ethernet の比較.

### 7.4 Myrinet と Ethernet の比較

Myrinet-2000 と 100BASE-T の FastEternet で比較を行った．1CPU から 128CPU まででそれぞれについて最も高いパフォーマンスを示すパラメータを用いた Xenia のスケールを Fig. 12 に示す.

この結果より，Myrinet-2000 を使用するとほぼ線形にパフォーマンスが上がっていることが分かる．また FastEternet では CPU 数が増えるにつれてパフォーマンスが収束している．これは，データの送受信などで頻繁にオーバーヘッドが起こっているからであると考えられる．大規模なクラスタではオーバーヘッドの

| | | | |
|---|---|---|---|
| 190 | **IBM**<br>Netfinity Cluster PIII 1.13 GHz - Eth/ 1040 | **326.00**<br>1175.00 | WesternGeco<br>USA/2002 |
| 191 | **IBM**<br>Netfinity Cluster PIII 1.13 GHz - Eth/ 1024 | **326.00**<br>1150.00 | Compagnie Generale de Geophysique (CGG)<br>UK/2002 |
| 192 | **IBM**<br>Netfinity Cluster PIII 1.13 GHz - Eth/ 1024 | **326.00**<br>1150.00 | Compagnie Generale de Geophysique (CGG)<br>UK/2002 |
| 193 | **IBM**<br>Netfinity Cluster PIII 1.13 GHz - Eth/ 1024 | **326.00**<br>1150.00 | Shell<br>Netherlands/2002 |
| 194 | **IBM**<br>Netfinity Cluster PIII 1.13 GHz - Eth/ 1024 | **326.00**<br>1150.00 | WesternGeco<br>Egypt/2001 |
| 195 | **Self-made**<br>Xenia / IBM Intellistation Xeon 2.4 GHz Myrinet/ 128 | **323.40**<br>614.40 | Intelligent Information Center, Doshisha University<br>Japan/2002 |
| 196 | **Sun**<br>Fire 15k/Fire 6800/Sun Fire Link/ 336 | **321.00**<br>633.00 | **High Performance Computing Virtual Laboratory**<br>Canada/2002 |
| 197 | **Aspen Systems Inc**<br>P4 Xeon Cluster 2 GHz - Myrinet/ 264 | **320.80**<br>1056.00 | University of Oklahoma<br>USA/2002 |
| 198 | **Fujitsu**<br>VPP700/160E/ 160 | **319.00**<br>384.00 | Institute of Physical and Chemical Res. (RIKEN)<br>Japan/1999 |
| 199 | **SGI**<br>ORIGIN 3000 400 MHz/ 512 | **315.50**<br>409.60 | CSAR at the University of Manchester<br>UK/2001 |

Fig. 13. 2002 年 11 月の TOP500.

少ない Myrinet が有効であることが分かる．

## 8. まとめ

Xenia の最大実行性能を LINPACK を用いて解析した．今回の結果より，HPL のソースに添付されたパラメータ設定方法以外に得られた知見を以下に示す．

- 並列計算に必要のないサービスを停止することにより，スワップが起こらない限界の値まで問題サイズを大きくすることができる．
- ブロックサイズは CPU のキャッシュサイズに応じた最適な値にする．
- コンパイラにより，パフォーマンスが変動する．
- パラメータには依存関係が存在する．

また，7 節で最適なパラメータおよびコンパイラについて検討し，Panel Factorization のアルゴリズム 3 通り，再帰的 Panel Factorization のアルゴリズム 3 通りを組み合わせた 9 通りで数回計測を行った．その結果，323.4GFlops という値を得ることができた．この値で Xenia は 2002 年 11 月度の第 20 回 Top500 リスト (Fig. 13) において 195 位を達成した．

高性能な PC クラスタの構築を目指して LINPACK を対象に性能を見積もり，それに応じてハードウェア，ソフトウェアを構成し，Xenia クラスタを構築した．Xenia のシステム構成にあった HPL のパラメータチューニングを行い，チューニングを行う前の計測値から大幅に実行性能を向上させることに成功した．得られた実行性能値はシステム導入前の性能見積り以上の結果となり，HPL の知見を数多く得ることができた．

## 参 考 文 献


1) Rajkumar Buyya. *High Performance Cluster Computing: Architecture and Systems*, Vol. 1. Prentice Hall, 1999.

2) Rajkumar Buyya. *High Performance Cluster Computing: Programming and Applications*, Vol. 2. Prentice Hall, 1999.

3) Kai Hwang. *Scalable Parallel Computing*. WCB/McGraw-Hill, 1998.

4) R. Brightwell, H. E. Fang, and L. Ward. Scalability and Performance of CTH on the Computational Plant. In *Proceedings of 2nd International Conference on Cluster-Based Computing*, 2000.

5) Myricom Home Page. `http://www.myri.com`.

6) TOP500 Supercomputer Sites. `http://www.top500.org`.

7) HPL A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. `http://www.netlib.org/benchmark/hpl/`.

8) T. Sterling, D. Savarese, D. J. Beeker, J. E. Dorband, U. A. Renawake, and C. V. Packer. Beowulf: A parallel workstation for scientific computation. *In Proceedings of the 24th International Conference on Parallel Processing*, pp. 11–14, 1995.

9) Donald J. Becker, Thomas Sterling, Daniel Savarese, John E. Dorband, Udaya A. Ra nawak, and Charles V. Packer. BEOWULF: A PARALLEL WORKSTATION FOR SCIENTIFIC COMPUTATION. In *Proceedings of International Conference on Parallel Processing*, 1995.

10) T. L. Sterling, J. Salmon, D. J. Beeker, Savarese, and D. F. Savarese. How to build a beowulf: A

guide to the implementation and application of pc clusters. *MIT Press*, 1999.

11) MPI Forum. *MPI: A Message Passing Interface Standard*. 1995.

12) The Message Passing Interface (MPI) standard. `http://www-unix.mcs.anl.gov/mpi`.

13) A. Geist, A. Beguelin, J. Dongarra, R. Manchek, W. Jiang, and V. Sunderam. *PVM: A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, 1994.

14) PC Cluster Consortium. `http://pdswww.rwcp.or.jp`.

15) H. Tezuka, A Hori, Y. Ishikawa, and M. Sato. Pm: An operating system coordinated high performance communication library. *In High-performance Computing and Networking97*, pp. 708–717, 1997.

16) GCC Home Page. `http://gcc.gnu.org`.

17) Intel Compilers. `http://www.intel.com/software/products/compilers`.

18) The Portland Group Compiler Technology. `http://www.pgroup.com`.

19) The linpack benchmark. `http://www.netlib.org/benchmark/top500/lists/linpack.html`.

20) Basic Linear Algebra Subprograms. `http://www.netlib.org/blas/`.

21) Automatically Tuned Linear Algebra Software. `http://math-atlas.sourceforge.net`.

22) 笹生健, 松岡聡. HPL のパラメータチューニングの解析. ハイパフォーマンスコンピューティング, Vol. 91, No. 22, pp. 125–130, 8 2002.