The Science and Engineering Doshisha University, Vol.XX, No.Y November 2003

Construction of Tera Flops PC Cluster System and evaluation of performance using Benchmark

Tomoyuki Hiroyasu* Mitsunori Miki* and Hiroshi Arakuta**

(Received October 6, 2004)

Complicated and diverse of objective problems with the development of technology, demand of high performance computer is increasing. Instead of vector superocomputer, in order to meet the needs of these demand, PC cluster systems have gotten a lot of attention in recent years. PC cluster systems consist of many PCs connected by network and are used for parallel or distributed computation. In scientific and engineering fields, HPC cluster systems are attractive to the computationally intensive tasks. We setting up of HPC cluster system called Supernova, toward a performance of 1 Tera Flops. Supernova is composed 256-node and running Linux Operating System. We evaluated this cluster system using High-Performance LINPACK Benchmark and Himeno Benchmark. In this paper, we present the result of performance and the combination of parameters using these Benchmarks in Supernova. Through experimenting we obtained knowledge to perform parameter tuning for better performance in these Benchmarks and knowledge about construction of PC Cluster System.

Key words ： PC Cluster, Himeno Benchmark, LINPACK Benchmark, Linux

キーワード ： PC クラスタ, 姫野ベンチマーク, リンパックベンチマーク, リナックス

テラフロップスクラスタの構築とBenchmarkによる性能評価

廣安知之・三木光範・荒久田博士

1. はじめに

科学技術の発達に伴う対象問題の複雑化，大規模化により，高性能計算機の必要性が高まっている．それらの要求を満たし，高い性能を実現したものは特別なハードウェア構成を持つ専用計算機であった．しかし近年，従来の超並列計算機に代わり，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する PC クラスタシステム [1, 2]（以下クラスタ）が注目されている．クラスタ最大の特徴は，同程度の性能を有する専用並列計算機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．クラスタの登場により，一部の研究機関や企業のみ所有が可能であった高性能計算機を研究室やグループ単位で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[3] においても，クラスタシステムはスーパーコンピュータにひけをとらない性能を見せている．TOP500 へのランクインは所有機関にとって高性能計算機の所有を世界にアピールする最大の機会であるため，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

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

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，1TFlops の実行性能値を有するクラスタシステムの構築を目標に，Supernova Cluster System（以下 Supernova）を構築した．本稿では姫野 Benchmark および High-Performance LINPACK Benchmark により，Supernova の性能評価を行う．

2. PC クラスタ

PC クラスタは単一で稼動するコンピュータの集合であり，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 などを有しており，以下のような特徴を持っている．

- 優れた価格対性能比
- 市場最新技術を導入可能
- 高拡張性

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
  HPC クラスタは，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．HPC クラスタの代表的なものとして，Beowulf クラスタがある．Beowulf クラスタは，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 [4, 5, 6] が提唱するクラスタシステムとして発展してきた．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である Linux や FreeBSD などの OS を用いて構築され，MPI や PVM を用いた並列処理プログラムを実行できる．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ことの典型的な実例である．
  クラスタリングソフトウェア SCore[7, 8] 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（Real-World Computing Partnership：新情報処理開発機構）において開発が始まった．Beowulf クラスタと異なり，SCore クラスタ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ソフトウェアとして PC クラスタコンソーシアムにより提供されている．
- HA(High Availability) クラスタ
  HA クラスタは，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る．HA クラスタの特徴は，冗長化構成により障害発生時には切り替え作業で対処できる点である．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，ファイルサーバの他，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. Supernova Cluster System

1TFlops のピーク性能値を有するクラスタの構築を目指し，本研究室では 2003 年 9 月に Fig. 1 に示すクラスタを導入した．導入したクラスタは Supernova Cluster System と呼び，以降 Supernova と示す．Supernova は，米国 AMD 社の 64bit CPU である Opteron プロセッサ 512CPU により構成される大規模なクラスタシステムである．主なハードウェア構成，ネットワーク構成を Table 1 に示す．

Fig. 1. Supernova Cluster System.

Table 1. Supernova のハードウェア構成.

ノード数	256
CPU	AMD Opteron 1.8GHz × 512
Memory	2GB × 256（total 512GB）
OS	Turbolinux for AMD64
通信ライブラリ	mpich-1.2.5
通信プロトコル	TCP/IP
通信媒体	Gigabit Ethernet

Opteron は 1 プロセッサにつき 2 つの浮動小数点演算ユニット（FPU）を有しており，各ユニットは 1 クロックで 1 回の演算を行うことが可能である．このことより，Supernova のピーク性能値（Rpeak）は，式 (1) より 1.8432TFlops となる．

$$Rpeak = \#CPU \times ClockFrequency \times \#FPU \quad (1)$$

3.1 Opteron プロセッサ

Supernova では，AMD Opteron プロセッサを使用している．Opteron プロセッサは次のような特徴を備えている．

- 統合メモリコントローラ
  Opteron プロセッサでは，ノースブリッジおよびメモリコントローラをプロセッサに内蔵している．統合メモリコントローラによりメモリとプロセッサが直接接続され，メモリアクセスをノースブリッジを介することなく行うことができるようになる．そのためメモリのレイテンシが著しく低減し，性能向上が実現する．

- HyperTransport リンク
  HyperTransport リンク [9] はプロセッサ間，I/O サブシステム間の接続バンド幅を拡張可能とする．最高で 3 つの HyperTransport リンクにより，プロセッサあたり最高 19.2GB/s のピーク時バンド幅を実現する．これにより，ボトルネックの解消および低減，バンド幅の向上とレイテンシの低減が行われ，システム全体の性能向上を可能とする．

3.2 ネットワークスイッチ

Supernova ではノード間通信を行うためのスイッチとして，米国 Force10 Networks 社の E1200 を使用している．E1200 は，1.44Tbps のバックプレーンを持ち，毎秒 5 億パケットの処理速度を有する．また，最高 336 ノード間のノンブロッキング通信が可能な超高性能スイッチである．Supernova は 256 ノードから構成されるため，全ノード間においてノンブロッキング通信が可能である．E1200 の使用により，スケーラブルな性能向上を期待することができる．

3.3 Supernova の特徴

Supernova の特徴の一つとして，利用しているプロセッサがあげられる．3.1 節で述べたとおり，Opteron は統合メモリコントローラ，HyperTransport リンクを備えている．また使用電力が比較的小さい．クラスタを構築する際の大きな障害として，多ノードから構成されるために使用電力が大きくなること，発熱量が大きくなることがあげられる．少しでも使用電力の小さいプロセッサを利用することは，この問題の解決につながる．

もう一つの大きな特徴として，ギガビットイーサネットの利用があげられる．イーサネットワークは大学やオフィスで通常使われているネットワークの規格であり，100Mbps の大域が現在主流である．ギガビットイーサネットワークは 1Gbps の大域を持ち，これまでその通信性能は高くないと考えられてきた．そのため PC クラスタの構築においては，米国 Myricom 社の Myrinet や，InfiniBand[10] の利用が主である．しかしこれらのネットワークは高価であり，コストパフォーマンスに優れない．これに対して Supernova では，安価でコストパフォーマンスに優れたギガビットイーサネットワークを利用している．

4. Benchmark

4.1 Benchmark とは

ベンチマークという言葉には「基準となるもの」という意味がある．コンピュータ用語としてのベンチマークの意味は，ハードウェアやソフトウェアの処理速度を計測する試験問題，および性能評価という意味を持つ．つまり，計算機システムの指標を意味する．コンピュータプログラム次第で，様々な挙動を示すため，用途に応じた評価の指標を決定し，その尺度にもとづいて性能を評価しなければならない．PC クラスタに関する並列計算ベンチマークとして挙げられるのが，姫野 Benchmark[11]，LINPACK[12]，Nas ParallelBenchmark[13] などである．本稿において使用するベンチマークは，姫野 Benchmark および LINPACK である．

4.2 姫野 Benchmark

姫野 Benchmark は，理化学研究所の情報基盤センター長である姫野龍太郎氏が非圧縮流体解析コードの性能評価のために考えたものである．構造格子系の解法である差分法に基づいて空間離散化を行い，流体解析，熱解析，電磁波解析などにおける編微分方程式の求解において現れる Poisson 方程式解法を Jacobi 法による反復で解く際の主要なループの処理速度を計測する．

4.2.1 姫野 Benchmark のパラメータ

姫野 Benchmark において，計測に大きく影響を及ぼすパラメータは，次のとおりである．

- 配列サイズ
  姫野 Benchmark は，三次元配列を用いて Jacobi 法による反復計算を行っている．三次元配列の要素数は Table 2 より選択することが可能である．配列要素は単精度実数型であり，大きな配列ほどメモリの使用量が大きくなる．計算機が搭載する総メモリ量に対して大きすぎる配列を選択した場合メモリ不足により計算ができない場合があるため，計算機に適した配列サイズを選択しなければならない．

Table 2. 姫野 Benchmark における配列要素.

Array Size	#Array Elements
XS	65 × 33 × 33
S	128 × 64 × 64
M	256 × 128 × 128
L	512 × 256 × 256
XL	1024 × 512 × 512

- グリッド分割
  グリッド分割とは，問題配列を x，y，z 方向に分割する際の分割数のことである．計算に使用するノードの各 CPU に任意の割合で問題配列を振り分けることができる．分割数の積は，計算に使用する CPU 数と同じでなくてはならない．

4.2.2 Poisson 方程式の解法

Poisson 方程式は，式 (2) と表現される．

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f(x, y, z) \tag{2}$$

式 (2) を，式 (3) のように差分方程式にする．

$$\begin{aligned} f_{i,j,k} = & \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{\Delta x^2} \\ & + \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{\Delta y^2} \\ & + \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{\Delta z^2} \end{aligned} \tag{3}$$

$\Delta x = \Delta y = \Delta z$ とすると，式 (3) は式 (4) のように変換することができる．

$$\begin{aligned} u_{i,j,k} = \frac{1}{6}[& u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i,j-1,k} \\ & + u_{i,j,k+1} + u_{i,j,k-1} - (\Delta x)^2 f_{i,j,k}] \end{aligned} \tag{4}$$

$u_{i,j,k}$ を求める手法として，Gauss-Seidel 法と Jacobi 法が知られている．姫野 Benchmark で利用されている Jacobi 法で求めるには，$u^{m+1}_{i,j,k}$(m：反復回数) を式 (5) のように計算する．

$$\begin{aligned} u^{m+1}_{i,j,k} = \frac{1}{6}[& u^m_{i+1,j,k} + u^m_{i-1,j,k} + u^m_{i,j+1,k} + u^m_{i,j-1,k} \\ & + u^m_{i,j,k+1} + u^m_{i,j,k-1} - (\Delta x)^2 f_{i,j,k}] \end{aligned} \tag{5}$$

姫野 Benchmark に Jacobi 法が用いられるのは，ベクトル化が容易であることと，行列の分割による並列処理を行うことが可能であるためである．

4.3 LINPACK Benchmark

LINPACK は米国 Tenessee 大学の J.Dongarra 博士らによって開発された主に浮動小数点演算のためのベンチマークであり，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．LINPACK は行列演算ライブラリである BLAS（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要した時間を計測する．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，ハードウェア及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を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 については次節で詳しく述べる．

LINPACK の演算量は，式 (6) で評価されることが規定されている．

$$\text{演算量} = \frac{2}{3}N^3 + O(N^2) \tag{6}$$

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，4.3.1 項で述べる．

4.3.1 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

$$Ax = b \tag{7}$$

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 の積で式 (8) のように表すことである．

$$A = LU \tag{8}$$

ここで，$n \times n$ 行列 L が下三角行列であるとは，行列 L の第 ij 成分を L_{ij} と書くとき，式 (9) が成立することである

$$L_{ij} = 0 \ (i > j) \tag{9}$$

また，$n \times n$ 行列 U が上三角行列であるとは，行列 U の第 ij 成分を U_{ij} と書くとき，式 (10) が成立することである．

$$U_{ij} = 0 \ (i < j) \tag{10}$$

行列 A が LU 分解されると，式 (7) は式 (11) のように解くことができる．

$$Ax = LUx = b \tag{11}$$

$y = Ux$ とおき，まず前進代入を解く．

$$Ly = b \tag{12}$$

三角行列を係数とする式 (12) は容易に解くことができる．$y = (y_1, y_2, \cdots, y_n)$ とすると，式 (12) を y_n , y_{n-1}, $\cdots$, y_1 の順に解けばよい．この計算は $O(n^2)$ 回の浮動小数点演算を解くことになる．y を求めた後，式 (13) に示す後退代入を解き，x を求める．この計算も $O(n^2)$ 回の浮動小数点演算で解くことができる．

$$Ux = y \tag{13}$$

4.4 High-Performance LINPACK Benchmark

HPL（High-Performance LINPACK Benchmark）は，LINPACK 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は，BLAS に準拠した ATLAS（Automatically Tuned Linear Algebra Software）や goto-library 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を行うことができ，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．

4.4.1 HPL のアルゴリズム

HPL では，まず Fig. 2 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. 3 のように Panel Factorization，Panel Broadcast 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り式 (13) の求解を行う．

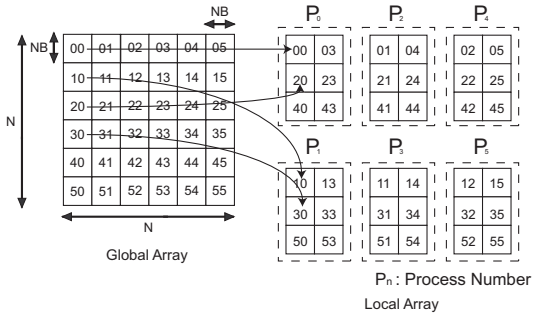


Fig. 2. ブロックサイクリック分割.

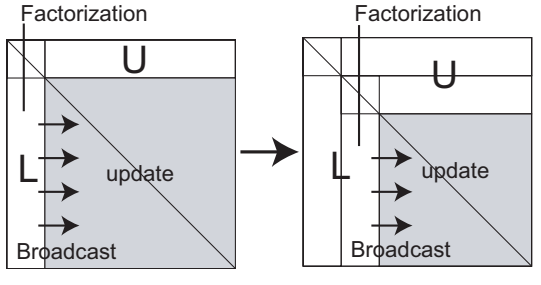


Fig. 3. 更新計算.

4.4.2 パラメータ

HPL では，次の 16 項目についてのパラメータを設定できる．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．

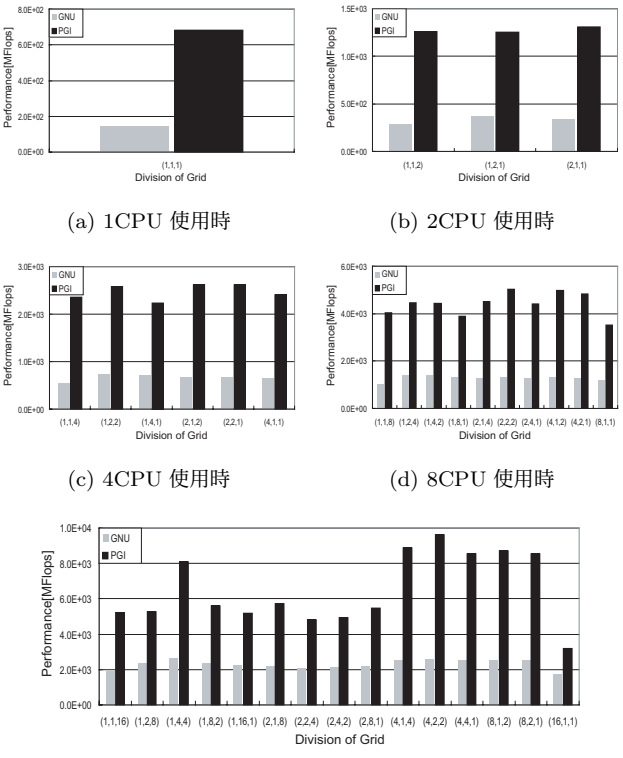
- 問題サイズ N
- ブロックサイズ NB
- プロセスグリッド （P，Q）
- Panel Broadcast のトポロジー
- Look-ahead の深さ
- Update における通信トポロジー
- long における U の平衡化処理の有無
- mix における行数の境界値
- L1 パネルの保持の仕方
- U パネルの保持の仕方
- メモリの alignment
- 解のチェックにおける残差の境界値
- Panel Factorization のアルゴリズム
- 再帰的 Panel Factorization のアルゴリズム
- 再帰的 Factorization におけるサブパ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

5. 姫野 Benchmark の計測

5.1 グリッド分割およびコンパイラの検討

4.2.1 項で述べたとおり，グリッド分割は姫野 Benchmark の計測値に大きく影響を及ぼす．そこで，グリッド分割に関する検討を行った．また，姫野 Benchmark ではコンパイラによって性能値が大きく異なるため，コンパイラに関する検討を同時に行った．Fig. 4，Fig. 5 に結果を示す．計測において配列サイズは M を用いた．使用 CPU 数は，1CPU，2CPU，4CPU，8CPU，16CPU である．コンパイラは GNU Fortran Compiler 3.2 および PGI Fortran Compiler 5.0 である．

Fig. 4 より，グリッド分割による実行性能値の差はノード数が増えるほど大きくなっていることがわかる．コンパイラに関しては，PGI が GNU と比べ明らかに優れた結果を示している．また Fig. 5 より，使用 CPU 数が多い程 PGI は GNU より良好な結果を示すと考えられる．



(a) 1CPU 使用時　(b) 2CPU 使用時

(c) 4CPU 使用時　(d) 8CPU 使用時

(e) 16CPU 使用時

Fig. 4. グリッド分割およびコンパイラの検討.

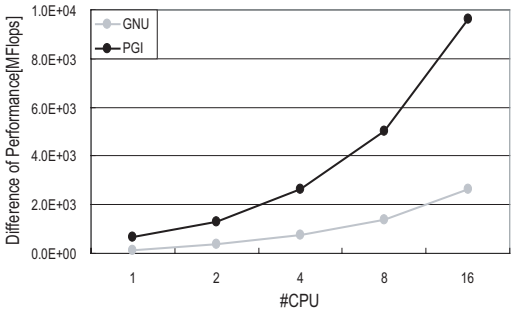


Fig. 5. 使用 CPU 数の増加に伴う性能値の推移.

グリッド分割に関して，PGI 使用時における使用 CPU 数ごとの最適なグリッド分割，(x，y，z) の組み合わせは，Fig. 4 より 1CPU：(1，1，1)，2CPU：(2，1，1)，4CPU：(2，1，2)，8CPU：(2，2，2)，16CPU：(4，2，2) であることがわかる．このことよりグリッド分割はできるだけ各 CPU へ均等に振り分けるように設定することで良好な結果を得ることができると考えられる．

5.2 コンパイルオプションの検討

5.1 節において，PGI が良好な計測値を示すことがわかった．そこで，PGI におけるコンパイルのオプションに関する検討を行った．検討を行ったコンパイルオプションは，下記に示すオプションを利用した全組み合わせである．

- -O1
  “-fthread-jumps” と “-defer-pop” を指定する．
- -O2
  “-O1” をさらに最適化する．空間と速度のトレードオフを含まないオプションをほぼ全て指定する．
- -O3
  “-O2” をさらに最適化する．“-O2” が行う最適化に加え，-finline-functions を有効にする．
- -fast
  キャッシュを最適化する．
- -Mvect=assoc
  ベクトル最適化において，ループの結合が可能であることを示す．
- -Mvect=cachesize:*
  ベクトル最適化における入れ子ループ処理において，仮定可能なキャッシュサイズを指定する．
- -Mcache_align
  キャッシュラインに合わせてデータを揃える．
- -Mnontemporal
  Prefetch においてデータ移動のスキームを変更する．

これらのオプションを用いることで可能な 128 種類の組み合わせに関して計測を行った．計測において配列サイズは M を用い，使用 CPU 数は 16 である．また，グリッド分割の組み合わせは，5.1 節で求めた (4, 2, 2) である．計測結果の一部を Table 3 に示す．

計測を行った結果，最も適したコンパイルオプションの組み合わせは “-fast -Mvect=assoc -O2” であることがわかった．このコンパイルオプションにより得られた計測値は 11271MFlops である．この計測値により Supernova は，2003 年 12 月に理化学研究所情報基盤センターによって行われた RIKEN BMT コンテスト 2003 の実践的 PC クラスタ部門において 1 位を達成した．

Table 3. 最適化オプションの検討結果（一部）．

Compile Option	Performance [MFlops]
None	9911
-fast -Mvect=assoc -O1	6610
-fast -Mvect=assoc -O2	11271
-Mcache_align -Mnontemporal	9661
-Mvect=cachesize:1048576 -O3	10918
-Mvect=assoc,cachesize:1048576	11209

6. HPL の主要パラメータ

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．4.4.2 項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説明する．

6.1 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．一般的に N の値が大きくなる程良い結果を得られるが，N の増加に伴い，メモリの使用量は増加する．

6.2 ブロックサイズ NB

ブロックサイズ NB は，HPL で解く問題の粒度である．NB が大きくなると通信量は減少する一方，ロードバランスが悪くなる．逆に，NB が小さくなると通信量は増加する一方，ロードバランスは良くなる．また良好な結果を示す NB の値があれば，その値の整数倍も良好な結果を示すことがある．

6.3 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ものである．P と Q の積が実行ノード数となる．一般的に，P，Q の値は等しい，もしくは P の値より Q の値が大きい方が良い結果が得られる．

6.4 Panel Broadcast のトポロジー

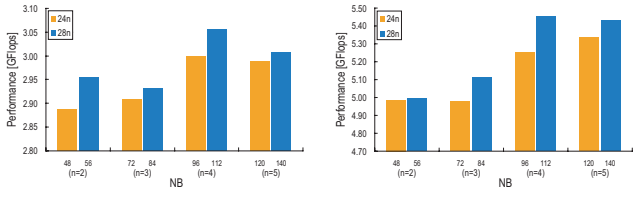
Panel Broadcast のトポロジーには Increasing-1ring，Increasing-2ring，Bandwidth-reducing の 3 種類と，次の Panel Factorization を行うプロセスにメッセージ送信をさせない modified 版がそれぞれ 3 種類の計 6 種類が存在する．normal 版と modified 版のトポロジーの流れは，次のとおりである．

- normal 版
  メッセージ受信 → メッセージ送信
  → Update → Panel Factorization
- modified 版
  メッセージ受信
  → Update → Panel Factorization

7. HPL パラメータの検討

7.1 ブロックサイズ NB 値の検討

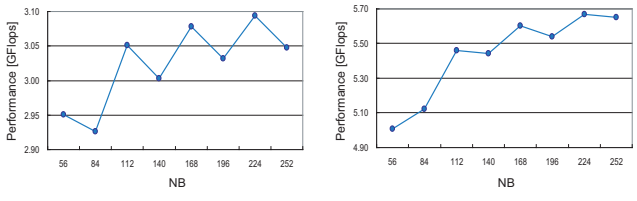
NB は通常，HPL のパラメータを決定する際に最も設定が困難であるとされている．また，32 から 256 の値において良好な結果を示すとされている [14]．最適な NB の値を求めるため，ATLAS が CPU のキャッシュサイズを認識する際に導き出した値より得られた 24 の倍数と 28 の倍数に関する計測を行った．NB 以外の主なパラメータは N:10000，BCAST:1ring であり，用いたコンパイラは gcc3.2，最適化オプションは-formit-frame-pointer -O3 -funroll-loops，演算ライブラリは atlas-3.5.6 である．結果を Fig. 6 に示す．



(a) 1CPU 使用時　　(b) 2CPU 使用時

Fig. 6. NB 値の検討．

Fig. 6 の結果より，NB には 28 の倍数が良好な結果を得ることができるといえる．28 の倍数に関する計測を続けて行い，最適な NB 値の検討を行った．計測の際に利用した NB 以外の主なパラメータ，コンパイラ，最適化オプションは先と同様である．結果を Fig. 7 に示す．



(a) 1CPU 使用時　　(b) 2CPU 使用時

Fig. 7. 最良 NB 値の検討．

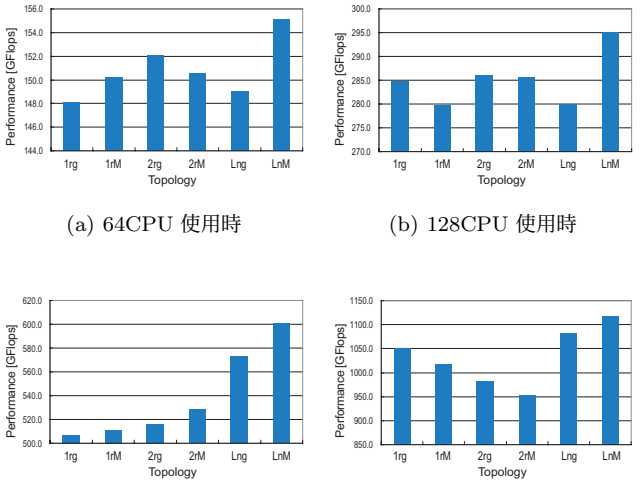
Fig. 7 より，最適な NB の値は 224 であることがわかる．また，NB に与えることで良好な結果が得られるのは 28 の倍数ではなく 56 の倍数であるといえる．

7.2 Panel Broadcast のトポロジーによる比較

Panel Broadcast の各トポロジーに関して計測を行った結果を Fig. 8 に示す．計測の際に利用した BCAST 以外の主なパラメータを Table 4 に示す．用いたコンパイラは gcc3.2，最適化オプションは-fomit-frame-pointer -O3 -funroll-loops，演算ライブラリは atlas-3.5.6 である．

Table 4. BCAST 検証に用いたパラメータ．

	64cpu	128cpu	256cpu	512cpu
N	80000	110000	160000	220000
NB	224			
(P，Q)	(8，8)	(8，16)	(16，16)	(16，32)



(a) 64CPU 使用時　　(b) 128CPU 使用時

(c) 256CPU 使用時　　(d) 512CPU 使用時

Fig. 8. Panel Broadcast の検証．

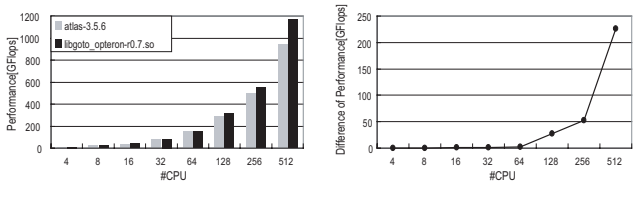
Fig. 8 より，BCAST には使用 CPU 数の違いにより様々な傾向があることがわかる．本来，normal 版と modified 版のトポロジーではメッセージ送信が無い分 modified 版を利用した場合が normal 版に比べ良好な結果を示すはずである．しかし Fig. 8 を見ると，normal 版が modified 版より良好な結果を示しているものがある．これらを踏まえ Fig. 8 よりわかるのは，使用 CPU 数の違いに関わらず，Supernova において BCAST は Long（bandwidth reducing modified）の modified 版が最も良い性能を出すということである．この Long 方式は一度の通信で大きなメッセージを送信する方式であり，Supernova のようにノードの処理速度が速く，比較的ネットワークが遅い環境に適している．

7.3 使用ライブラリによる比較

HPL 実行の際に使用する行列演算ライブラリを ATLAS を用いた場合と goto-library を利用した場合の計測結果を Fig. 9 に示す．計測の際に用いた主なパラメータは，Table 5 のとおりである．用いたコンパイラは gcc3.2，最適化オプションは-fomit-frame-pointer -O3 -funroll-loops である．

Table 5. ライブラリの比較に用いたパラメータ.

	4cpu	8cpu	16cpu	32cpu	64cpu
N	20000	28000	40000	56000	80000
NB	224				
(P, Q)	(2, 2)	(2, 4)	(4, 4)	(4, 8)	(8, 8)
library	atlas-3.5.6 libgoto_opteron-r0.7.so				
BCAST	Increasing-1ring				



(a) 異なるライブラリ使用時　　(b) 実行性能値の違いの推移

Fig. 9. 使用ライブラリの検討 .

Fig. 9(a) は atlas，goto-library を同じパラメータに対して使用した際の計測結果を比較したものであり，Fig. 9(b) は，atlas と goto-library の実行性能値の違いを示したものである．Fig. 9(a) より，goto-library が優れた結果を示していることがわかる．また Fig. 9(b) より，使用 CPU 数の増加に伴いライブラリの違いによる実行性能値の違いが大きくなっていることがわかる．このことより，使用するプロセッサ数が多いほど goto-library がより効果的であるといえる

7.4 プロセスグリッド (P，Q) の検討

プロセスグリッドにおいて，P と Q の積が実行プロセッサ数となった場合，良好な計測値を得ることができる．この条件と，6.3 節で述べた条件を満たす 512CPU 使用時の組み合わせは，(P, Q)：(16, 32) となる．この組み合わせが Supernova において適した組み合わせであるかを調べるため，複数の組み合わせについて計測を行った．計測結果を Fig. 10 に示す．計測に用いた主なパラメータは Table 6 であり，用いたコンパイラは gcc3.2，最適化オプションは-fomit-frame-pointer -O3 -funroll-loops である．

Table 6. グリッド分割の検証に用いたパラメータ.

N	200000
NB	224
(P, Q)	(1, 512), (2, 256), (4, 128) (8, 64), (4, 128)
library	libgoto_opteron-r0.7.so
BCAST	Increasing-1ring

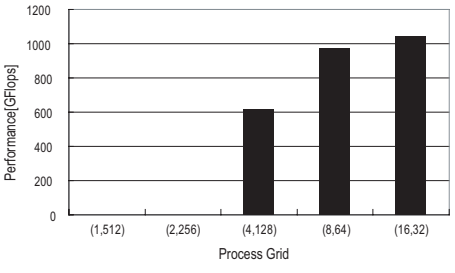


Fig. 10. プロセスグリッドの検討.

Fig. 10 より，(P, Q)：(16, 32) という組み合わせが最も良好な結果を示していることがわかる．(1, 512), (2, 256) の組み合わせに関しては，HPL 実行中，処理の終了前に HPL が停止してしまい，結果を得ることができなかった．HPL の実行中，メモリの使用率を確認したところ，一部のプロセッサは 100%近いメモリを利用していた一方で，メモリ使用率の極端に低いプロセッサが存在することがわかった．このことより，(1, 512), (2, 256) という組み合わせでは適切な問題分割が行われず，プロセッサの処理能力を超えるほど多くのプロセスを与えられるプロセッサと，少しの問題しか与えられないプロセッサが生じてしまったと考えられる．

7.5 問題サイズ N の検討

6.1 節で述べたとおり，N の値は大きくなるほど良好な結果を得られ，HPL の結果に大きな影響をもたらす．Supernova においてメモリの約 80%を使用する N の値は 226274 であるが，N を 220000 のような値で HPL の実行を行った際，計算機への負荷が高くなりすぎ，結果が出る前にプロセスが停止してしまうことがわかった．そこで N を 200000 以上の値から，5000 ず

Table 7.　ネットワークスイッチの違いによる検証の際に用いた主なパラメータ.

	1cpu	2cpu	4cpu	8cpu	16cpu	32cpu	64cpu	128cpu	256cpu	512 cpu
N	14000	20000	28000	40000	56000	80000	113000	160000	220000	220000
NB	224									
(P, Q)	(1, 1)	(1, 2)	(2, 2)	(2, 4)	(4, 4)	(4, 8)	(8, 8)	(8, 16)	(16, 16)	(16, 32)
BCAST	Increasing-1ring									
library	atlas-3.5.6									

つ増加させながら計測を行った．計測結果を Fig. 11 に示す．計測に用いた主なパラメータは NB：224，(P, Q)：(16, 32)，BCAST：LnM である．用いたコンパイラは gcc3.2，最適化オプションは-fomit-frame-pointer -O3 -funroll-loops である．

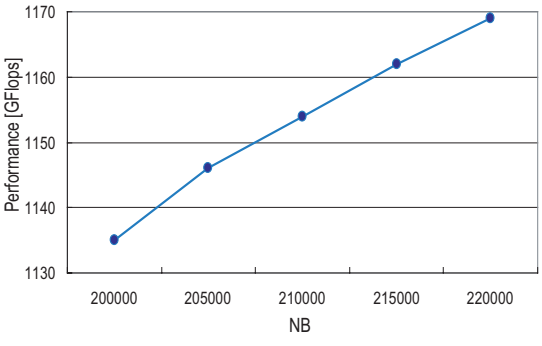


Fig. 11.　HPL 最高値の検討.

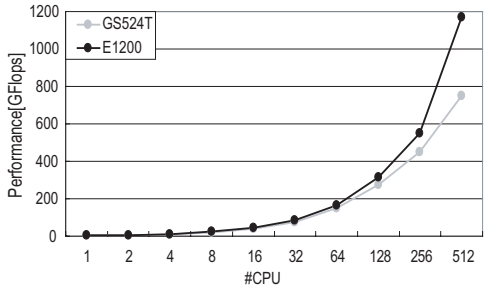
Fig. 11 より，N が 220000 で最も良好な結果を示していることがわかる．220000 を超える問題サイズでは前述したとおり，スワップを生じ実行完了前にプロセスが停止してしまうことがわかった．これ以上問題サイズを大きくしても，良好な結果を得ることができないと考えられる．これまでの計測により得られた Supernova における HPL の最適パラメータは Table 8 のようになる．

Table 8.　主なパラメータの最適値.

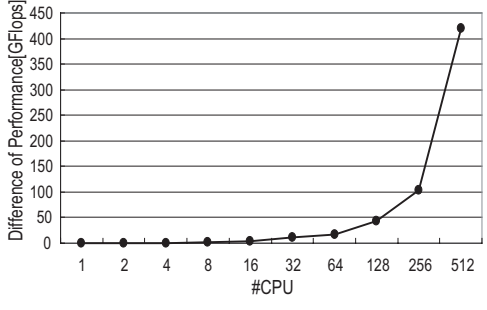
N	220000
NB	224
(P, Q)	(16, 32)
BCAST	Long（bandwidth reducing modified）
library	libgoto_opteron-r0.7.so

8.　ネットワークスイッチの検討

3.2 節で述べたとおり，Supernova ではネットワークを接続するためのスイッチとして E1200 を用いている．高速スイッチである E1200 をしようした場合と，米国 NETGEAR 社 [15] の比較的安価で低速の Gigabit Switch，GS524T を使用した場合の比較を行った．1CPU 使用時から 512CPU 使用時まで，Table 7 に示すパラメータを用いて計測を行った．計測結果を Fig. 12 に示す．用いたコンパイラは gcc3.2 であり，最適化オプションは-fomit-frame-pointer -O3 -funroll-loops，演算ライブラリは atlas-3.5.6 である．



(a) スイッチの比較



(b) 実行性能値の違いの推移

Fig. 12.　スイッチによる実行性能値の比較.

Fig. 12(a) より，E1200 が優れた結果を示していることがわかる．また Fig. 12(b) より，使用 CPU 数の増加に伴い E1200 使用時と GS524T 使用時における実行性能値の違いが大きくなっていることがわかる．このことより，使用 CPU 数が多いほど E1200 は良好な結果を示し，スケーラブルな性能を発揮するといえる．また Fig. 12(a) より，使用するネットワークスイッチに関わらず 128CPU 程度までは，実行性能値に大きな差はないことがわかる．このことより，ネットワークにギガビットイーサネットを利用にした際には使用 CPU 数が 128CPU 程度までの規模の小さい PC クラスタであれば，GS524T のように比較的低速ではあるが安価なスイッチを利用することにより，コストパフォーマンスに優れ，十分な性能を持つ PC クラスタを構築可能であるといえる．

9. まとめ

姫野 Benchmark においてパラメータの検討およびコンパイラの比較を，LINPACK の実装の一つである HPL においてパラメータチューニング，ライブラリおよびネットワークスイッチの比較を行った．これらを通じて得られた知見は，以下のとおりである．

- 姫野 Benchmark では，クラスタの規模が大きくなるにつれグリッド分割が大きく性能値に影響を及ぼす
- 姫野 Benchmark において使用するコンパイラとしては，PGI が良好な結果を示す
- HPL においてブロックサイズ NB を決定するには，ATLAS のインストールログより選定することで最良の値を見つけることができる．
- HPL の実行に使用するライブラリとして，goto-library が有効である
- 小規模な PC クラスタを構築する際には，使用するネットワークやスイッチは安価なものでよい

姫野 Benchmark において，パラメータおよびコンパイラの検討を行った．それにより得られた Supernova，16CPU 使用時における最大実行性能値は 11271MFlops である．この結果により，2003 年 12 月の RIKEN BMT コンテスト 2003 の実践的 PC クラスタ部門において Supernova は 1 位にランキングされた．

HPL のパラメータチューニングにおいては，チューニングを行う以前の計測から大幅に実行性能値を向上させることができた．Supernova において得られた最大実行性能値は 1.169TFlops であり，これはピーク性能値の約 63.4%である．この結果により，2003 年 11 月度の TOP500 において 93 位にランキングされた．またこの結果は国内で 6 位，国内の PC クラスタとしては 1 位である．これらの結果より，Supernova は国内だけでなく世界に誇ることのできる高速計算機であるといえる．

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

4) 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.

5) 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.

6) 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.

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

8) 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.

9) HyperTransport Consortium. `http://www.hypertransport.org/`.

10) InfiniBand Trade Association Home Page. `http://www.infinibandta.org/`.

11) Himeno Benchmark xp Home Page. `http://www.w3cic.riken.go.jp/HPC/HimenoBMT/index.html`.

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

13) The NAS Parallel Benchmarks Home Page. `http://www.nas.nasa.gov/Software/NPB/`.

14) HPL Algorithm. `http://www.netlib.org/benchmark/hpl/algorithm.html`.

15) NETGEAR Home Page. `http://www.netgear.com/`.