情報処理学会論文誌　コンピューティングシステム　Vol.6　No.3　22–30 (Sep. 2013)

スーパーコンピュータ「京」における地震動シミュレーションコードの高性能化

井上 俊介[1,a)]　堤 重信[2]　前田 拓人[3]　南 一生[1]

受付日 2012年12月21日, 採録日 2013年2月9日

概要：理化学研究所では，スーパーコンピュータ「京」の高性能化を目的とし，6 本の重点アプリケーションを選定し，高性能化，高並列化を進めてきた．うち地球科学の分野から選択された地震動シミュレーションコードである Seism3D については，比較的高い Byte/Flop 値を要求する演算と，隣接プロセス間のみの通信という特徴があげられる．よって，Seism3D の高性能化，高並列化のポイントとして，メモリバンド幅を最大限に生かすこと，キャッシュの効率的な利用をすること，6 次元メッシュ上での最適な隣接通信を実現すること，に絞られる．我々はコードの持つ要求 Byte/Flop から求まるピーク比性能の推定を実施し，詳細プロファイラ機能を活用することにより問題点を把握し，実測，チューニングを実施し，CPU 単体性能向上策の検証と通信部の検証を進めた結果，82,944 並列で理論ピーク比 17.9%（1.9 PFLOPS）に達したため，本稿で報告する．

キーワード：スーパーコンピュータ「京」，地震動シミュレーションコード，性能評価，性能最適化

Performance Optimization of Seismic Wave Simulation Code on the K computer

Shunsuke Inoue[1,a)]　Shigenobu Tsutsumi[2]　Takuto Maeda[3]　Kazuo Minami[1]

Received: December 21, 2012, Accepted: February 9, 2013

Abstract: In order to optimize performance of the K computer, we selected six applications from various scientific fields. We optimized CPU performance and massively parallelization to them. Seism3D which was selected from earth science field is seismic wave simulation code. It has calculation parts which demands high Byte/Flop and communication parts between neighborhood processes. So optimization points are using enough memory bandwidth, using cache effectively and realization of optimal neighborhood communications on six-dimensional mesh/torus network. We estimated theoretical performance from required Byte/Flop of code and utilized advanced profiler to have a clear grasp of bottle neck. As a result, we achieved 17.9% per peak performance by using 82,944 cpus.

Keywords: K computer, seismic wave simulation code, performance evaluation, performance optimization

1. はじめに

2011 年 6 月と 11 月に TOP500 ランキング [1] で 2 期連続 1 位となったスーパーコンピュータ「京」（以下,「京」と略す）は，2012 年 9 月に共用開始の運びとなった．演算性能だけではなく 2011 年の HPC Challenge [2] で 4 部門すべてで 1 位を獲得するなど，様々な分野のアプリケーションの高性能化が期待されている．理化学研究所計算科学研究機構では，利用者に有用となる「京」の高性能化技術を集約すべく，6 本の重点アプリケーションを選定し，「京」の計算性能最適化手法を継続的に検証してきた．

本稿では，こうして選定されたアプリケーションの 1 つである地震動シミュレーションコード Seism3D において，

[1] 独立行政法人理化学研究所計算科学研究機構
RIKEN Advanced Institute for Computational Science, Kobe, Hyogo 650–0047, Japan
[2] 株式会社富士通九州システムズ
Fujitsu Kyushu Systems Ltd., Fukuoka 814–8589, Japan
[3] 東京大学地震研究所
Earthquake Research Institute, The University of Tokyo, Bunkyo, Tokyo 113–0032, Japan
[a)] inoue.shunsuke@riken.jp

ハードウェア性能との比較を軸とした高性能化，高並列化ならびにその成果について報告する．Seism3D に関しては，我々はこれまでにも主要な計算カーネルを題材とした性能予測手法および性能の評価を実施してきた [3]. 本稿ではアプリケーション全体の高性能化のプロセスに主眼を置き，オリジナルコードを用いた「京」での性能評価およびツールを使った問題点の抽出法についてより詳細に述べ，さらに未評価であった計算/通信ルーチンについても考察を加え，「京」の全ノードにおける測定結果について論じる．なお，個々の測定値においては，本稿においても開発段階でのシステムを利用した結果となっているため，過去の研究結果 [3] とは測定値が異なるケースも散見されるが，提案する性能評価手法およびチューニング手法は「京」において汎用的である．

以降 2 章で Seism3D の概要について，3 章で「京」の概要について述べる．4 章でオリジナルコードの測定結果に基づきチューニングの課題を明確にし，5 章でツールを用いた計算カーネルの性能評価方法について述べる．6 章で計算カーネルの実測および性能向上手法を検討する．7 章で通信部の評価をし，8 章で「京」の全ノードでの測定結果を報告する．

2. アプリケーション概要

Seism3D は粘弾性体運動方程式を Staggered grid 差分法を用いて空間 4 次，時間 2 次精度で陽的に解く MPI/OpenMP ハイブリッド並列地震動シミュレーションコードである [4]．近年，従来の運動方程式に重力項を加味することにより，地震，地殻変動，津波の連成計算を可能にし [5],「京」でのより高精度なシミュレーションが期待されている．Seism3D は以下の運動方程式および粘弾性体の構成方程式を差分法により数値的に解く．

$$
\begin{aligned}
\rho\frac{\partial v_x}{\partial t} &= \frac{\partial \sigma^D_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} - \rho_w g_0 \frac{\partial \eta}{\partial x} \\
\rho\frac{\partial v_y}{\partial t} &= \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma^D_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} - \rho_w g_0 \frac{\partial \eta}{\partial y} \\
\rho\frac{\partial v_z}{\partial t} &= \frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma^D_{zz}}{\partial z} \\
\frac{\partial \sigma_{xx}}{\partial t} &= (\lambda+2\mu)\frac{\partial v_x}{\partial x} + \lambda\left(\frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right) \\
\frac{\partial \sigma_{yy}}{\partial t} &= (\lambda+2\mu)\frac{\partial v_y}{\partial y} + \lambda\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z}\right) \\
\frac{\partial \sigma_{zz}}{\partial t} &= (\lambda+2\mu)\frac{\partial v_z}{\partial z} + \lambda\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right) \\
\frac{\partial \sigma_{xy}}{\partial t} &= \mu\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) \\
\frac{\partial \sigma_{yz}}{\partial t} &= \mu\left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right) \\
\frac{\partial \sigma_{xz}}{\partial t} &= \mu\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right)
\end{aligned}
$$

ここで，v_i は地震動の速度場，σ_{ij} および σ^D_{ii} は応力テンソル，ρ は弾性体の質量密度，λ と μ は等方弾性体を特徴付ける係数（Lame の定数）であり，地球内部構成物質に依存する値を持つ．また，地震波の P 波・S 波速度の速さはこれらのパラメータと $V_p = \sqrt{(\lambda+2\mu)/\rho}$，$V_s = \sqrt{\mu/\rho}$ の関係にある．Seism3D では固体地球だけでなく海水も $V_s = 0$ の弾性体として取り扱うことで，地震動と海中音波・津波の統一的な扱いを可能にしている．

コードの実装は，上記速度場および応力場の微分方程式を交互に解くため，以下の処理を繰り返す．

（a）応力空間微分計算
（b）速度時間積分計算
（c）速度時間積分計算（境界部）
（d）速度袖通信
（e）速度空間微分計算
（f）応力時間積分計算
（g）応力時間積分計算（境界部）
（h）応力袖通信

演算部の特徴として，再利用性のないストリーム配列を多く必要とする $O(N)$ の演算が主体であることから要求 Byte/Flop が高いといえる．また，演算部によって求められた速度と応力を隣接領域に転送するだけであるため，通信部は隣接通信のみである．

3. 「京」の概要

CPU は新規に開発された富士通社製の SPARC64™ VIIIfx [6], [7], [8], [9] であり，1 チップ上に 8 コアの演算器を持つ．ピーク性能は 1 チップあたり 128 GFLOPS で，メモリバンド幅は理論値が 64 GB/s（0.5 B/F）である．浮動小数点レジスタは 256 本に拡張され，コンパイラによる命令スケジューリングが容易となっている．また SIMD 命令の導入による，ベクトル計算，マスク演算が可能となり，後者はコンパイラによりプログラム中の分岐での命令スケジューリングの向上につながる．さらにキャッシュにセクタキャッシュ機能が導入され，一部再利用性のあるデータを留め置くことが可能となり，メモリバンド幅を圧迫し，性能最適化にキャッシュの有効活用を必要とするアプリケーションでの効果を期待できる．スレッド並列についてはコア間の並列処理のための同期をとるためのハードウェアバリア機構を備えることで，複数コアでのスレッド効率が飛躍的に向上している．

これらの計算ノードは，Tofu と呼ばれる 6 次元メッシュ/トーラスネットワーク [10] で結合されている．バンド幅は各次元双方向 5 GB/s で接続されており，バイセクションバンド幅は 30 TB/s である．各計算ノードからは 10 本の接続が可能で，そのうち 6 本が 3 次元メッシュ-トーラスとして接続され，残りの 4 本で残りの 12 計算ノード（2 × 3 × 2 のメッシュ）の内部結合に用いられている．この構造によ

り，ユーザの視点で見ると，システムに収まる任意サイズの 3 次元トーラスを切り出すことが可能であり，さらに故障ノードがあっても 3 次元トーラスを確保するルートが切り出し可能であるという利点があり，利便性だけでなく信頼性も向上している．

システムは，1 枚のシステムボードに 4 個の CPU が搭載され，1 台の筐体には 24 枚のシステムボードが搭載されている．「京」ではこの筐体が 864 台設置される．ピーク性能 10.62 PFLOPS，全メモリ量 1.26 PiB である．

4. 課題の検討

4.1 測定パラメータおよび測定環境

Seism3D は，パラメータ指定により任意の地表面における地震動および津波のシミュレーションが可能である．今回，「京」の全ノードを用いた性能評価を実施するため，1,200 km × 800 km，深さ 200 km の地表モデルを想定した．結果として 1 メッシュの格子間隔は 50 m，ノード内は $(X, Y, Z) = (60, 80, 4000)$ のメッシュ規模を設定し，想定される大規模地震のシミュレーションに対応可能とした．また，「京」における測定環境を**表 1** に示す．

4.2 Tofu へのマッピング

Seism3D はコード開発当初は地震動のみの解析であったため，高並列化に向けた 3 次元分割モデルであった．しかし津波の連成解析も実施するように改修されたため，現在は垂直方向には分割のない 2 次元分割モデルとなっている．「京」はユーザビューとしては 3 次元トーラスであるため，2 次元モデルを 3 次元上にマッピングする際に隣接通信のコストがどう影響するかを評価する必要がある．しかし，先に述べた Tofu インタコネクトでは，3 次元，2 次元両分割モデルともに，隣接通信が 1 ホップであることが保障されている．したがって，隣接袖通信が主体である Seism3D では，マッピングを物理軸に合わせることにより，隣接通信の問題はほぼ解決すると考えられる．

4.3 オリジナルコードによる測定と課題の定義

「京」における高性能化に向けた具体的なアプローチを検討する．まずはチューニング前のコードに対し，設定したメッシュモデルを用いて各並列規模でウィークスケーリングによる測定を実施し，課題を洗い出す．

チューニング前のコードの TimeStep = 200 における測定結果を**図 1** に示す．16 プロセス（4 × 4），8 スレッドで，ピーク比は 11.8%であった．また，ウィークスケーリングによる評価のため，並列数の増加にともなう実行時間の増加がないことを確認する．水平方向にそれぞれ 2 倍ずつプロセス数を増やし，16,384 プロセス（128 × 128）までプロファイラを用いてサンプリングにより測定した結果を**表 2** に示す．

図 1 より，微分計算ルーチン（a），（e）は各プロセスともほぼ均等なコストであり，速度および応力の積分計算は構造内部と境界部の和がほぼ等しくなるため，アプリケーション全体としてロードインバランスは発生していない．表 2 では，サンプリングによる簡易測定のため，実行時間とピーク性能比の関係に若干の誤差が発生しているが，ウィークスケール時のプロセス増加にともなうオーバヘッドは発生していないことが確認できる．また，各ルーチンにおいても並列数の増加にともなった傾向の変化は観察されなかったため，高並列化の観点では大規模なチューニングを検討する必要はない．したがって，分析と性能向上策の検討は CPU 単体性能の向上を主とし，高度化の方針を以下と定める．ただし並列化に係る通信処理についても（2）に示す評価は実施する．

（1）演算部について，「京」の CPU 単体性能として十分であるか，より詳細に分析する．具体的には，コードを各サブルーチン，処理単位（カーネル）ごとに特性や処理データ量を精査し，そのカーネルが持つ要求 Byte/Flop を求

表 1 「京」における測定環境
Table 1 Measurement environment of K computer.

コンパイラ	Fujitsu Fortran　K-1.2.0-04
オプション	-Kfast,parallel,openmp,ocl

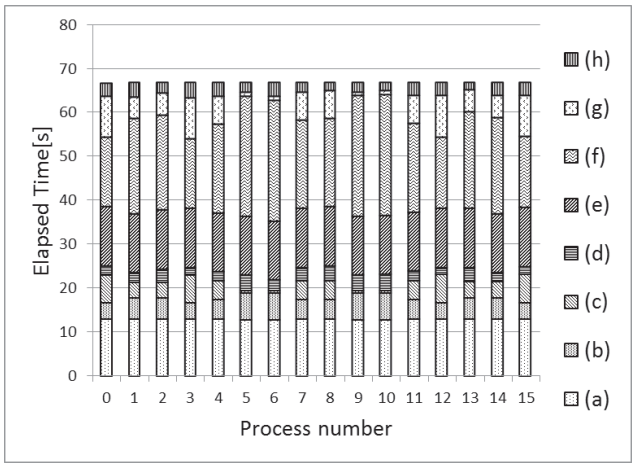

図 1 16 プロセスにおける測定結果
Fig. 1 Measurement result of 16 processes.

表 2 16,384 プロセスまでの測定結果
Table 2 Measurement result between 16 and 16,384 processes.

Number of process	Elapse（sec）	Peak ratio（%）	Efficiency
16（4x4）	68.8	11.8	1.00
64（8x8）	69.5	11.9	0.99
256（16x16）	71.3	11.7	0.96
1024（32x32）	69.1	12.2	0.99
4096（64x64）	69.2	12.2	0.99
16384（128x128）	69.8	12.1	0.98

め，そこから導かれる実効性能の推定値と実測データを比較する．推定値と実測データに著しい乖離があれば原因を探り対処する．さらに性能向上の余地があれば，性能向上策を評価，適用する．

(2) 隣接通信部はデータのパック/アンパック処理，通信を正確に測定するための同期処理を含んでおり，図 1 の結果はそれらのコストも含む結果となっている．この箇所を詳細に分析することで，「京」の通信性能として十分であるかを検証する．

5. 性能の推定および評価

5.1 評価基準値の設定

各演算部の性能が十分であるかを評価するため，まずは評価基準値を設定する．我々は，Seism3D がストリーム配列を多く必要とする要求 Byte/Flop が高いコードであることを考え，メモリ性能を測定する STREAM Benchmark [11] Triad の結果を用いて，評価基準値を設定することとした．「京」では，詳細プロファイラ機能 [12] が提供されており，SPARC64^{TM} VIIIfx に備わったハードウェアカウンタを用いることで CPU 単体性能の特徴や問題点を比較的容易に得ることができる．本機能を用いて，STREAM Benchmark Triad を 1 ノード 8 スレッドで測定した結果を**表 3** に示す．

本ベンチマークコードは，再利用性のない 3 つの倍精度配列の積和演算である．表 3 より，「京」におけるメモリ要求が高いループについて，以下の考察が可能であり，これらを Seism3D の CPU 単体性能を評価するうえでの基準値とする．

・「京」のキャッシュラインは 128 byte であるため，たとえば単精度 4 byte のデータでは，連続アクセスでは 32 要素のうち先頭の 1 要素だけがミスをする．すなわち，単精度では 1/32 = 3.125%，倍精度では 1/16 = 6.25%が連続アクセスにおける各キャッシュレベルでのミス率の基準値である．表 3 中の L1D ミス率および L2 ミス率はそれぞれのキャッシュレベルによるミス率を表している．本測定は倍精度演算のため，L1，L2 キャッシュミスとも基準値となり，これはストリーム配列のみのループにおける，キャッシュの有効利用性を見る指標となる．

・L1D ミス dm 率とは L1D ミス率のうち，load，store 命令のアクセスによるミス率である．L1D ミス hwpf，swpf 率とはそれぞれ hardware prefetch（hwpf），prefetch 命令（swpf）のアクセスによるミス率であり，これら 3 つが L1D ミス率の内訳となる．「京」では最内 16 ストリーム以下であればデフォルトでは swpf は発行されず，すべて hwpf で賄われる．また，L1D ミス hwpf，swpf 率が高いほど，prefetch が効果的に発行されていると判断する．本測定においてもこれがいえる．

・L2 スループットが 33.23 GB/s と観察されている．メモリスループットは，単位時間あたりの L2 キャッシュへのデータ供給および書き戻し（write back）の総量であるのに対し，L2 スループットは単位時間あたりの L1 キャッシュへのデータ供給量を表す．したがって，Stream Benchmark Triad のような単純にメモリからデータが供給されるプログラムでは，本来であればメモリスループットと L2 スループットは同量であるが，上記の理由により L2 スループットがメモリスループットより少なく観察される．L2 スループットは L2 キャッシュにおける再利用性が高くなると増大する傾向がある．

・「京」の理論メモリスループットは 1 ノード 64 GB/s に対し，実測で 44.33 GB/s である．測定により若干の誤差が発生するため，「京」の実効メモリスループットは 46 GB/s，メモリの実効 Byte/Flop 値は 0.36 と定義する．

表 3　STREAM Benchmark Triad の測定結果
Table 3　Measurement result of STREAM Benchmark Triad.

L1D ミス率	6.25%
L1D ミス dm 率	0.61%
L1D ミス hwpf 率	99.38%
L1D ミス swpf 率	0.00%
L2 ミス率	6.26%
L2 スループット	33.23GB/s
メモリスループット	44.33GB/s
Peak ratio	2.16%

```
do j = 1, NY
   do i = 1, NX
      do k = 3, NZ-1
            DZV (k,i,j) = (V(k,i,j)   -V(k-1,i,j))*R40z&
                        - (V(k+1,i,j)-V(k-2,i,j))*R41z
      end do
   end do
end do
```

図 2　Staggered grid 第 1 軸差分計算ループ
Fig. 2　The spatial derivatives calculation of Z direction.

5.2 性能推定と検証

一般的に，要求 Byte/Flop が高いコードは同時にメモリスループットで律速する．つまり演算時間はメモリスループットで決定される．「京」においても同様のことがいえるため，「京」の 1 ノードにおける本コードの実効性能は，要求 Byte/Flop とメモリの Byte/Flop の比によって推定が可能である．例として，本コードで頻繁に利用される Staggered grid 差分による微分項計算ループ（**図 2**）を考える．

具体的な性能推定手法と測定結果は過去の研究 [3] を参照されたいが，本ループは第 1 軸である K 軸が差分化されている場合であり，(NX, NY, NZ) = (60, 80, 4000) である

表 4　Staggered grid 第 1 軸差分計算ループの測定結果
Table 4　Measurement result of the spatial derivatives calculation of Z direction.

L1D ミス率	2.09%
L1D ミス dm 率	0.17%
L1D ミス hwpf 率	99.83%
L1D ミス swpf 率	0.00%
L2 ミス率	2.09%
L2 スループット	31.18GB/s
メモリスループット	46.74GB/s
Peak ratio	15.1%

本ループの推定性能値はピーク比 15%である，第 2 軸が差分となっているループの場合は 15%，第 3 軸が差分となっているループは 7.5%となる．5.1 節と同様に詳細プロファイラ機能を用いて第 1 軸差分計算ループを測定した結果を**表 4** に示す．

キャッシュミス率については，配列 V の隣接要素が L1 キャッシュ上で再利用されることにより，単精度の連続アクセスの基準値より低く測定される．またメモリスループットが 46.7 GB/s と基準値より高めに測定されていることから，ピーク比は推定値より若干良い傾向であるが，高精度で一致することが分かる．また，表 3 および表 4 で測定されたように，ミス率が連続アクセスの基準値に収まっていることが，推定性能に達しているかどうかの 1 つの指標となる．

Seism3D はどのルーチンにおいても，メモリ要求が高く，「京」の実効メモリ Byte/Flop である 0.36 を下回らない．したがって，本コードの「京」における演算部の高性能化は，以下の手順で進めることとした．

1) 各ルーチンにおいて，推定性能まで達しているか確認する．
2) 推定性能に達していない場合に，詳細プロファイラを用いた原因の調査とそれに対する改善策を検討する．
3) 推定性能に達している場合，さらなる性能向上手法を検討する．

6. CPU 単体性能の測定と性能向上手法の評価

6.1 （f），（g）応力時間積分部

本ルーチンは，応力テンソル計算のため，最内ループに用いられるストリーム数が多く存在することが特徴である．この場合，「京」の 1 次キャッシュは 2way であることを考慮すると，1 次キャッシュにおけるキャッシュ競合を回避させるチューニングが効果的である．

（g）の境界部の要求 Byte/Flop は 2.79 であり，推定性能値は 12.9%となる．オリジナルコードでは L1D ミス dm 率が 49.93%と高く，L1D ミス率も基準値より悪い 3.54%と測定された．5.1 節で示したように，「京」では，ストリーム

表 5　（g）応力時間積分部（境界部）の測定結果
Table 5　Measurement result of (g) calculation of stress by time integration (boundary parts).

	Original	Tuning
L1D ミス率	3.54%	2.71%
L1D ミス dm 率	49.93%	11.83%
L1D ミス hwpf 率	25.99%	88.17%
L1D ミス swpf 率	24.08%	0.00%
L2 ミス率	2.11%	1.97%
L2 スループット	42.32GB/s	35.79GB/s
メモリスループット	39.86GB/s	42.87GB/s
Peak ratio	8.80%	10.05%

配列が支配的なループにおいて，L1D ミス率が 3.125%（倍精度では 6.25%以上）かつ L1D ミス dm 率が 20%を超える場合に，キャッシュ競合が発生していると考える．これを改善するため，応力 3 成分を 1 つの配列に融合し，ループ分割を適用することにより，最内でアクセスするストリームの絶対数を減らすアプローチを採用した．結果として，L1D ミス dm 率が 11.83%となり，最内ストリーム数の削減により効率の良い hwpf が生成され，メモリスループットも 42.87 GB/s まで改善された．また，L2 スループットはオリジナルに比べて下がったが，L1 でのキャッシュ競合により発生していた不要な L2 アクセスが解消されたものと考える．オリジナルコードとチューニングコードの測定結果を**表 5** に示す．

同様に，要求 Byte/Flop が 1.44，推定性能が 25.0%である（g）応力時間積分部においてもキャッシュ競合が観察されたため，同様なチューニングを実施したところ，メモリスループットが 34.84 GB/s から 42.74 GB/s へ，ピーク性能比は 16.24%から 21.24%まで改善された．

6.2 （b），（c）速度時間積分部

推定性能値を算出する際に，コードから判別できないケースがある．**図 3** に示す（b）速度時間積分計算部においては，単精度の割り算が逆数近似計算のため，コード上は 1 演算でも演算数を 8 として算出する必要がある．結果として本ループの要求 Flop は 52 となる．対して，要求 Byte は配列 den の再利用性を考慮すると 4 byte × 18 となり，要求 B/F は 1.38 である．したがって，推定性能は 26%となる．対して，実測値は**表 6** に示すように 22.29%となった．

再利用性があるため，各キャッシュレベルでのミス率が基準値より低下しており，L2 スループットも向上している．最内のストリーム数が 18 と多く，そのために swpf が発行されている．メモリスループットも 45 GB/s を超える結果となり，観察される値に問題ないことを考慮すると，性能のボトルネックはメモリであり，本ループは性能の上限まで達していると判断した．ただし，最内のストリーム

```
do j = NY00, NY01
 do i = NX00, NX01
  do k = NZ00, NZ01
    ROX = 2.0_PN/( den(k,i,j) + den(k,I+1,J) )
    ROY = 2.0_PN/( den(k,i,j) + den(k,I,J+1) )
    ROZ = 2.0_PN/( den(k,i,j) + den(k+1,I,J) )
    Vx(k,i,j) = Vx(k,i,j) &
              + ( dxSxx(k,i,j)+dySxy(k,i,j)+dzSxz(k,i,j) )*ROX*DT
    Vy(k,i,j) = Vy(k,i,j) &
              + ( dxSxy(k,i,j)+dySyy(k,i,j)+dzSyz(k,i,j) )*ROY*DT
    Vz(k,i,j) = Vz(k,i,j) &
              + ( dxSxz(k,i,j)+dySyz(k,i,j)+dzSzz(k,i,j) )*ROZ*DT
    vx(k,i,j) = vx(k,i,j) &
              - grav(k,i,j) * dxEta(i,j) * dt * den_s(i,j) * ROX
    vy(k,i,j) = vy(k,i,j) &
              - grav(k,i,j) * dyEta(i,j) * dt * den_s(i,j) * ROY
  end do
 end do
end do
```

図 3　速度時間積分計算ループ

Fig. 3　The calculation loop of velocity by time integration.

表 6　(b) 速度時間積分部の測定結果

Table 6　Measurement result of (b) calculation of velocity by time integration.

L1D ミス率	2.87%
L1D ミス dm 率	12.90%
L1D ミス hwpf 率	87.01%
L1D ミス swpf 率	0.09%
L2 ミス率	2.38%
L2 スループット	45.57GB/s
メモリスループット	45.39GB/s
Peak ratio	22.29%

数が 18 であること，および過去のシステム環境による測定ではキャッシュ競合により 7.5%であったことを考慮すると，他の問題サイズにおいてキャッシュ競合の発生確率が高まる．本ルーチンもループ分割と配列融合によって，あらかじめリスクを回避しておく修正を実施した．

(c) 境界部においては，L1D ミス率が 3.36%，L1D ミス dm 率が 42.63%とキャッシュ競合が観察されたため，同様にループ分割および配列融合を実施することにより，メモリスループットが 39.1 GB/s から 42.8 GB/s に，ピーク性能比が 15.88%から 17.52%まで向上した．本処理部の推定性能は 19.9%であることおよび実測のメモリスループットを考慮すると，十分な性能といえる．

表 7　(a)，(e) 微分項部のオリジナルコード測定結果

Table 7　Measurement result of (a) (e) original code of derivative parts.

	応力微分項	速度微分項
L1D ミス率	2.96%	2.95%
L1D ミス dm 率	3.95%	4.21%
L1D ミス hwpf 率	96.05%	95.88%
L1D ミス swpf 率	0.00%	0.00%
L2 ミス率	2.19%	2.19%
L2 スループット	43.98GB/s	42.30GB/s
メモリスループット	43.32GB/s	41.81GB/s
Peak ratio	10.41%	10.05%

```
do j = 1, NY
  do i = 1, NX
    do k = 3, NZ-1
      DZV (k,i,j) = (V(k,i,j)   -V(k-1,i,j))*R42 &
                  - (V(k+1,i,j)-V(k-2,i,j))*R43
      DXV (k,i,j) = (V(k,i,j)   -V(k,i-1,j))*R40&
                  - (V(k,i+1,j)-V(k,i-2,j))*R41
    end do
  end do
end do
```

図 4　ループ融合

Fig. 4　Loop fusion.

6.3 (a)，(e) 微分項部

1 タイムステップあたり，速度微分項を求める staggered 差分計算が 9 回，応力微分項を求める staggered 差分計算が 9 回の計 18 回利用されるルーチンであり，全経過時間の 40%を占める．速度微分，応力微分とも 3 次元配列の第 1 軸，第 2 軸，第 3 軸が各軸方向で差分計算されるため，5.2 節の性能推定値を用いると，$(15 + 15 + 7.5)/3 = 12.5\%$が期待する推定性能値である．**表 7** にオリジナルコードの測定値を示す．

表 7 から，特にキャッシュまわりにおいて問題は観察できず，また測定されたメモリスループットから考慮すると大きな性能劣化要因はなく，ほぼ推定性能値に達しているといえる．

次に，本ルーチンにおける性能向上策を検討する．本ルーチンは，すべての軸においてメモリ要求が高く，特に第 3 軸差分計算は問題サイズによりキャッシュ上では再利用性が発生しないため，他の軸に比べてメモリ要求が 2 倍高くなっている．このメモリ要求を低減する手法として，以下 3 つの方法を採用した．

1)　ループ融合

図 4 は，第 1 軸と 2 軸のループ融合を実施した例である．右辺が 1 回のロードで，キャッシュ上で再利用できる

```
!$OMP DO SCHEDULE(static,1)
do j = 1, NY
   do i = 1, NX
      do k = 1, NZ
            DXV (k,i,j) = (V(k,i,j)    -V(k,i,j-1))*R40&
                        - (V(k,i,j+1)-V(k,i,j-2))*R41
      end do
   end do
end do
```

図 5　Cyclic 分割
Fig. 5　Cyclic distribution.

表 8　Staggered grid 第 3 軸差分計算ループの測定結果
Table 8　Measurement result of the spatial derivatives calculation of Y direction.

	Original	Tuning（cyclic）
L1D ミス率	3.13%	3.13%
L1D ミス dm 率	0.39%	8.95%
L1D ミス hwpf 率	99.61%	91.04%
L1D ミス swpf 率	0.00%	0.00%
L2 ミス率	3.14%	1.49%
L2 スループット	38.45GB/s	68.96GB/s
メモリスループット	46.33GB/s	46.84GB/s
Peak ratio	7.49%	13.45%

変数が 6 つに増え，メモリ要求が左辺 2 × 2，右辺 1，演算が 10 となり，要求 Byte/Flop が 2 となる．したがって推定性能は 18%となる．実測値は 17.4%となり，推定性能に近づく，効果の高い性能向上策であることが検証された．

2）　Cyclic 分割

オリジナルコードでは最外ループに対し自動スレッド並列を機能させており，第 3 軸差分計算ループではロードされるすべての変数に再利用性がない．しかし OpenMP による Cyclic 分割により，他スレッドがメモリからロードしたデータを，自スレッドが再利用可能になり，メモリ負荷の低減が狙える．これにより第 3 軸差分計算ループのメモリ要求が第 1，2 軸差分と同等程度になることが期待でき，推定性能はオリジナルの 7.5%から 15%となる．**図 5** のとおり，指示行で Cyclic 分割を指定することにより，Peak 比が 13.6%まで上昇した．測定結果を**表 8** に示すが，L2 キャッシュの再利用性が高まるため，L2 ミス率が低く，L2 スループットが高めに出ていることが観察される．またミス率やメモリスループットに特に問題は観察されないため，（b）と同様に性能のボトルネックはメモリとなり，性能の上限値まで達していると判断する．1），2）を組み合わせた手法の性能推定および実測に関しては，過去の研究 [3] を参照されたい．

表 9　各軸差分ループの XFILL 指示行の測定結果
Table 9　Measurement result of the derivative loops with XFILL directives.

	第 1 軸	第 2 軸	第 3 軸
L1D ミス率	2.06%	3.27%	3.12%
L1D ミス dm 率	4.70%	9.10%	3.01%
L1D ミス hwpf 率	50.30%	75.80%	79.85%
L1D ミス swpf 率	44.98%	15.09%	17.13%
L2 ミス率	1.08%	0.68%	2.52%
L2 スループット	42.88GB/s	89.62GB/s	45.85GB/s
メモリスループット	43.42GB/s	34.69GB/s	45.85GB/s
Peak ratio	21.22%	16.13%	8.93%

3）　XFILL 指示行

「京」では，XFILL と呼ばれる高速ストア機能が提供されている．本機能は連続かつ定義参照のないストア配列にのみ有効である．具体的な例としては，図 2 のように，左辺で定義される配列 DZV が右辺で参照されず，かつ連続アクセスとなるループに対して指示することができる．本機能を用いることにより，書き込み用のラインをキャッシュ上に確保しストア命令時のリードアクセスがキャッシュヒットするため，左辺のメモリ要求が半分になり，要求 Byte/Flop を下げることが可能である．結果として，オリジナルコードでは第 1, 2 軸差分 15%，第 3 軸差分 7.5%だった推定性能値が，第 1，2 差分 22.5%，第 3 軸は 9%である．本機能は第 1，2 軸差分においても適用可能であり，これらを使った結果を**表 9** に示す．

XFILL を用いた場合，最内のストリーム数に関係なく，L2 キャッシュから L1 キャッシュに swpf が発行される仕様となっているため，各軸とも L1D ミス swpf 率が観察されている．第 1 軸および第 3 軸は推定性能付近まで高速化されたが，第 2 軸のメモリスループットが他と比べてかなり低いことが分かる．

「京」のメモリおよびキャッシュの構成は，メモリから L2 キャッシュへのバスと L2 キャッシュから L1 キャッシュへのバスがハード的に影響を及ぼしあう仕様になっており，ある程度の L2 スループットが要求された場合に，メモリスループットが低下する傾向がある．この現象を L2 エンジンスループットネックと表現し，この上限値は 180 GB/s 程度であることが判明している．XFILL を用いた場合，L2 エンジンスループットは L2 スループット + メモリスループット ×1.5 と表され，第 2 軸では 141.65 GB/s となる．上限値の約 8 割となっているが，第 1，3 軸はそれぞれ 108 GB/s，114 GB/s であることを考慮すると，かなり高い値であり，これがメモリスループットの低下要因であると考える．したがって，第 2 軸の実測値に関しても，ハードウェアの性能によって決定されるという意味では妥当と考えられるが，L2 エンジンスループットネックと性能

表 10　通信部のコスト内訳
Table 10　The detail of cost of communication parts.

	速度袖通信（sec）	応力袖通信（sec）
パック処理	0.623	0.620
Mpi_isend,recv	0.013	0.013
Mpi_waitall	0.660	0.665
アンパック処理	0.578	0.556

推定および評価に関しては，さらなる精査が必要と考える．

微分項部においては，1），2），3）で論じた手法を組み合わせることにより，（a）応力微分項，（e）速度微分項はそれぞれ 10.4%から 17.3%へ，10.1%から 19.2%への性能向上が確認された．

7.　通信部の評価

「京」は前述のとおり，Tofu と呼ばれる 6 次元メッシュ/トーラスネットワークで結合されており，2 次元分割モデルは隣接通信が 1 ホップであることが保障されている．また，4 つの Tofu ネットワークインタフェース（TNI）を搭載しており，4 方向送受信が同時に発行される．その場合，メッセージ長が 1 TNI あたり 10^6 バイト程度以上だとピークとして 13.2 GB/s の通信バンド幅となる [13]．

Seism3D では水平方向に隣接する 4 領域に対し，2 メッシュ分の袖領域の速度成分および応力成分を非同期に送受信する．したがって，先に述べた 13.2 GB/s と比較することにより，「京」の通信ネットワークの性能を十分に活用できているかを評価できる．

16 プロセスにおけるオリジナルコードの通信ルーチン部における測定時間の詳細を**表 10** に示す．なお，本測定では，プロセス間の演算のインバランスの影響を回避するため，通信の前に同期処理を挿入した．

ターゲットモデルの場合，速度，応力ともに 1 回あたり両 X 方向に 80×4000×2×3 素，両 Y 方向に 60×4000×2×3 素の送受信が同時に発生する．よって，MPI_waitall の時間を用いて通信バンド幅を求めると，どちらの通信ルーチンでも 8.1 GB/sec となる．

ピーク性能の約 6 割の性能となっているが，シリアライズされた場合の通信を考えると，本データ量の場合 1.2 秒の通信時間となり，いくつかの通信は並列に実行されている測定結果である．4 TNI が完全に同期するのはタイミング依存であることを考慮すると，妥当な経過時間であるといえる．また，完全に通信が重なった場合でも，アプリケーション全体のうちの 1%未満の改善にすぎないため，本ルーチンの改善は不要と考える．

8.　チューニング後の測定結果

6 章で検証したチューニング手法をオリジナルコードに適用し，16 プロセスで測定した演算部の結果を**表 11** に，

表 11　チューニング前後の性能一覧
Table 11　The performance of before and after tuning.

	Original		Tuning	
	Elapse（sec）	Peak Ratio（%）	Elapse（sec）	Peak Ratio（%）
全体	68.1	11.8	48.8	17.6
(a)	13.0	10.4	7.9	17.3
(b)	3.6	22.3	3.5	22.3
(c)	6.4	15.9	5.6	17.5
(e)	13.5	10.1	7.1	19.2
(f)	15.8	16.2	12.6	21.2
(g)	9.3	8.8	8.1	10.1

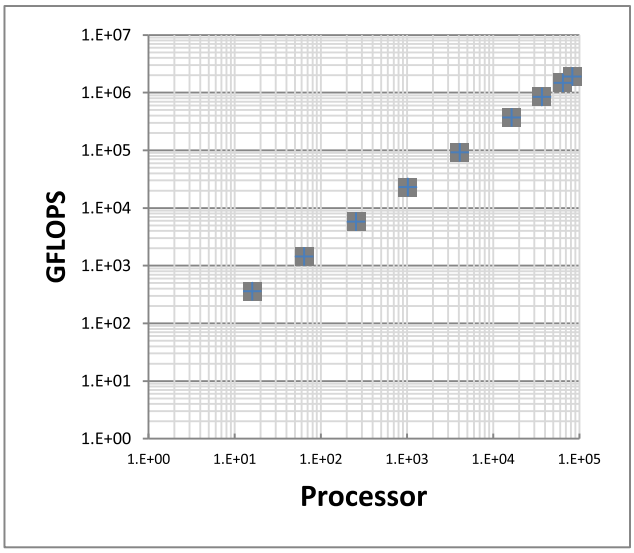

図 6　82,944 並列までの性能
Fig. 6　Measurement result between 16 and 82,944 processes.

チューニングコードを用いて 82,944 並列まで測定した結果を**図 6** に示す．本結果より，効果的な演算部のチューニングと，82,944 並列までの安定したスケーラビリティが確認できる．

9.　おわりに

今回，「京」に備わるハードウェアの基礎性能から推定される性能値を算出し，Seism3D で実装されている主要ルーチンの実測値と比較することにより，チューニング手法を検討，評価するというアプローチを採用した．その際，「京」に提供されている詳細プロファイラ機能を用いて，キャッシュミス率やスループット値を中心とした分析を実施することにより推定性能との差を埋めた．さらに変数に再利用性のあるルーチンではキャッシュを有効活用することによるメモリプレッシャの軽減を狙ったチューニングを施すことにより，さらなる性能向上を実施し，「京」の全ノードを用いて 1.9 PFLOPS を達成した．一部，推定性能と乖離しているカーネルがあるが，演算のバランスなどを含めた検

証は現在実施中である．しかし，本稿で用いた分析手法および高速化手法は，「京」においてアプリケーションを高性能化する際に汎用的なものであり，今回評価したSeism3Dに限らず，差分スキームのようなメモリバンド幅で律速するアプリケーションには有用なアプローチだと考える．

謝辞　本報告に際し，システムソフトウェア開発者の立場でご討論いただいた，富士通株式会社次世代TC開発本部の青木正樹氏，杉山浩一氏，杉崎由典氏，ミドルウェア事業本部アプリケーションマネジメント・ミドルウェア事業部第四開発部の千葉修一氏，庄司智子氏，ならびに理化学研究所計算科学研究機構運用技術部門の皆様に感謝します．本稿の結果は，理化学研究所計算科学研究機構が保有するスーパーコンピュータ「京」の試験利用によるものです．

参考文献

[1] TOP500, available from ⟨http://www.top500.org/⟩.
[2] HPC challenge, available from ⟨http://icl.cs.utk.edu/hpcc/⟩.
[3] 南　一生，井上俊介，堤　重信，前田拓人，長谷川幸弘，黒田明義，寺井優晃，横川三津夫：「京」コンピュータにおける疎行列とベクトル積の性能チューニングと性能評価，ハイパフォーマンスコンピューティングと計算科学シンポジウム論文集，Vol.2012, pp.23–31 (2012).
[4] Furumura, T. and Chen, L.: Parallel simulation of strong ground motions during recent and historical damaging earthquakes in Tokyo, Japan, *Parallel Computing*, Vol.31, pp.149–165 (2005).
[5] Maeda, T. and Furumura, T.: FDM simulation of seismic waves, ocean acoustic waves, and tsunamis based on tsunami-coupled equations of motion, *Pure Appl. Geophys.*, in press, DOI: 10.1007/s00024-011-0430-z (2013).
[6] Maruyama, T.: 2009, SPARC64 VIIIFX: Fujitsu's New Generation Octo-core Processor for Peta Scale Computing, *Hot Chips 21* (2009).
[7] Maruyama, T.: 2010. SPARC64 VIIIFX: A New-SPARC International, The SPARC Architecture Manual Version, Prentice-Hall (1994).
[8] Sparc Joint ProgramminGB/specification (JPS1): Commonality, architecture manual, Sun Microsystems and Fujitsu Ltd. (2002).
[9] SPARC64 VIIIFX Extensions, Fujitsu Ltd., architecture manual (2008).
[10] Ajima, Y., Sumimoto, S. and Shimizu, T.: Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers, *IEEE Computer*, pp.36–40 (2009).
[11] STREAM Benchmark, available from ⟨http://www.streambench.org/⟩.
[12] 村井　均，住元真司，滝康太郎，山中栄次：プログラミング環境—超大規模並列計算機の性能を活かすプログラミング環境，情報処理，Vol.53, No.8, pp.780–786 (2012).
[13] Adachi, T., Shida, N., Miura, K., Sumimoto, S., Uno, A., Kurokawa, M., Shoji, F. and Yokokawa, M.: The design of ultra scalable MPI collective communication on the K computer, *COMPUTER SCIENCE — RESEARCH AND DEVELOPMENT 2012*, DOI: 10.1007/s00450-012-0211-7 (2012).

井上 俊介

1999年横浜国立大学教育学部卒業．同年株式会社富士通長野システムエンジニアリング（現，富士通システムズ・イースト）入社．2010年理化学研究所次世代スーパーコンピュータ開発実施本部に出向．現在，スーパーコンピュータ「京」におけるアプリケーション高度化に従事．

堤 重信

1979年久留米工業高等専門学校電気工学科卒業．1984年現，富士通九州システムズ（株）入社．並列計算機向けプログラムの高速化に従事．2012年より京コンピュータを利用したアプリケーションの高度化に従事．

前田 拓人

2001年東北大学理学部宇宙地球物理学科卒業．2003年同大学院理学研究科博士前期課程，2006年同博士後期課程修了．博士（理学）．2006年防災科学技術研究所契約研究員，2009年東京大学大学院情報学環特任研究員，2011年同特任助教，2012年東京大学地震研究所助教．

南 一生

1981年日本大学理工学部物理学科卒業．同年富士通株式会社入社．主に原子力分野のシミュレーションコードのスパコンへの性能最適化の仕事に従事．2000年財団法人高度情報科学技術研究機構入社．地球シミュレータ用ソフトウェア性能最適化研究に従事．2008年理化学研究所次世代スーパーコンピュータ開発実施本部開発グループアプリケーション開発チームリーダー，2012年理化学研究所計算科学研究機構運用技術部門ソフトウェア技術チームヘッド．2011年ゴードン・ベル賞受賞．