ハイパフォーマンス
コンピューティング 91−10
（2002．8．21）

地球シミュレータ上での流体コードのスケーラビリティ評価

宇野篤也[†] 板倉憲一[†] 横川三津夫[☆,††]
石原卓[†††] 金田行雄[†††]

SMP クラスタでは，MPI と共有メモリプログラミングを組み合わせたハイブリッドプログラミングと，MPI のみを用いて並列化するフラットプログラミングを行うことができる．ハイブリッドプログラミングは，フラットプログラミングと比較して幾つかの点で有利であると思われるが，フラットプログラミングを越える性能を得ることは難しいとされている．本研究では，SMP クラスタである地球シミュレータ上で，3 次元 FFT を用いた一様等方性乱流シミュレーションコードについて，ハイブリッドプログラミングとフラットプログラミングを行い，そのスケーラビリティ評価を行った．その結果，地球シミュレータ上ではハイブリッドプログラミングはフラットプログラミングよりも高い性能を得ることが可能であることが分かった．また，地球シミュレータでの特性を考慮した性能チューニングのポイントについて考察する．

Scalability Evaluation of Direct Numerical Simulation on Earth Simulator

Atsuya Uno,[†] Ken'ichi Itakura,[†] Mitsuo Yokokawa,[☆,††]
Takashi Ishihara[†††] and Yukio Kaneda[†††]

There are two programming models on the shared-memory architecure. One, called the flat programming, is using MPI, and the other, called the hybrid programming , is using MPI and shared-memory models simultaneously. In general, it is difficult that the hybrid programming outperforms the flat programming.

In this study, we evaluated a scalability of large-scale direct numerical simulations of the Navier-Stokes equations on the Earth Simulator. As a result, the hybrid programming could outpeform the flat programming on the Earth Simulator. Also, we discuss the tuning strategies to obtain higher performance on the Earth Simulator.

1. はじめに

地球シミュレータ[1)2)]は，大気大循環シミュレーションなど地球変動研究に代表される超大規模科学技術計算向けプラットフォームとして開発された超高速並列計算機で，640 台の計算ノードが単段クロスバネットワークで接続された大規模分散メモリ型並列計算機である．地球シミュレータは平成 14 年 2 月末に完成し，同年 3 月より運用が開始されている．

地球シミュレータのアーキテクチャは SMP クラスタである．このようなプラットフォームでは，MPI のみを用いたフラットプログラミングと MPI と共有メモリプログラミングを組み合わせたハイブリッドプログラミングが可能である．特に，ハイブリッドプログラミングはフラットプログラミングに比べ，通信回数やオーバーヘッド等の点で有利であると考えられるが，ハイブリッドプログラミングがフラットプログラミングの性能を越えたという研究結果はほとんど報告されていない．

本研究では，地球シミュレータ上に一様等方性乱流の直接数値シミュレーション (DNS) を行うプログラムをハイブリッドプログラミングとフラットプログラミングの両方で実装し，スケーラビリティの評価を行った．また，その結果から地球シミュレータでの性能チューニングのポイントについて考察を行った．

2. 地球シミュレータのアーキテクチャ

地球シミュレータの構成を図 1 に示す．地球シミュレータは 640 台の計算ノード (PN:Processor Node) を単段のクロスバネットワークで結合したメモリ分散

† 海洋科学技術センター 地球シミュレータセンター
Japan Marine Science and Technology Center
†† 日本原子力研究所
Japan Atomic Energy Research Institute
☆ 現在，産業技術総合研究所 グリッド研究センター
Presently with National Institute of Advanced Industrial Science and Technology
††† 名古屋大学大学院工学研究科
Nagoya University

型並列計算機である．PNはピーク性能 8Gflop/s の計算用ベクトルプロセッサ (AP:Arithmetic Processor) 8台，16GB の主記憶ユニット (MMU:Main Memory Unit)，リモートアクセス制御装置 (RCU:Remote Access Control Unit) 及び入出力プロセッサ (IOP:I/O Processor) から構成されている．地球シミュレータ全体では AP 5120 台，ピーク性能 40Tflop/s，主記憶 10TB となる．

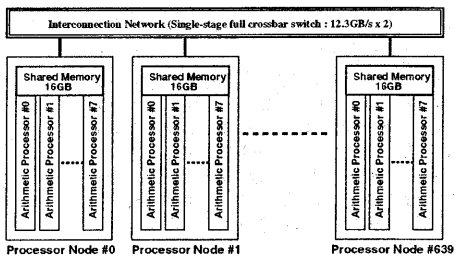

図 1　地球シミュレータの構成

AP は，ベクトル処理部 (VU:Vector Unit)，スカラ処理部 (SU:Scalar Unit)，メモリアクセス制御部が 1 つの LSI 上に実装されており，クロック周波数 500MHz (一部 1GHz) で動作する．VU は，6 種類 (加算，乗算，除算，論理，ビット列論理，ロード/ストア) のベクトル演算器と 72 個のベクトルレジスタからなるベクトル演算器セット 8 個で構成され，最大 8Gflop/s の性能を持つ．MMU は 2048 バンク構成で，各々の AP と MMU 間のバンド幅は 32GB/s，1PN で 256GB/s のバンド幅を実現している (図 2)．

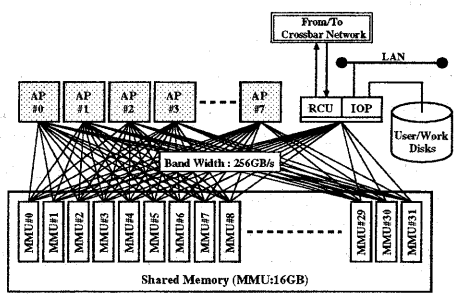

図 2　計算ノード (PN) の構成

RCU は，クロスバネットワークに直接接続され，クロスバによる双方向通信と AP とを独立に動作させることができる．データ転送機能では，同期型転送と非同期型転送がサポートされており，主記憶上の連続領域のデータを転送するブロック転送の他に，非同期型転送としてスライド付きベクトル転送，リストベクトル転送，3 ディスタンス転送がサポートされている．データ転送性能は，レイテンシがソフトウェアのオーバーヘッド込みで 約 1.2μs，理論最大スループットは送受信とも 12.3GB/s である．実際に PN 間の MPI 転送性能を評価した結果，MPI_PUT と MPI_GET による ping 転送において，約 11.63GB/s のスループットであった[3]．

地球シミュレータ上の並列プログラミングでは，AP 内ベクトル処理，PN 内共有メモリ並列処理，PN 間分散メモリ並列処理の 3 階層の並列プログラミング環境を利用でききる．PN 間の分散メモリ並列は Fortran90 または C をベースに MPI-1/MPI-2 による並列化を行う．PN 内共有メモリの並列プログラミングでは，コンパイラによる自動並列化や OpenMP を利用し，マイクロタスクによる並列実行を行う．マイクロタスクとは，地球シミュレータの並列実行形式であり，ループ文をコンパイラによりタスク分割し，ハードウェアが提供するタスク間の同期機構を利用して高速に実行する方式である．

地球シミュレータでは MPI だけを用いたフラットプログラミングと，MPI とマイクロタスクを組み合わせたハイブリッドプログラミングの 2 通りのプログラミングが可能である．SMP クラスタ等ではハイブリッドプログラミングはフラットプログラミングを上回る性能を得にくいと言われているが，地球シミュレータ上ではフラットプログラミングとハイブリッドプログラミングには大きな差は無く，条件によってはハイブリッドプログラミングの方が性能がでる場合もあることが報告されている[4]．本研究では，フラットプログラミングで作成したコードとハイブリッドプログラミングで作成したコードの両方を対象とし，そのスケーラビリティを評価した．

3. 一様等方性乱流シミュレーション

3.1 概　　要

一様等方性乱流シミュレーション行うため Trans7 が開発されている．このコードは，ナビエ・ストークス方程式の直接数値シミュレーション (DNS) をフーリエスペクトル法を用いて行う．

3 次元立方領域 $\Omega = [0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$ を基本周期領域として，単位密度の非圧縮性流体の運動を考えると，流体の運動は外力のある 3 次元ナビエ・ストーク方程式と連続の式で表すことができる．

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\nabla p + \nu \Delta u + f \qquad (1)$$

$$\nabla \cdot u = 0 \qquad (2)$$

ここで，$u = (u_1, u_2, u_3)$ は速度，p は圧力，ν は動粘性係数，f は $\nabla \cdot f = 0$ を満たす外力である．Trans7 では，式 (1) の非線形項を渦度形式で表現した方程式 (3) を使用している[5]．

$$\frac{\partial u}{\partial t} = u \times \omega - \nabla \Pi + \nu \nabla^2 + f \qquad (3)$$

ここで，$\omega = \mathrm{rot}u = (w_1, w_2, w_3)$ は渦度，$\Pi = p + \frac{1}{2}u^2$ である．

この NS 方程式を離散化する方法として，Trans7 ではスペクトル法を用いている．ここでは，周期境界条件に対応するために 3 次元フーリエ級数展開を用いた．u 及び Π をフーリエ展開した式は，

$$u(x_j, t) = \sum_{\boldsymbol{k}} \tilde{u}_{\boldsymbol{k}}(t) e^{i\boldsymbol{k}\cdot x_j} \tag{4}$$

$$\Pi(x_j, t) = \sum_{\boldsymbol{k}} \tilde{\Pi}_{\boldsymbol{k}}(t) e^{i\boldsymbol{k}\cdot x_j} \tag{5}$$

である．ここで，物理空間の各座標に対し，すべて同じ項数 N で離散化すると，$x_j = \frac{2\pi}{N}(j_1, j_2, j_3), j_\alpha = 0, 1, \cdots, N-1$ である．また，$\boldsymbol{k} = (k_1, k_2, k_3)$ はそれぞれの座標のフーリエ成分のベクトルであり，$-\frac{N}{2} \leq k_\alpha \leq \frac{N}{2} - 1$ の整数である．i は虚数単位 $\sqrt{-1}$, また $\tilde{u}_{\boldsymbol{k}}(t)$ などのフーリエ係数は，

$$\tilde{u}_{\boldsymbol{k}}(t) = \frac{1}{N^3} \sum_{\boldsymbol{j}} u(x_{\boldsymbol{k}}, t) e^{-\boldsymbol{k}\cdot x\boldsymbol{j}} \tag{6}$$

で表される．式 (4)，(5) を式 (2)，(3) に代入し整理すると，次の式が得られる．

$$\left(\frac{d}{dt} + \nu|\boldsymbol{k}|^2\right) \tilde{u}_{\boldsymbol{k}} = \tilde{s}_{\boldsymbol{k}} - \boldsymbol{k}\frac{(\boldsymbol{k}\cdot\tilde{s}_{\boldsymbol{k}})}{|\boldsymbol{k}|^2} + \tilde{f}_{\boldsymbol{k}} \tag{7}$$

$$\tilde{s}_{\boldsymbol{k}} = -\widetilde{(u \times \omega)}_{\boldsymbol{k}} \tag{8}$$

非線形項 $\tilde{s}_{\boldsymbol{k}}$ の計算には擬スペクトル法が有効であるが[5]，擬スペクトル法を用いた非線形項の計算ではエイリアジング誤差が生じるため，その誤差を除去する必要がある．Trans7 では，$|\boldsymbol{k}| < \frac{\sqrt{2}N}{3}$ のモードに対するエイジアリング誤差を完全に取り除く phase shift を 2 回実行し，畳み込み演算の際には 3 次元実 FFT を用いる方式を採用している．なお，式 (7) の時間積分には 4 次精度の 4 段ルンゲ・クッタ法を用いた[6]．

3.2 並列化手法

地球シミュレータ上では第 2 節で述べたように，PN 間分散メモリ並列処理，PN 内共有メモリ並列処理，AP 内ベクトル処理の 3 階層の並列化が可能である．

Trans7 では PN 間の並列化にあたり，スペクトル空間のフーリエ係数を k_3 方向に，物理空間の変数を y 方向に分割する領域分割法を用いた (図 3)．これにより，必要となるデータ転置は，3 次元実 FFT において z 方向に 1 次元 FFT を適用する部分と y 方向に 1 次元 FFT を適用する部分の中間で行う必要がある．PN 間の通信は MPI ライブラリを用いて実現している．データ転置では，全ての MPI プロセス間同士でのデータ交換が必要になる．総 MPI プロセス数を n_{mpi} とすると，1 回の 3 次元実 FFT において，ひとつの MPI プロセスから他の分割領域の計算を担当する $(n_{mpi} - 1)$ 個のうちの一つの MPI プロセスに送られるデータ個数は $N^3/2(n_{mpi})^2$ 個である．各 MPI プロセスは，$N^3/2(n_{mpi})^2$ 個のデータを $(n_{mpi} - 1)$ 回転送し，他の $(n_{mpi} - 1)$ 個の MPI プロセスからデータを受け取ってデータ転置が終了する．コードでは転送前にバリア同期をとった後，MPI_Put でデータを送信し，MPI_Win_Fence を実行している．

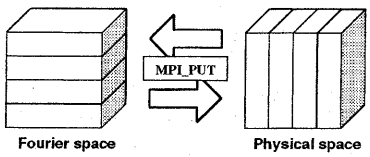

図 3 PN 間並列化 (FFT におけるデータ転置)

PN 内共有メモリの並列化には，コンパイラによる自動並列化を用いた．Trans7 のコード中の特定の最外側の do ループを指示行を追加することでタスク分割し，並列実行を行う．また，ループの最内側をベクトル処理の対象とし，バンクコンフリクトを避けるため，配列の第一要素数を奇数個に設定した (図 4)．

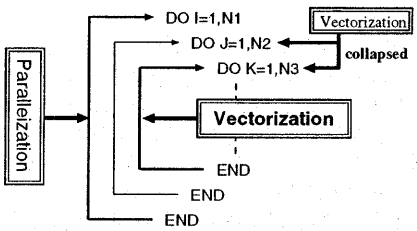

図 4 PN 内並列化

ただし，地球シミュレータのベクトルレジスタ長は固定の 256 要素 (8 バイト長) なので，問題サイズによっては十分な性能を得ることができない場合がある．

4. Trans7 のスケーラビリティ評価

表 1 に示す条件下で，ハイブリッドプログラミングとフラットプログラミングの両方についてシミュレーションを行う．フラットプログラミングでは，3.2 節で述べた並列化のうち，コンパイラの自動並列化機能をオフにしてコンパイルする．評価指数として，ハードウェアモニタによる 100 ステップ当たりの実行時間を用いる．

4.1 単一ノード内での並列性能

単一ノード内において，問題サイズが 32^3, 64^3, 128^3, 256^3 の 4 通りの場合の並列性能について計測した．メモリ容量の関係から単一ノードでは問題サイズは 256^3 までしか計算できない．

ハイブリッドプログラミングとフラットプログラミ

表 2 マイクロタスクのオーバーヘッド

計測条件	全体 (sec)	転置 (sec)	cache miss (sec)	演算時間 (sec)	オーバーヘッド (sec) と割合
Flat (2 MPI processes)	2.5150	0.1500	0.0807	2.2843	
Hybrid (2 CPU)	3.0720	0.1490	0.2642	2.6588	0.3745 (12.2%)
Flat (4 MPI processes)	1.3650	0.1485	0.0703	1.1463	
Hybrid (4 CPU)	1.9520	0.1190	0.2603	1.5727	0.4265 (21.8%)
Flat (8 MPI processes)	0.8518	0.1880	0.0786	0.5852	
Hybrid (8 CPU)	1.3910	0.1060	0.2589	1.0261	0.4409 (31.7%)

表 1 シミュレーション条件

ノード数	1 ～ 256
問題サイズ	32^3 ～ 512^3
MPI プロセス数	1 ～ 256
動粘度	0.00011
刻み幅	0.0004
タイムステップ数	100

ングそれぞれについて，ノード内の使用 AP(CPU) 数または MPI プロセス数が 1,2,4,8 の場合それぞれについて計測した．図 5 に計算時間を，図 6 に速度向上率をそれぞれ示す．速度向上率の計算では，1MPI プロセスで実行した場合の実行時間を基準とした．

問題サイズが 32^3 の場合，ハイブリッドプログラミングの方が，同じ AP を使用するフラットプログラミングの場合よりも性能が大きく劣っている．Trans7 での計算時間は，大別すると 1) 転置処理 (プロセス間通信とメモリコピー) 2) キャッシュミス (バンクコンフリクトを含む) 3)FFT などの演算時間 (ハイブリッドプログラミングの場合，マイクロタスクのオーバーヘッドも含まれる) に分けることができる．

問題サイズが 32^3 の場合の各時間は表 2 のようになっている．転置処理の時間及びキャッシュミスは，ハードウェアモニタから求めた．マイクロタスクによるオーバーヘッドは，100 ステップで約 0.414 秒である．問題サイズ 32^3 を 100 ステップ計算した場合，30,000 回マイクロタスクが起動されるので 一回あたりの起動コストは約 13μ 秒となる．問題サイズが 32^3 の場合には全体の計算時間が短いため，マイクロタスクのオーバーヘッドの影響が顕著に現れているが，問題サイズが 256^3 になるとオーバーヘッドの割合は 1%未満になりその影響はほとんどない．

問題サイズが 256^3 の場合，1AP での実行性能が約 4.55Gflop/s でピークの実行性能の約 56.9%，フラットプログラミングの 8MPI プロセスの場合で約 35.3Gflop/s(ピーク性能に対して約 55.2%)，ハイブリッドプログラミング 8CPU の場合で約 34.3Gflop/s(ピーク性能に対して約 53.5%) という結果が得られた．速度向上率はフラットプログラミングで約 7.8 倍，ハイブリッドプログラミングで約 7.5 倍となっており，ノード内並列処理では十分なスケーラビリティを達成していると言える．

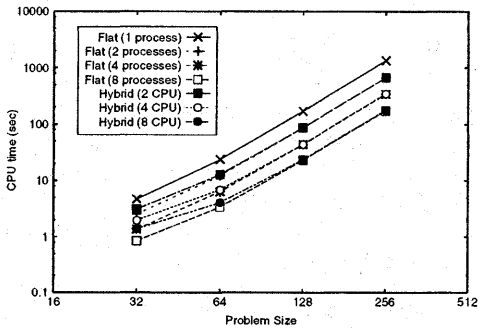

図 5 単一ノード内での計算時間

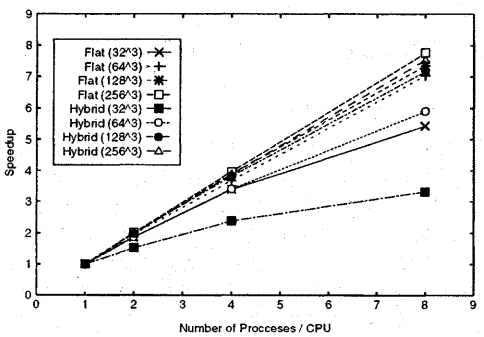

図 6 単一ノード内での速度向上率

4.2 マルチノード上での並列性能

問題サイズが 128^3, 256^3, 512^3 の 3 通りについて，ハイブリッドプログラミングとフラットプログラミングの両方でノード数とノード内での使用 AP 数を変化させた時の 100 ステップの実行時間を測定した．速度向上率の計算では，問題サイズが 128^3,256^3 の場合，1 ノード 1MPI プロセスで実行した場合の実行時間を基準とした．問題サイズが 512^3 の場合は，全体で約 26GB のメモリ量を使用し，ノードあたりのメモリ量の制限から 4 ノード以上ないと動作しないため，4 ノード 1MPI プロセスで実行した場合の実行時間を基準とした．Trans7 ではノード間の並列化をスペクトル空間の k_3 方向で分割しており，実 FFT の複素共役性を利用してフーリエ項数の半分で表現しているので，並列化では，最大で格子数の半分でしか分割で

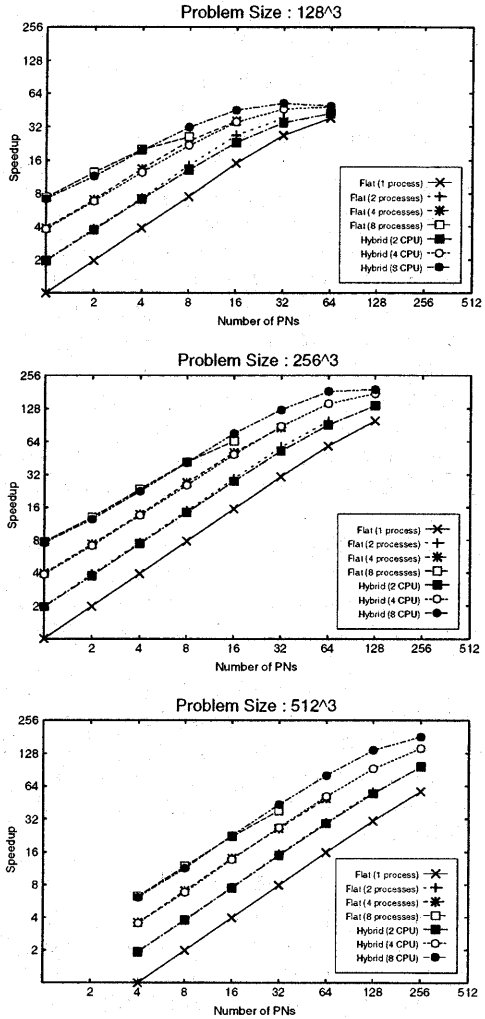

図 7　マルチノードでの速度向上率

きない．そのため，実行時の MPI プロセス数に上限があり，例えば 問題サイズが 512^3 の場合，256MPI プロセスが上限となる．図 7 に速度向上率を示す．

図 7 から分かるように，ノード間並列においても十分なスケーラビリティが得られている．また，フラットプログラミングとハイブリッドプログラミングの性能差は小さく，問題サイズが大きくなるにつれ，ハイブリッドプログラミングの方が性能が良いことがわかる．これは，フラットプログラミングの場合，対象問題の分割数が同一ノード数でのハイブリッドプログラミングと比べて多くなり，1 プロセスあたりの担当計算領域が小さくなることが原因と思われる．演算量に関してはハイブリッドプログラミングもフラットプログラミングもほぼ同じスケーラビリティがあると考えられる．4.1 節で述べたように，ハイブリッドプログラミングはフラットプログラミングに比べてマイクロタスクのオーバーヘッドやキャッシュミスが大きいが，その差は計算時間全体と比較すると小さなものであり，ある程度の大きさの問題サイズであれば全体の性能差は出にくい．一方，1 プロセスあたりの通信データサイズは $N^3/2(n_{mpi})^2 \times (n_{mpi}-1) \times 8$ バイトであり，問題サイズに関係なく，プロセス数に反比例して小さくなる．地球シミュレータのノード間通信性能は図 8 に示すように通信データサイズにより大きく変化する．そのため，ハイブリッドプログラミングとフラットプログラミングでの通信データサイズの差による通信性能の差が，マイクロタスクのオーバーヘッドやキャッシュミスに起因する性能差を上回り，結果としてハイブリッドプログラミングの方が性能が出るものと考えられる．

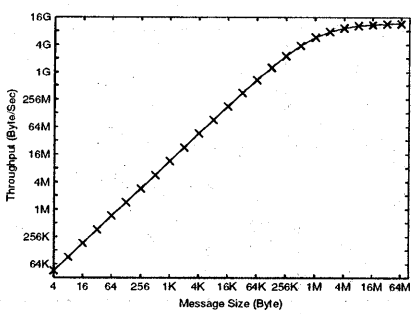

図 8　MPI_Put によるノード間通信性能

例えば問題サイズが 256^3 の場合，16 ノードのフラットプログラミングでの実効性能が約 296.4Gflop/s でピーク性能の約 28.9%，16 ノードのハイブリッドプログラミングの場合で約 346.9Gflop/s でピーク性能の約 33.9%である．フラットプログラミングの実効性能は 1 ノード 1 プロセスの場合の約半分になっているが，ハイブリッドプログラミングの場合は約 60%である．全体の計算時間のうち転置処理がが占める時間の割合は図 9 のようになっている．16 ノードの時のフラットプログラミングでは全体の約半分が転置処理で占められているのに対し，ハイブリッドプログラミングでは約 32%である．この通信データサイズによる通信性能の差が，フラットプログラミングとハイブリッドプログラミングの性能差に現れている．

これらのことから，地球シミュレータでは一般的な SMP クラスタとは違い，ハイブリッドプログラミングとフラットプログラミングの性能には大きな差は無く，特にプロセス数に反比例して通信データサイズが決まるような場合には，ハイブリッドプログラミングの性能がフラットプログラミングの性能を越えることが可能となる．しかし，計算モデルが小さく，並列化によって平均ベクトル長が短くなってしまう場合や，

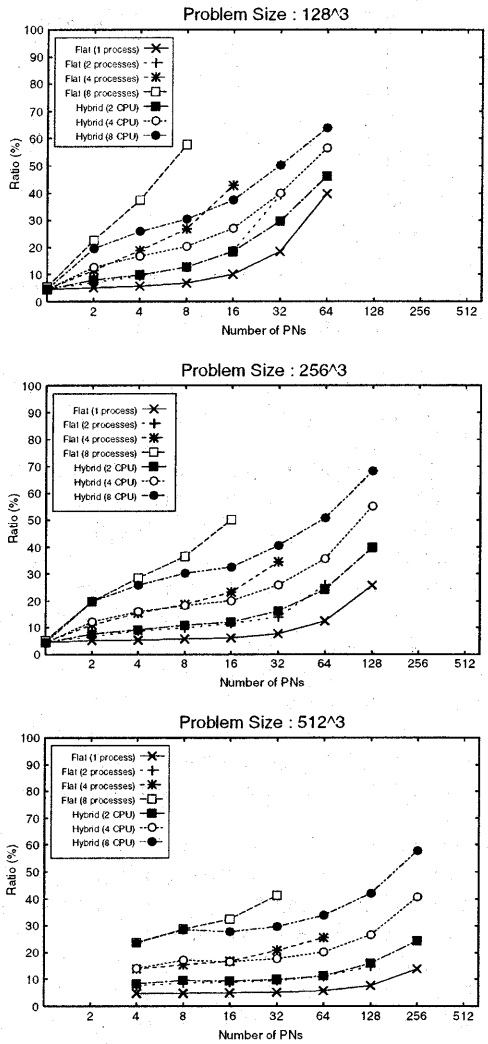

図 9　マルチノードで転置処理が占める割合

通信データサイズが小さく，ハイブリッドプログラミングとフラットプログラミングで通信時間に大きな差が生じないような場合にはフラットプログラミングの方が性能を出しやすいであろう．Trans7 では，問題サイズが 256^3 以上の場合，平均ベクトル長は約 256 である．これ未満の問題サイズでは平均ベクトル長は短くなっており，十分な性能がでない原因の一つになっている．

以上のことを考慮すると，大規模科学技術計算を主な計算対象としている地球シミュレータでは，ハイブリッドプログラミングが適したプログラミングモデルであり，1回あたりの通信データサイズを大きくし，平均ベクトル長を 256 に近づけることが地球シミュレータにおけるチューニングポイントであると言える．

5. ま と め

本研究では，地球シミュレータ向けに開発された一様等方性乱流の DNS を行う Trans7 を例に，ハイブリッドプログラミングとフラットプログラミングそれぞれに基づいたコードのスケーラビリティの評価を行った．一般に SMP クラスタの場合，ハイブリッドプログラミングではフラットプログラミングの性能を越えることはないと言われている．しかし，地球シミュレータの場合，そのハードウェア機構からフラットプログラミングとハイブリッドプログラミングとの性能差は小さく，地球シミュレータが対象とするような大規模科学技術計算では，ハイブリッドプログラミングの方が性能を出しやすいことが分かった．また，性能を十分に出すためにはノード内外のアーキテクチャの違いを考えながらプログラミングをしなければならないフラットプログラミングに比べ，ノード内はコンパイラの自動並列化にまかせることのできるハイブリッドプログラミングの方が全体としてプログラミングのコストを低く押えることができ，地球シミュレータにマッチしたプログラミングモデルであるといえる．

謝辞

本報告を発表する機会を与えてくださった佐藤 哲也 地球シミュレータセンター長に感謝致します．また，本研究を進めるにあたり，Trans7 の並列化に御協力頂いた NEC 情報システムズの斎藤 実氏及び，日頃から御討論頂く地球シミュレータセンターの諸氏に感謝致します．

参 考 文 献

1) 谷 啓二, 横川 三津夫：地球シミュレータ計画, 情報処理, Vol 41, No.3, pp.249–254 (2000).
2) 横川 三津夫, 谷 啓二：地球シミュレータ計画, 情報処理, Vol 41, No.4, pp.369–374 (2000).
3) Hitoshi Uehara, Masanori Tamura, and Mitsuo Yokokawa : *An MPI Benchmark Program Library and Its Application to the Earth Simulator*, ISHPC 2002, LNCS 2327, pp.219 – 230(2002).
4) 板倉 憲一, 宇野 篤也, 上原 均, 斎藤 実, 横川 三津夫：地球シミュレータ上のハイブリッドプログラミングの性能評価, HPC 90–4, pp.19 – 24 May (2002).
5) T.Ishihara, Y.Yamazaki, and Y.Kaneda: *Statistics of small-scale structure of homogeneous isotropic turbulence*, Proc. of the IUTAM Symposium on Geometory and Statistics of Turbulence, PP.133 – 138 (2001).
6) 横川 三津夫, 斎藤 実, 石原 卓, 金田 行雄：地球シミュレータ上の一様等方性乱流シミュレーション, HPCS2002, pp.125 – 131 January (2002).