「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

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†

疎行列とベクトルの積は，流体や構造計算等の工学や地球科学の分野で多く使用されている計算カーネルであり，プログラムの要求する B/F 値が高く，スカラマシンでは高い CPU 単体性能を得る事が難しい．本稿では，京速コンピュータ「京」の汎用マシンとしての性能を実証するために準備しているアプリケーションである Seism3D と FrontFlow/Blue の計算カーネルを題材に，性能の予測方法の提示，実測による性能評価，評価結果に基づくチューニング手法の提示，チューニング結果の性能評価について述べる．

Performance Tuning and Evaluation of Sparse matrix-vector multiplication on the K computer

KAZUO MINAMI †, SHUNSUKE INOUE †, SHIGENOBU TSUTSUMI ‡,
TAKUTO MAEDA §, YUKIHIRO HASEGAWA †, AKIYOSHI KURODA †,
MASAAKI TERAI † and MITSUO YOKOKAWA †

In products of sparse matrix and vectors, which are often seen in application programs such as structural analysis code using finite element methods, the higher ratio of memory bandwidth and floating-point rate (B/Flop) is required by the program. A scalar machine is difficult to obtain high performance in a CPU due to its low ratio of B/Flop.
We are developing the K computer as a 10 Peta scale super computer and have to demonstrate its performance by using real applications. Application programs Seism3D and FrontFlow/blue, which have products of sparse matrix and vectors as kernels in the programs, are suitable to show how to obtain higher performance by tuning them to the K computer. In this paper, techniques on how to estimate performance of the codes and to tune them are presented. Results on performance evaluation of tuning versions are also described.

1. はじめに

理化学研究所では，京速コンピュータ「京」(以下「京」と略す)の共用開始に先立ち，システム性能を実証するためのアプリケーション群の整備を進めている．整備の中ではアプリケーション群に対して，超並列性を最大限に引き出すとともに，プロセッサに導入された新機能や強化された機能

† 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 RIKEN, Next-Generation Supercomputer R&D Center
‡ 株式会社 富士通九州システムズ FUJITSU KYUSHU SYSTEMS Ltd.
§ 東京大学大学院情報学環総合防災情報研究センター
The University of Tokyo, Center for Integrated Disaster Information Research (CIDIR)

ⓒ 2012 Information Processing Society of Japan

を十分活用するためのプログラムの書替え作業を進めている．

これらのアプリケーション群は，「京」の汎用性を活かし，様々な応用分野のアプリケーションが幅広く高い性能を発揮できることを実証できるように選択されている．また今後の計算機開発に役立つように，計算機科学的特性の観点を含め選択された．ここで云う計算機科学的特性とは，一つは並列化の面から見て，比較的シンプルな並列化手法で良好な並列性能が得やすいものと，複雑な並列化手法を採用しないと高い並列性能が得られないものの両極端から選択することである．また二つ目は，CPU 単体性能においてアプリケーションが要求するメモリバンド幅と浮動小数点演算数の比（B/F 値）が高く，スカラ型計算機では高い単体性能が得にくい傾向にあるものと，アプリケーションが要求する B/F 値が低く，比較的に高い単体性能が得やすい傾向にあるものから選択する事である．これら二つの観点を基に，地球科学分野のアプリケーションを２本（NICAM［1］, Seism3D［2,3］），ナノ分野のアプリケーションを２本（PHASE［4］，RSDFT［5］），工学分野のアプリケーションを１本（FrontFlow /Blue［6］以下 FFB と略す），物理分野のアプリケーション１本（LatticeQCD［7］），の合計 6 本のアプリケーションを選択した．本稿では，二つ目の観点である CPU 単体性能に焦点を絞り，「京」での高速化手法について述べる．

先に述べたように CPU 単体性能の面から見ると，要求 B/F 値が低いアプリケーションと高いアプリケーションの２つのタイプに分類される．前者は，例えば計算の主要部が行列・行列積の形に書く事ができ，基本的には高い CPU 単体性能を出せるタイプの計算である．先にあげた 6 本のアプリケーションのうち，このタイプに属するのが RSDFT や PHASE といった第一原理計算のためのアプリケーションである．

後者の中では，特に重要な計算カーネルとして，疎行列とベクトルの積が含まれる．先にあげた 6 本のアプリケーションのうち，このタイプに属するのが NICAM や Seism3D といった地球科学分野のアプリケーション，FFB や Lattice QCD がこのタイプのアプリケーションである．疎行列とベクトルの積は，プログラムの要求する B/F 値が高く，スカラ型計算機では高い CPU 単体性能を得る事が難しいが，「京」の汎用性を実証するためには，避けられない計算カーネルである．

2. 目的

CPU 単体性能チューニングにおいては，対象とするコーディングの限界性能値が分からないという問題がある．本稿では，どの段階までチューニング作業を進めるかの判断基準を設ける事を目的に，疎行列とベクトルの積のコーディングから性能を予測する手法を提案する．また，Seism3D 及び FFB を対象に「京」における具体的なチューニング手法を提案する．

本稿では，次の手順で議論を進める．まず理論的に対象のコーディングについて性能の予測値を求める．次に実際の測定結果を予測された性能と比較する．さらにチューニング案を示しチューニング対象のコーディングの性能予測値を示す．その後実際の測定結果と比較しチューニングの効果を評価する．

3. 要求 B/F 値と性能の関係

3.1 疎行列とベクトルの積の特徴

まず，疎行列とベクトルの積について特徴を示す．物理的に３次元の問題を解くのであれば，一般に係数も３次元であり，疎行列も３次元の配列である．しかしコーディング上は１次元や２次元の配列として表現されている場合もある．何れの場合も必要とするメモリ量は大きいのでメモリバンド幅を消費する場合が一般的である．しかし，係数行列を１次元や２次元の配列で表現出来る場合や，係数が定数で表される場合はスカラ量で表現できる場合がある．これらの場合，疎行列の計算部分はメモリバンド幅を消費する事がなく，キャッシュやレジスタにおくことが出来る．ベクトルも物理的に３次元の問題であれば一般的に３次元配列である．しかし行列と同様に，コーディング上は１次元や２次元配列として表現されている場合もある．ベクトル配列の特徴は，行列の各行に含まれる要素数の平均を M 個とすると，ベクトルの次元を L としたとき，M 個程度の再利用性があることである．なぜなら疎行列とベクトルの積の演算数は加算と乗算がそれぞれ M×L 個あり，演算を行なうために使用するベクトルの要素数は L 個であるので，ベクトルの１個の要素は平均 M 回参照されるからである．ベクトルのメモリ量は行列のメモリ量の M 分の１程度の大きさである．ベクトルへのアクセスもメモリバンド幅を消費するが，ここに示した M 回の再利用性を生かしてキャッシュを効率的に利用することが CPU 単体性能を向上させる上で重要なことである．またベクトル量を一次元の量としてリストベクトルで表現されているプログラムもあるが，その場合は，リストベクトルが疎行列と同じ長さだけ必要となりメモリバンド幅を消費することとなる．

Seism3D の疎行列とベクトルの積部分に現れる疎行列はスカラ配列のタイプであり，ベクトルは 3 次元の配列として表現されている．疎行列の各行の平均要素数は数個程度であるため，ベクトル要素の再利用性は，数個程度である．FFB に含まれるカーネルの疎行列は物理的には３次元の量を１次元の配列で表現しているタイプであり，ベクトルも１次元の配列として表現されている．ベクトルはリストアクセスであり，したがって行列と同じ要素数のリスト用の配列が用いられている．リ

ⓒ 2012 Information Processing Society of Japan

ストベクトルの要素の再利用性は，20 から 30 個程度である．

3.2 「京」の CPU 概要

CPU 単体性能について議論するために「京」の CPU の概要について述べる．一つの計算ノードは，一つの CPU（富士通製 SPARC64 TMVIIIfx），16GB のメモリ，計算ノード間のデータ転送を行うインターコネクト用 LSI（ICC: Inter-Connect Controller）で構成されている（図 1）．CPU は，8 つのプロセッサコア，コア共有の 2 次キャッシュメモリ（6MB/12 way/ Write back 方式），メモリ制御ユニットを持っている．CPU の LSI の大きさは縦 22.7mm×横 22.6mm である．各コアは，L1 データキャッシュ(32KB/2way/ Write back 方式)，4 つの積和演算器，256 本の倍精度浮動小数点レジスタを持っており，一つの SIMD 命令（ベクトル処理の一種）により，2 つの積和演算器を同時に動作させることができ，2 つの SIMD 命令を同時に実行することにより一つのコアはクロックサイクル毎に 8 個の浮動小数点演算ができる．従ってコアの理論性能は 16GFLOPS，CPU（8 コア）の理論性能は，単精度/倍精度とも 128GFLOPS となる．また，コア間の並列処理の同期を取るためのハードウェアバリア機構，計算に必要なデータを事前にキャッシュに取り込むプリフェッチ機構，プログラマブルなキャッシュ制御を可能とするセクタキャッシュ機構，など科学技術計算のための様々な機構を備えている．理論メモリバンド幅は 64GB/秒，B/F 値は 0.5 である．L2 キャッシュの理論バンド幅は 256GB/秒，B/F 値は 2.0 である．L1 キャッシュの理論バンド幅はコア毎に 64 GB / 秒，B/F 値は 4.0 である．またメモリ及び L2 キャッシュは 1 ライン(128byte)毎にアクセスされる．

CPU の DGEMM 性能は 123.6GFLOPS（効率 96.6 %），コア間のハードウェアバリア性能は 49nsec であった．また，STREAM ベンチマークコードの triad によるメモリアクセス性能は，46.6GB/秒であった．

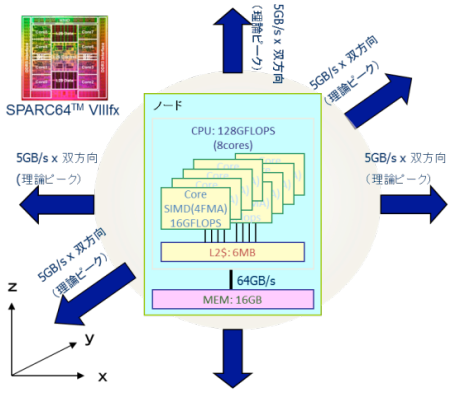

図 1 計算ノード構成

3.3 高い性能を得るための要素

アプリケーションがプロセス間で高い並列化効率を実現できている事とプロセス内でも高いスレッド並列化効率を実現できている事が，アプリケーション全体の性能向上に重要である．この事を前提として，CPU 単体性能向上のために重要と考えられる要素を以下に示す．

(1)プリフェッチの有効利用

3.2 節で示したように「京」の CPU は 64GB/秒という高いメモリバンド幅を実現している．このメモリバンド幅を生かすためには，メモリから L2 キャッシュ，L2 キャッシュから L1 キャッシュへのアクセスは，共にプリフェッチが有効に動作している事が条件である．プリフェッチはロード命令におけるデータアクセスのレイテンシを隠すものであり，プリフェッチが効かずロード命令が発効された時点でメモリ，L2 キャッシュへのアクセスが発生すると，レイテンシ分の大きなペナルティが発生する．L1 キャッシュへのレイテンシを 1 とすると L2 キャッシュのレイテンシは 10 程度，メモリへのレイテンシは 100 程度である．

(2)ラインアクセスの有効利用

3.2 節に示したようにメモリと L2 キャッシュについては，1 ライン(128 バイト)毎にアクセスされる．高い性能を得るためには，ロードしてきた 1 ラインのデータをなるべく多く使用した演算を行なう事が重要である．1 ラインのデータのうち例えば 8 バイトのデータ 1 個しか使用できない場合は，大きなペナルティとなり見かけ上のメモリバンド幅は 1/16 となる．

(3)キャッシュの有効利用

要求 B/F 値が低いアプリケーションの場合は，レジスタやキャッシュブロックのテクニックにより高い性能を得ることが出来る．要求 B/F 値が高いアプリケーションの場合でも，キャッシュの利用率を高める事が性能向上に大きな効果がある．またキャッシュの利用においては，キャッシュのスラッシングを無くす事が重要である．

(4)効率の良い命令スケジューリング

「京」は 256 本という多数の浮動小数点レジスタが装備されている．この浮動小数点レジスタをコンパイラが有効利用し，ソフトウェアパイプラインニングやループアンローリング等でループ内の演算のスケジューリングができている事が性能向上のために重要である．コンパイラがうまくスケジューリングできない場合，手動でループ分割やアンロールをする事で性能が向上する場合がある．スケジューリングがうまくできていない場合，浮動小数点演算待ち時間や 1 命令の実行時間の割合が増大し著しく性能が劣化する．

ⓒ 2012 Information Processing Society of Japan

(5)SIMD 演算器の有効利用

演算が SIMD 化され SIMD 命令が有効に利用される事が性能向上に必要である．SIMD 命令は積和演算を実行する場合に高い実行効率を発揮するため，演算全体における積和演算の割合を高める事も性能向上にとって重要である．

3.4 高い性能を得るための要素と要求 B/F 値の関係

Seism3D と FronrFlow/Blue の疎行列とベクトルの積のカーネルは，要求する B/F 値が大きい計算カーネルである．要求する B/F 値が大きいアプリケーションについては，CPU 演算性能を最大限使用することよりも，使用しするメモリバンド幅が，実効メモリバンド幅に出来るだけ近いこと，つまりメモリバンド幅を使い切る事が大事であり，上記の５つの項目のうち最も重要なのは(1)(2)である．さらに本稿で扱う疎行列とベクトルの積の場合は，ベクトルの再利用性を生かすために(3)が重要であり，ベクトルをなるべくオンキャッシュにする事が必要である．これら(1)～(3)が満たされ計算に必要なデータが演算器に供給された状態で，それらのデータを十分使える程度に(4)のスケジューリングができて，さらに(5)の SIMD 演算器が有効に活用できる状態である事が必要である．

4. 性能予測手法

性能予測の例としてまず，図 2 の seism3D の Z 方向の速度の差分項のコーディングを用いて説明する．このコーディングは疎行列とベクトルの積であり，疎行列が定数のタイプである．ここで NZ=4000, NX=60,NY=80 である．またこのプログラムは単精度でコーディングされている．したがって V 及び DZV の配列全体の大きさは，76.8MB，K 軸×I 軸の大きさは 960KB，K 軸の大きさは 16KB となる．したがって，配列全体はキャッシュに入る大きさではなく，K 軸×I 軸の一つ分は L2 キャッシュに入る大きさであり，K 軸は L1 キャッシュに入る大きさである．

この例では，Flop 値は 5 であり，コーディングが要求する B/F 値は以下のように求める．

まず要求 Byte の値を計算する．V(K-2,I,J)はメモリからロードされるが，この時 V(K-1,I,J), V(K,I,J), V(K+1,I,J)が同一キャッシュラインにあると仮定すれば，一緒に L1 キャッシュにオンキャッシュとなっているため，メモリからロードする必要はなく，L1 キャッシュからロードされる．DZV(K,I,J)は一度メモリからロードされ，またメモリにストアされる．したがってメモリからのロード/ストア数は３となる．またキャッシュからのロード/ストア数は 6 である．データをレジスタに持ってくるまでにどこにボトルネックがあるかは，以下のように考えた．メモリから L2 キャッシュ，L2 キャッシュから L1 キャッシュ，L1 キャッシュからレジスタへのデータ転送のバンド幅は，CPU とメモリ間の実効メモリバンド幅 46GB/sec を基準に考えると，1：5.6(=256/46)：11.1(=512/46)である．したがって，それぞれの間のデータ移動時間比は，メモリと L2 キャッシュ間は単精度データ 3 個の移動，L2 キャッシュと L1 キャッシュ間，L1 キャッシュとレジスタ間は単精度データ 6 個なので，3：6/5.6：6/11.1 = 3:1.1:0.5 となり，メモリから L2 キャッシュへの移動が支配的になる．この部分が要求する Byte 値はメモリへのアクセスのみ考慮すれば良い．以上より，図 2 のコーディングが要求する Byte 値は 3×4 バイトで 12 バイトとなり．要求 B/F 値は 12/5=2.4 となる．

一方，実効的な B/F 値は，理論 B/F 値（理論メモリバンド幅とピーク性能の比）0.5 に，理論的なメモリバンド幅と実効メモリバンド幅の比 0.72（=46/64）を乗じて 0.36 と考える．したがって，コーディングの要求 B/F 値が 2.4 に対してハードウェの実効 B/F 値が 0.36 であるので，予測性能値として 0.36/2.4=0.15 が求まり，ピーク性能比の 15%の性能がこのコーディングの最大性能であると予測できる．

5. Seism3D のチューニング

本節以後の実行結果は，表 1 の計算環境による評価結果である．また，これ以降にあらわれる表中の性能予測値，実測値における%付きの数値は対ピーク性能比の値である．

表 1「京」を用いた評価環境

ハードウェア	京速コンピュータ「京」 SPARC64™ VIIIfx, 2GHz, 8core/CPU, 1CPU(8core)/node
ソフトウェア	・Linux ・「京」向け言語開発環境（Fortran, MPI)
使用コンパイルオプション	-Kvisimpact,ocl,ilfunc,preex,array_private

5.1 Seism3D コードの概要

Seism3D は，有限差分法により数値的に粘弾性方程式を時間発展させることにより，地震伝播と津波を連動して解く，大規模な並列化に対応しているアプリケーションである．Seism3D コードは以下の６つの計算部分から構成される．

(a)応力空間微分計算

(b)速度空間微分計算

(c)応力時間積分計算

(d)応力時間積分吸収計算

(e)速度時間積分計算

(f)速度時間積分吸収境計算

ⓒ 2012 Information Processing Society of Japan

5.2 応力及び速度の空間微分計算の性能予測

5.1 節に示した 6 つの計算ブロックのうち(a)と(b)の空間微分計算は同じ計算を実施している．それぞれ Z 方向，X 方向，Y 方向の 3 つの差分計算のループを含んでいる．それぞれのコーディングを図 2～図 4 に示した．

```
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))*R40 &
                  - (V(k+1,I,J)-V(k-2,I,J))*R41
    end do
  end do
end do
```

図 2 Seism3D Z 方向速度差分計算

```
do J = 1, NY
  do I = 1, NX
    do K = 1, NZ
      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 3 Seism3D X 方向速度差分計算

```
do J = 1, NY
  do I = 1, NX
    do K = 1, NZ
      DYV (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 4 Seism3D y 方向速度差分項

それぞれのループにおいて最外ループの J ループでブロック分割によるスレッド並列化が行なわれている．Z 方向のループの性能予測値は，4 章に示した通りである．X 方向の性能予測は，ほぼ Z 方向の性能予測と同じである．異なるのは 4 つの V のロードのうち 3 つの V の要素が，Z 方向の差分では L1 オンキャッシュと予測されるのに対し，X 方向差分では，16KB×4=64KB の範囲にあるため L1 か少なくとも L2 キャッシュにオンキャッシュと予測されることである．いずれにしてもメモリのアクセスは発生しないため，要求 byte 値は Z 方向差分と同様に 2.4 となり予測性能は 15%となる．Y 方向の性能予測は，Z/X 方向の予測とは大分異なる．K 軸×I 軸の大きさは一つ 960KB もあり，J 軸でスレッド並列されている事を考えると，V の 4 つの要素は，何れもキャッシュに残っていないと考えられる．従ってメモリからのロード/ストアは 6，要求 byte は 24 となり，要求 B/F 値は 24/5=4.8 となり予測性能は 7.5%となる．

5.3 空間微分計算の実測結果と評価

5.2 節に示した要求 B/F 値と性能予測値を実測値と合わせて表 2 にまとめる．

表 2 を見ると予測値と実測値が良く一致している事が分かる．

図 2～図 4 のコーディングを見ると連続アクセスのデータであり，基本的にプリフェッチが効くパターンであり 3.3 節の(1)の条件:プリフェッチの有効利用を満たしていると考える．また連続アクセスである事から(2)の条件：ラインアクセスの有効利用も満たしている．これらの条件を満たしているため，予測値と実測値が良く一致しているものと考える．この評価結果により，このコーディング自体の性能改善の余地はないものと考える．

表 2 Seism3D ZXY 方向速度差分項の結果

	Z 方向差分	X 方向差分	Y 方向差分
要求 B/F 値	2.4	2.4	4.8
性能予測値	15.0%	15.0%	7.5%
実測値	15.3%	15.1%	7.6%

5.4 サイクリック分割スレッド並列

更なるチューニングの方法として，まず ZXY の各ループのループ融合を考えた．3 つのループの中にそれぞれ V(K,I,J)の要素の参照があるため，ループを融合することにより全体では V(K,I,J)のロードを 2 個分減らす効果があると考えられる．つぎに最外の J ループ のスレッド並列をブロック分割によるスレッド並列から，サイクリック分割によるスレッド並列に変更する事が考えられる．J 軸のループにおいて V(K,I,J)はメモリからロードされる．しかし V(K,I,J-1)，V(K,I,J-2)，V(K,I,J+1)の 3 つの配列については，J 軸のサイクリック分割によるスレッド並列化により，両隣のスレッドが L2 キャッシュにロードしている V の要素を利用する事を期待できる．これは，L2 キャッシュが 8 つのコアで共有されている「京」の特性を生かすものである．図 5 にチューニング後のコーディングを示す．

5.5 チューニングコードの性能予測

5.4 節で述べたチューニング後のコーディングの予測性能値を考える．まず Flop 値は 15 である．今までの議論のように V のうち一つの要素はメモリからロードされる．その際，他の 11 個の V の要素は L1 か L2 キャッシュに乗っているものと考えら

ⓒ 2012 Information Processing Society of Japan

れる．

DZV(K,I,J)，DXV(K,I,J)，DYV(K,I,J)は一度メモリからロードされ，その後メモリにストアされる．従ってメモリからのロード/ストアは 7 となる．図 5 のコーディングが要求する byte 値は 7×4 バイトで 28 バイトとなり，要求する B/F 値は 28/15=1.86 となる．したがって性能予測値は，0.36/1.86=0.19 より 19%と予測される．

```
!$OMP DO SCHEDULE(static,1),PRIVATE(I,J,K)
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))*R40 &
                  - (V(k+1,I,J)-V(k-2,I,J))*R41
      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
      DYV (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 Seism3D y 方向速度差分項(cyclic 分割スレッド並列化)

このコーディングの性能予測値と実測結果を表 3 にまとめた．ZXY 全体で 18%程度の性能が得られており，予測と実測値もよく一致している．オリジナルの性能と比較してチューニングの効果が得られていることも明らかとなった．

表 3 Seism3D ZXY 方向速度差分項融合(サイクリック分割スレッド並列化)の結果

要求 B/F 値	28/15 = 1.86
性能予測値	0.36/1.86 = 0.19
実測値	17.7%

5.6 応力時間積分計算の性能予測と実測評価

4.1 節に示した 6 つの計算ブロックのうち(c)に示した応力時間積分計算の性能予測と実測結果を表 4 に示す．応力計算部はここまで評価してきたような-1，+1 のような差分計算はない．従って疎行列とベクトルの積でもベクトルの再利用性がないパターンと云える．全ての配列が，ZXY の差分計算に出現した DXV，DYV，DZY のアクセスのように(K,I,J)のインデックスで配列をアクセスする．従って全ての配列は，メモリをアクセスする配列となっている．コーディングが約 100 行と長いためここには記載しないが，今までの議論と同様の評価を実施し要求 B/F 値と性能予測値を求めた．

表 4 応力時間積分計算の結果

要求 B/F 値	252/175 = 1.44
性能予測値	0.36/1.44 = 0.25
実測値	17.4%

表 4 を見ると予測値よりも実測値が大分低いことが分かる．

5.7 応力時間積分計算のチューニング

5.7 節の結果を受け，もう少し性能向上の見込みがないかチューニング方法を検討した．詳細に計測データを検討すると，使用したメモリバンド幅の測定結果は 35GB/sec 程度であり，3.3 節(1)の条件を満たしていないことが分かった．またプリフェッチの状況を見るとハードウェアプリフェッチが効率的に出ていない事が分かった．そこでいくつかの配列を融合し配列の数を減らす事でストリームの数を減らし，ハードウェアプリフェッチの効率を高めた．このチューニングにより実行性能が 17.4%から 21.8%まで向上し性能予測値に近づくとともに．メモリバンド幅値も実効値 : 46GB/秒に近い 42.4GB/値となった．

5.8 Seism3D の全体のチューニング結果

ここまで 5.1 節に示した計算部分(a)～(c)について議論してきた．これ以外の計算部分(d)～(f)についても同様の評価とチューニングを実施した．結果を表 5 にまとめる．全体は通信を含めた対ピーク性能を示しており 10.3%から 15.3%に性能向上している．微分・積分計算については通信を含めない CPU 単体性能を示しており，何れも性能向上が見られる．

表 5 Seism3D 全体のチューニング結果

	オリジナル 時間(秒)	peak 比	チューニング 時間(秒)	peak 比
全体	75.5	10.3%	52.9	15.3%
応力空間微分	13.0	10.4%	9.2	14.7%
速度空間微分	13.4	10.1%	7.7	17.7%
応力時間積分	12.8	17.0%	11.5	21.8%
応力時間積分境界	12.5	7.6%	10.3	10.2%
速度時間積分	9.7	7.5%	3.0	23.0%
速度時間積分境界	7.9	15.3%	6.9	17.5%

6. FFB のカーネルチューニング

FFB コードは，有限要素法を用いた流体計算のプログラムである．有限要素法には全体剛性マトリクスを構築するタイプの計算と全体構成マトリクス

ⓒ 2012 Information Processing Society of Japan

を構成せずに要素剛性マトリクスのみで計算を進めるエレメント・バイ・エレメント法がある．本コードは，新バージョンにおいて両方のソルバに対応しているが，本稿で議論する疎行列とベクトルの積は，前者のソルバで使用される計算カーネルである．以下では，このカーネルのチューニングについて述べる．

6.1 計算カーネルの性能予測

カーネルのコーディングを図 6 に示す．

```
      ICRS=0
      DO 110 IP=1,NP
         BUF=0.0E0
         DO 100 K=1,NPP(IP)
              ICRS=ICRS+1
              IP2=IPCRS(ICRS)
              BUF=BUF+A(ICRS)*S(IP2)
100      CONTINUE
         AS(IP)=AS(IP)+BUF
110   CONTUINE
```

図 6 FrontFlow/Blue コードのカーネル

本評価で用いたモデルは，隣接節点数が高々27 の 6 面体要素と，同 24 の 4 面体要素である．カーネルは疎行列とベクトルの積であり 2.2 節の最後で述べたような特徴を持っている．行列データの格納方式には一般的な CSR（Compressed Sparse Row）形式を用いている．このコーディングはベクトルがリストアクセスとなっているため，ベクトル要素の参照でメモリにアクセスする場合は，メモリアクセスのレイテンシが発生することと，1 ライン(128 バイト)のうち 1 要素しか使用しない事による大きなペナルティが発生することになり，著しい性能低下が予測される．ベクトルの要素が L2 キャッシュにオンキャッシュになった場合についても，メモリアクセス程には低下しないが，L2 キャッシュに対して同様のペナルティが発生し，かなりの性能低下が予測される．もし計算に用いるベクトルの部分が L1 キャッシュに乗っていれば，レイテンシおよびラインアクセスに対するペナルティはなくなるため，ベクトルのメモリへのアクセスペナルティを全く無視してよい．その場合の性能予測値は，要求 byte は 2 要素×4byte で 8 となり，flop の値は 2 であるため要求 B/F 値は 4 となる．実効 B/F 値 0.36 を使うと予測性能は，0.36/4=0.09 となり 9%となる．

6.2 計算カーネルの実測結果と評価

オリジナルコードのカーネルはスレッド並列されていないため 1 コアで測定した．測定結果を表 6 に示す．数値は 1 コアのピーク性能 16GFLOPS に対するピーク性能比である．メモリバンド幅を 1 コアで占有する場合の STREAM ベンチマークの結果は 20GB/秒である．従って理論的な B/F 値は 20GB/16GFLOP で 1.25 となる．要求 B/F 値は 4 であるので，予測性能値は 1.25/4=0.31 で 31%となる．この予測値に対して表 6 の値は著しく小さいと考えられる．原因は，6.1 節に述べたベクトルアクセスの大きなペナルティに加えて，最内ループの回転数が高々27 であることに起因する非効率な命令スケジューリングであると推測される．

表 6 計算カーネルの測定結果(1 コア)

	6 面体(peak 比)	4 面体(peak 比)
オリジナル	5.9%	2.4%

6.3 フルアンロールによるデータ格納形式

D. Guo らによれば CSR 形式を拡張した S-CSR (Streamed Compressed Sparse Row) 方式に行列の格納方法を変更する事で性能向上する事が報告されている[8]．S-CSR-2 方式を採用した場合のコーディング例を図 7 に示す．

```
      ICRS=0
      DO 110 IP=1,NP,2
         TBUF1=0.0E0
         TBUF2=0.0E0
         L=NPP(IP)
         DO 100 K=1,L
            ICRS=ICRS+1
            TBUF1=TBUF1+A(ICRS  )*S(IPCRS(ICRS)  )
            TBUF2=TBUF2+A(ICRS+L)*S(IPCRS(ICRS)+L)
100      CONTINUE
         ICRS=ICRS+(1+L)
         AS(IP  )=AS(IP  )+TBUF1
         AS(IP+1)=AS(IP+1)+TBUF2
110   CONTUINE
```

図 7 S-CRS-2 の格納方式の場合のコーディング

実測の結果は，報告の通り性能向上はするものの 6 面体で 7%程度，4 面体で 10%程度の性能向上であった．そこで図 8 のような内側のループをフルアンロールするデータの格納形式とコーディングを

```
      ICRS=0
      DO 110 IP=1,NP
        BUF=0.0E0
          BUF=BUF+A(ICRS+ 1)*S(IPCRS(ICRS+ 1))
     &           +A(ICRS+ 2)*S(IPCRS(ICRS+ 2))
        ・・・・・省略・・・・・
     &           +A(ICRS+26)*S(IPCRS(ICRS+26))
     &           +A(ICRS+27)*S(IPCRS(ICRS+27))
          ICRS=ICRS+27
          AS(IP)=AS(IP)+BUF
  110 CONTINUE
```

図 8 提案した格納形式の場合のコーディング

採用した．図 8 の例は 6 面体の例であり，4 面体の場合は展開の数を 24 とする．このデータ格納形式では 6 面体の場合，IP の各行に対し行列要素が 27 個に満たない場合は，展開数を固定するために 27 個になるまで 0 を詰めることになる．0 を詰める要素数は 6 面体の場合，全体の 2%程度，4 面体の場

ⓒ 2012 Information Processing Society of Japan

合は 35%程度となり，メモリ量が増加するが，計算性能を上げるためには必要なものと考える．

6.4 ベクトルデータのオーダリング手法

6.1 節に示したベクトルへのアクセスペナルティを取り除くために節点の順序付けを変更した．
図 9 の左のように節点の XYZ の 3 次元座標を使って 3 次元の空間に節点番号をマッピングする．つぎに各軸方向に 3 次元空間を分割する．図 9 では 3 分割となっているが，今回は 10 分割とした．基本的には分割された一箱の中に含まれる節点が連続に並ぶように順序付けを変更し，箱の番号も図 9 の中央の図のように XYZ の順番に順序付けする．また一箱の中の節点も，図 9 の右側の図のように内部と外周に分割し，内部に含まれる節点を先に順序付けし，その後外周部に含まれる節点を順序付けする．

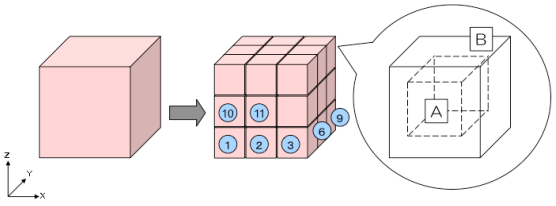

図 9 節点の順序付けの変更

このように順序付けを変更する事で，物理的に近い節点が配列の並びとしても近い位置に配置される事となり，ひいては一要素を構成する節点の番号も近くなる．このチューニングで一箱の大きさを調整することにより，ベクトルのリストアクセスの多くに対し，同一キャッシュラインにのるデータを多くすることにより，ペナルティが発生しない L1 オンキャッシュの状態になる事が期待できる．

チューニング後の実測の結果を表 7 に示す．表中の数値は 1 コアの場合は，1 コアのピーク性能 16GFLOPS に対するピーク性能比，8 コアの場合は，8 コアのピーク性能 128GFLOPS に対するピーク性能比である．フルアンロールにより 1 コア測定において 4 面体も 6 面体も 2 倍程度の性能向上を得られている．フルアンロールと節点の順序付けの変更により 8 コアの測定において，6.1 節に示したベクトルが理想的に L1 オンキャッシュである時の理論性能値である 9%に近い性能値が得られている．この事より，6.4 節に示したチューニング手法が有効であったことが分かる．

表 7 計算カーネルのチューニング結果

	６面体	４面体
オリジナル(1core)	5.9%	2.4%
フルアンロール(1core)	10.8%	4.2%
フルアンロール(8core)	5.4%	3.0%
フルアンロール ＋ リオーダリング(1core)	10.2%	10.2%
フルアンロール ＋ リオーダリング(8core)	8.1%	7.7%

7. まとめ

本稿では，重要な計算カーネルである疎行列とベクトルの積の「京」における CPU 単体性能のチューニングについて述べた．具体的には，まず対象とするコーディングの性能予測手法を提案した．次に，Seism3D を例として具体的チューニング手法であるサイクリックスレッド並列の手法を提案した．また FFB を例として具体的チューニング手法である，フルアンロールによるデータ格納手法とベクトルデータのリオーダリング手法を提案した．「京」を使用した実測の結果，提案した性能予測手法で対象コーディングの性能が良く予測できる事が確認された．Seism3D では，提案したチューニング手法により，コード全体で 1.5 倍程度の性能向上が得られ，チューニングの有効性が確認された．計算カーネルについても、ほぼ理論上の性能予測値まで性能が高められた．FFB の計算カーネルにおいても，提案したチューニング手法の有効性が確認され，ほぼ理論上の性能予測値まで性能が高められた事が分かった．本稿に示した，性能予測手法，チューニング手法が，「京」の上で色々なタイプの疎行列とベクトルの積のプログラムに応用することが期待できる．

8. 謝辞

本性能最適化に際し御討論頂き貴重な助言を頂いた，東京大学地震研究所の古村孝志教授，東京大学生産研究所の加藤千幸教授，富士通株式会社の井上晃氏，並びに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の諸氏に感謝します．また，次世代スーパーコンピュータ開発実施本部プロジェクトリーダー渡辺貞氏の暖かい励ましに感謝いたします．本論文の結果は，理化学研究所計算科学研究機構が保有する京速コンピュータ「京」の試験利用によるものです．

ⓒ 2012 Information Processing Society of Japan

参 考 文 献

[1] M. Satoh, T. Matsuno, H. Tomita, H. Miura, T. Nasuno and S. Iga, “Nonhydrostatic Icosahedral Atmospheric Model (NICAM) for global cloudresolving simulations.”, Journal of Computational Physics, the special issue on Predicting Weather, Climate and Extreme events, 227, pp3486-3514, 2008.

[2] T. Furumura and L. Chen, “Parallel simulation of strong ground motions during recent and historical damaginge earthquakes in Tokyo, Japan”, Parallel Computing, 31, pp149-165, 2005.

[3] 古村孝志, “差分法による３次元不均質場での地震波伝播の大規模計算”, 地震 2, 61 巻, S83-S92, 2009.

[4] 「乱流音場解析ソフトウェア FrontFlow/Blue: 」 http://www.ciss.iis.u-tokyo.ac.jp/riss/project/device/

[5] J. Iwata, D. Takahashi, A. Oshiyama, T. Boku, K. Shiraishi and S. Okada, "A massively-parallel electronic-structure calculations based on real-space density functional theory", Journal of Computational Physics 229, pp2339-2363, 2010.

[6] http://www.ciss.iis.u-tokyo.ac.jp/rss21/theme/multi/fluid/fluid_softwareinfo.html

[7] S.Aoki, K.-I.Ishikawa, N.Ishizuka, T.Izubuchi, D.Kadoh, K.Kanaya, Y.Kuramashi, Y.Namekawa, M.Okawa, Y.Taniguchi, A.Ukawa, N.Ukita and T.Yoshie, “2+1 Flavor Lattice QCD toward the Physical Point”, Physical Review D79, 034503, 2009.

[8] D. Guo and William Gropp, “Optimizing Sparse Data Structures for Matrix-vector Multiply”, IJHPCA, vol. 25, no. 1, pp115-131, 2011.

ⓒ 2012 Information Processing Society of Japan