

# キャッシュの効果을考慮した ルーフラインモデルの拡張によるプログラムの性能予測

南 一生<sup>1,a)</sup> 井上 俊介<sup>3,b)</sup> 千葉 修一<sup>2,c)</sup> 熊畑 清<sup>1,d)</sup> 横川 三津夫<sup>4,e)</sup>

受付日 2015年11月9日, 採録日 2016年2月20日

**概要:** 現代のスーパーコンピュータにおける高性能計算において, プロセッサ単体の実効性能を高くするチューニングは非常に重要である. このため, 対象とするプログラムの期待する実効性能 (限界性能) の予測手法が望まれている. ルーフラインモデルは, メモリバンド幅が性能律速要因となっているプログラムの実効性能を予測することができ, 広く用いられている. しかし, キャッシュアクセスが増える場合に, ルーフラインモデルによる予測性能と実測性能が乖離することが分かった. 本論文では, キャッシュアクセスが増大した場合にも適用可能なプログラムの性能予測モデルを提案する. このモデルをスーパーコンピュータ「京」上の L2 キャッシュアクセスのある実際のアプリケーションの実効性能予測に適用し, 提案モデルによる予測が可能であることを明らかにした.

**キーワード:** HPC, 性能チューニング, 性能評価, ルーフラインモデル, キャッシュ

## Performance Estimation of Programs by an Extension of the RoofLine Model Considering Cache Effects

KAZUO MINAMI<sup>1,a)</sup> SHUNSUKE INOUE<sup>3,b)</sup> SYUICHI CHIBA<sup>2,c)</sup> KIYOSHI KUMAHATA<sup>1,d)</sup>  
MITSUO YOKOKAWA<sup>4,e)</sup>

Received: November 9, 2015, Accepted: February 20, 2016

**Abstract:** A tuning technique which makes the sustained performance of programs on a single processor higher is very important in high performance computing on modern supercomputers. Therefore, prediction of the marginal performance of programs is a big concern to know how extent we can tune programs. The roofline model can estimate the marginal performance of programs well, if the performance is limited by effective memory bandwidth. The model, however, could not be applied to the performance prediction in the case of increasing of L2 cache accesses. In this paper, we proposed a new prediction model of the marginal performance which can be applied in the case of increasing of accesses to L2 cache. It is found that the new model works well for the marginal performance prediction by applying it to actual programs on the K computer and other systems.

**Keywords:** HPC, performance tuning, performance evaluation, roofline model, cache memory

<sup>1</sup> 理化学研究所計算科学研究機構運用技術部門  
RIKEN Advanced Institute for Computational Science Operations and Computer Technologies Div., Kobe, Hyogo 650-0047, Japan  
<sup>2</sup> 富士通株式会社次世代テクニカルコンピューティング開発本部  
NEXT GENERATION TECHNICAL COMPUTING UNIT, FUJITSU LIMITED, Numazu, Shizuoka 410-0396, Japan  
<sup>3</sup> 株式会社富士通システムズ・イースト解析シミュレーション部  
Analysis Solutions Department, Fujitsu Systems East Limited, Nagano 380-0813, Japan  
<sup>4</sup> 神戸大学大学院システム情報学研究科  
Graduate School of System Informatics, Kobe University, Kobe, Hyogo 657-8501, Japan

### 1. はじめに

科学技術計算におけるアプリケーションプログラム開発では, 過去, 数理モデルの定式化や離散化に基づく式に忠実に, かつ物理現象に沿った素直なプログラミングを行う

a) minami.kaz@riken.jp  
b) inoue.shunsuke@jp.fujitsu.com  
c) shuc@jp.fujitsu.com  
d) kuma@riken.jp  
e) yokokawa@port.kobe-u.ac.jp

ことが一般的であり、その実行性能はコンパイラのコード生成能力と計算機アーキテクチャの計算性能に任せていた。しかし現在は、計算機アーキテクチャの変化によりプログラミング方法に変化が生じている。計算機アーキテクチャの変化とは、1つは、単体 CPU の性能向上の限界を突破しシステム全体の性能を向上させるための超並列アーキテクチャの採用である。もう1つは、メモリウォール問題に対処するためにキャッシュ等を用いた多階層からなるメモリ構造の採用である。これらのアーキテクチャの変化に対応するための「超並列性を引き出すプログラミング」と「プロセッサ単体の実行性能を引き出すプログラミング」は、現代のスーパーコンピュータ、特に8万個あまりにおよぶプロセッサを備え、数々の数値計算のための新機能が導入されている「京」のようなスーパーコンピュータにおいては、研究者やプログラマが意識すべきプログラミング上の重要な点となってきた。

プロセッサ単体の実行性能を引き出すチューニングにおいては、キャッシュを有効利用するためのプログラミング上の様々なテクニックが提示されている [1], [2]。このようなテクニックを駆使し、CPU 本来の性能を引き出し、シミュレーション時間を最小化することは、限られた計算資源の有効活用、また研究期間の短縮においては重要である。しかし、プロセッサ単体性能のチューニングを進める場合、対象とするプログラムの期待する実効性能、つまり限界性能が容易に予測できないことが問題となっている。

Williams らは、理論メモリバンド幅、CPU の理論ピーク性能、Operational Intensity (Flop/Byte) を用いた性能予測手法である Roofline Model (ルーフラインモデル) を提案した [3]。我々も、「京」においてルーフラインモデルを用いて、プログラムのコーディングを精査することにより限界性能を予測し、その予測値を目標に CPU 単体性能チューニングを実施している [4]。しかし、これらの性能チューニングの中でルーフラインモデルは、メモリバンド幅が性能律速要因となっている場合には予測性能と実測性能がよく一致するが、キャッシュアクセスが増える場合には、予測性能と実測性能が乖離してくることが分かった。

本論文では、「京」上のプロセッサ単体性能の評価、およびチューニングを通して明らかになった事例を基に、キャッシュアクセス頻度が高いプログラムの限界性能値を予測するモデルを提案し、メモリバンド幅が性能律速要因となっている状態から、キャッシュアクセスが増大しキャッシュメモリバンド幅が性能律速要因となるプログラムの性能予測に適用可能であることを示す。

まず、2章において関連研究について述べる。4章においてルーフラインモデルの検証結果を述べ、ルーフラインモデルの適用限界例について示す。5章において、メモリとキャッシュが混合して使用される状態での性能限界値予測モデル、その予測モデルの検証結果を示す。さらに6章

において、「京」における具体的な性能予測手法を示す。最後に7章で、「京」における実アプリケーションでの性能予測と評価結果について述べる。

## 2. 関連研究

プロセッサ単体の実行性能を引き出すチューニングにおいては、キャッシュの有効利用が必須である。そのために、キャッシュラインを考慮した最適化手法 [2] やステンシル計算の性能最適化の研究 [5]、疎行列ベクトル積の性能最適化の研究 [6] 等多くの研究がなされている。特に空間方向と時間方向の両方を考慮してキャッシュの有効利用を図る、テンポラルブロッキングの手法に関する研究も多くなされている [1], [7], [8]。

プロセッサ単体の実行性能を引き出すチューニングの研究に関連し、プログラムの限界性能を予測する手法として、Williams らは Roofline Model (ルーフラインモデル) を提案した [3]。この研究では、ソースプログラムの情報から得られるアプリケーションの要求する Operational Intensity (Flop/Byte) を用い、ハードウェアの理論メモリバンド幅および CPU 理論ピーク性能から、アプリケーションの限界性能が予測可能であることを示した。このルーフラインモデルが、メモリバンド幅に性能が支配されるプログラムの性能限界値の予測に対し有効な手法であることが、Rossinelli らの研究 [9]、佐藤らの研究 [10] で示されている。我々も、STREAM ベンチマーク [11] で測定された実効メモリバンド幅を理論メモリバンド幅の代わりに用いることで、地震波伝播アプリケーション Seism3D [12] と汎用流体解析コード FrontFlow/Blue [13] のチューニングの際に、ルーフラインモデルによる性能予測が、実アプリケーションの性能をよく予測できることを確認した [4]。

しかし Williams らのルーフラインモデルは、ステンシル計算の例 [5] や疎行列ベクトル積の例 [6] について、プログラムの要求 Operational Intensity (Flop/Byte) の計算において、キャッシュに載っているデータについては考慮しておらず、キャッシュの効果を考慮しないモデルとなっている。

Ilic らは、L1 キャッシュ、L2 キャッシュ、ラストレベルキャッシュ (LCC)、およびメモリからコアへの実測に基づく実効バンド幅を用いたキャッシュの効果を取り入れたアプリケーション性能予測モデル (Cache-aware Roofline model) を提案している [14]。このモデルは、メモリおよび複数のレベルのキャッシュのいずれか1つに、すべてのデータが載っていると仮定しているほか、アセンブリ言語によるプログラムを用いたキャッシュの実効バンド幅は、ほぼ理論バンド幅と同じとしている。一般のプログラムでは、データはメモリおよび複数レベルのキャッシュに分散して載っている状態が一般的であり、また4.2節で後述するようにキャッシュのデータ転送バンド幅が理論バンド幅

に達することはほとんどない。

本論文では、キャッシュとメモリの両方にデータが載った状態での性能予測モデルに拡張するとともに、Fortran等の高級言語を使用したステンシル計算や疎行列ベクトル積といったアプリケーションに近いプログラムを対象に拡張した性能予測手法を提案する。

### 3. 「京」のCPU概要

プログラムの「京」での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 [15]。1つの計算ノードは、1つのCPU (富士通製 SPARC64<sup>TM</sup>VIIIfx), 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。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/12way/Write back方式)、メモリ制御ユニットを持っている。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、1つのSIMD命令により、2つの積和演算器を同時に動作させることができる。したがって、クロックサイクルごとに2つのSIMD命令を同時に実行することにより1つのコアは8個の浮動小数点演算ができ、したがってコアの理論ピーク性能は16GFLOPS、CPU (8コア)の理論ピーク性能は、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と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ブリフェッチ機構、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等科学技術計算のための様々な機構を備えている。メモリの理論バンド幅は64GB/s、B/F値は0.5、L2キャッシュの理論バンド幅は256GB/s、B/F値は2.0、L1キャッシュの理論バンド幅はコアごとに512GB/s、B/F値は4.0である。メモリおよびL2キャッシュは1ライン (128byte) ごとにアクセスされる。CPUのDGEMM性能は123.6GFLOPS (実行効率96.6%)、コア間のハードウェアバリア性能は49ナノ秒であった。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/sであった。

### 4. 「京」でのルーフラインモデルの検証

#### 4.1 ルーフラインモデル

ルーフラインモデルは、キャッシュメモリからのデータロードを考慮しないメモリアクセスに限定した場合の性能予測モデルである。ハードウェアの理論メモリバンド幅をB、理論ピーク性能をFとする。また、アプリケーションの要求フロップ値f、アプリケーションの要求バイト値bを用いたアプリケーションの演算強度を $X = f/b$ とすると、アプリケーションの実効性能は、 $\min\{F, BX\}$ で表される [3]。BXを、

$$BX = B \times \left(\frac{f}{b}\right) = \left(\frac{B}{F}\right) / \left(\frac{b}{f}\right) \times F \quad (1)$$

と変形すると、BXは、ハードウェアの持つ実効B/F値をアプリケーションの要求b/f値で除した値に理論ピーク演算性能を掛けた値とみることができる。また、両辺をFで除すればピーク性能比となる。本論文では、B、F、bおよびfを用いてアプリケーションの性能Pを

$$\min\left\{F, \left(\frac{B}{F}\right) / \left(\frac{b}{f}\right) \times F\right\} \quad (2)$$

アプリケーションのピーク性能比 $C_p$ を

$$\min\left\{1.0, \left(\frac{B}{F}\right) / \left(\frac{b}{f}\right)\right\} \quad (3)$$

で考える。

#### 4.2 ルーフラインモデルの検証

Ilicらは、キャッシュの効果を取り入れたルーフラインモデルを提案している [14]。その議論の中で複数のレベルのキャッシュについては、特別なプログラムを用いることで理論バンド幅が達成できるが、メモリについては、理論バンド幅は達成できないため、実測により実効バンド幅を求めている。我々は、FORTRAN言語を用いた一般的なプログラムで、L2キャッシュとメモリについての実効バンド幅性能を求めた。

この検証では、配列サイズを変化させることにより、すべての配列がメモリに載った状態 (onメモリ)、またはL2キャッシュに載った状態 (onL2) となるテストプログラムを用意した。メモリおよびL2キャッシュ基礎性能テストプログラムを図1(a)に示す。このプログラムを基本形として、要求バイト数を変えないように、図1(a)の代入文の右辺に対し、(右辺) \*c1(i,j)+zのように項を追加することにより演算数を増やし、要求b/f値を変化させた (図1(b))。このような手法を採用したのは、加算、乗算のSIMD演算器の有効利用を図り、演算に対する不要な性能低下要因を導入しないためである。jのループに対して

```

do j = 1,N      !(a)基本プログラム
  do i = 1,M
    c10(i,j) = z+c1(i,j)*x
  enddo
enddo

do j = 1,N      !(b)項を追加したプログラム例
  do i = 1,M
    c10(i,j) = ((z+c1(i,j)*x)* &
                c1(i,j)+z)* &
                c1(i,j)+z
  enddo
enddo
    
```

図1 メモリおよびL2キャッシュ基礎性能テストプログラム  
Fig. 1 Base performance test programs of memory and L2 cache.

表 1 メモリ基礎性能テストの結果

Table 1 Result of memory performance base test.

要求 b/f 値	予測性能 (peak 比)	実行時間 (sec)	実測性能 (peak 性能比)	メモリバンド幅 (GB/s)
0.5	0.72	3.34	0.7186	45.92
1.0	0.36	3.29	0.3647	46.61
2.0	0.18	3.32	0.1807	46.22
3.0	0.12	3.28	0.1220	46.81
4.0	0.09	3.30	0.0909	46.55
6.0	0.06	3.34	0.0599	46.02
12.0	0.03	3.34	0.0299	45.98

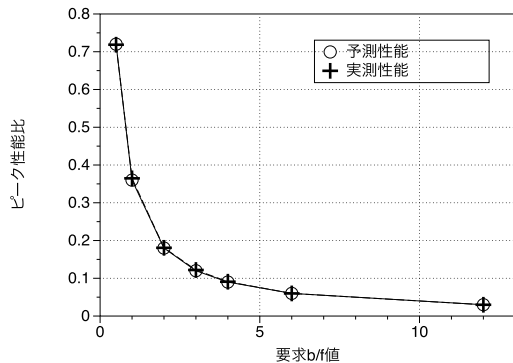


図 2 メモリ基礎性能テスト結果

Fig. 2 Result of base memory performance test.

は、ブロック分割による 8 スレッド並列化を行った。またループ内のすべての変数を on メモリとする場合は、配列サイズ  $M=8000000$ ,  $N=80$  を使用し、onL2 とする場合は、 $M=23000$ ,  $N=16$  を使用した。

まず、すべての変数が on メモリの場合のメモリ性能テストの結果を表 1 に示す。要求 b/f 値を 0.5 から 12.0 まで変化させたときの実行時間を計測し、それぞれの場合のメモリバンド幅を求め、これらのメモリバンド幅の値の平均値より、実効メモリバンド幅を 46.3 GB/s とした。

この実効メモリバンド幅と理論ピーク性能から実効 B/F 値を 0.36 とし、ルーフラインモデルによるピーク性能比の予測値を式 (3) を用いて計算した。表 1 に予測性能を示す。また予測性能と実測性能の比較グラフを図 2 に示す。この結果から、変数がメモリ上にある場合にはルーフラインモデルによる予測値が実測値とよく一致していることが確認された。

次に、L2 キャッシュ性能テストの結果を表 2 に示す。要求 b/f 値が 2.0 以上の L2 キャッシュのバンド幅の平均値から、L2 キャッシュの実効バンド幅を 159.2 GB/s とした。実効バンド幅と理論ピーク性能から、L2 キャッシュの実効 B/F 値は  $159.2/128 = 1.24$  となる。式 (3) より求めた性能予測値を表 2 の予測性能の欄に示した。また、予測性能と実測性能のグラフを図 3 に示す。

要求 b/f 値が 1.24 よりも大きい場合に、予測性能と実測性能がほぼ一致していることが分かる。要求 b/f 値が 1.24

表 2 L2 キャッシュ基礎性能テストの結果

Table 2 Result of L2cache performance base test.

要求 b/f 値	予測性能 (peak 比)	実行時間 (sec)	実測性能 (peak 性能比)	L2 キャッシュ バンド幅 (GB/s)
0.5	1.00	10.97	0.8806	56.36
1.0	1.00	5.91	0.8173	104.61
2.0	0.63	3.91	0.6176	158.12
3.0	0.42	3.87	0.4160	159.75
4.0	0.31	3.82	0.3161	161.84
6.0	0.21	4.04	0.1993	153.03
12.0	0.10	3.78	0.1065	163.56

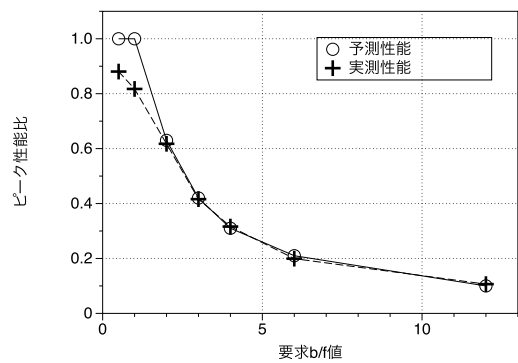


図 3 L2 キャッシュ基礎性能テスト結果

Fig. 3 Result of base L2-cache performance test.

より小さい場合は、予測性能がピーク性能になっている。特にチューニングされた DGEMM のような数学ライブラリでも実効性能としてピーク性能が得られることはなく [16]、ここでも同様に要求 b/f 値が 1.24 よりも小さい場合は、実測性能はピーク性能まで届いていない。このような場合、5.1 節に後述するように、演算器律速になっており、L2 キャッシュのバンド幅を使いこなすことができないため、L2 キャッシュバンド幅の測定値が低くなっている。このため、1.24 より小さい要求 b/f 値の値を L2 キャッシュの実効バンド幅の平均値の計算から除外した。

「京」における L2 キャッシュの理論バンド幅は、256 GB/sec である。したがって、ここで得られた L2 キャッシュの実効バンド幅、159.2 GB/s という結果は、キャッシュにおいては理論ピーク性能が得られるという Ilic らの主張とは異なる結果である。これは Ilic らが、メモリおよび各キャッシュ階層のバンド幅の測定において、特別なアセンブリコードを使用しているためと考える。

これらの結果から、L2 キャッシュにおいても L2 キャッシュのデータ供給能力に対し、演算器律速になっていない場合であれば、実効バンド幅を使用することにより、ルーフラインモデルが適用可能であることが明らかになった。

ここで得られた結果は、Ilic らの主張のように、メモリか L2 キャッシュいずれかにデータが載っている場合は、実効バンド幅を使用することでルーフラインモデルが適用可能であることを示している。またメモリバンド幅につい



```

do k=1,96
  do n=1,16769
    scl(n,k,1)=(
      &
      +cdiv(0,n,1,1)*vx(n ,k,1) &
      +cdiv(1,n,1,1)*vx(n+1 ,k,1) &
      +cdiv(2,n,1,1)*vx(n+131,k,1) &
      +cdiv(3,n,1,1)*vx(n+130,k,1) &
      +cdiv(4,n,1,1)*vx(n-1 ,k,1) &
      +cdiv(5,n,1,1)*vx(n-131,k,1) &
      +cdiv(6,n,1,1)*vx(n-130,k,1) &
      +cdiv(0,n,1,2)*vy(n ,k,1) &
      :
      +cdiv(0,n,1,3)*vz( n ,k,1) &
      :
    )*fact
  enddo
enddo

```

図 4 ルーフラインモデルによる性能評価があてはまらないカーネルプログラム

Fig. 4 A kernel loop to which the roofline model cannot be applied.

では、Ilic らの主張と同様に実効バンド幅と理論バンド幅に差異がある結果が得られたが、L2 キャッシュについては、lic らの主張と異なり、理論バンド幅と実効バンド幅に差異があるという結果が得られた。

### 4.3 ルーフラインモデルの適用限界について

ここでは、メモリのアクセスに比してキャッシュへのアクセスが増大するカーネルループを考える (図 4)。このループでは、メモリまでアクセスする配列変数は、ロードが vx, vy, vz の 3 個とストアが scl の 1 個で計 4 つである。scl については、配列アクセス数を 2 と計算する。

理由は、「京」の特別なオプションを使用する場合<sup>\*1</sup>を除いて、データのストアのために、一度データをキャッシュに読み込む必要があるためである。このため、scl へのアクセス数を 2 とし、メモリへのアクセス回数は合計 5 となる。一方、cdiv の配列サイズを、L2 キャッシュに常時キープされるサイズとすると、メモリアクセスと L2 キャッシュアクセスの比は 5 : 21 となり、メモリアクセスに比べ L2 キャッシュのアクセス比率が増大している。

このループに従来のルーフラインモデルを適用すると、倍精度浮動小数点数の演算数 43、変数サイズ 8 バイトであるから要求バイト数は、 $8 \times 5 = 40$ 、要求 b/f 値は  $40/43 = 0.93$  となり、メモリアの実効 B/F 値 0.36 を用いると、予測性能比は、 $0.36/0.93 = 0.39$  (39%) となる。しかし、実測によるピーク性能比は 14.7% しかなく、ルーフラインモデルによる予測性能比との間に大きな乖離がある。

このことからメモリアクセスとキャッシュアクセスの混合状態での限界性能の予測は、従来のメモリ性能だけによ

<sup>\*1</sup> 「京」では、ある条件の下、キャッシュにロードせずデータをストアするオプション XFILL がある。以降は、条件を揃えるため、すべて XFILL オプションを使用しないこととして議論を進める。

るルーフラインモデルでは難しいことが分かった。また本カーネルループのように、データがメモリおよび複数のレベルのキャッシュに分散して載っている場合は、Ilic らの方法は適用できない。

## 5. メモリとキャッシュ混合状態での性能限界値予測モデル

### 5.1 性能限界値予測モデルの概要

一般に、プログラムの CPU 単体性能は、演算数  $N_c$ 、CPU の理論ピーク性能  $P_{peak}$  としたとき、実行時間  $t_E$  は  $t_E > N_c/P_{peak}$  である。つまり実行時間は、 $P_{peak}$  という演算器の限界演算性能に律速されている。実際には、演算器だけでなく、様々なハードウェア要素により性能が抑えられる。

本論文で対象とする「京」では、メモリ、L2 キャッシュ、L1 キャッシュ、演算器の 4 つの構成要素について、それぞれが限界性能を持つとし、それらの限界性能に律速された状態での実行時間を以下で表す。

- $t_M$  : メモリ律速により決定される実行時間
- $t_{L2}$  : L2 キャッシュ律速により決定される実行時間
- $t_{L1}$  : L1 キャッシュ律速により決定される実行時間
- $t_C$  : 演算器律速により決定される実行時間

このとき、実行時間は、次の 4 つのケースとなると仮定する。

- ケース 1: メモリ律速あり、L2 および L1 キャッシュ律速なし、演算律速なしの場合の実行時間は  $t_M$  である。
- ケース 2: メモリ律速なし、L2 キャッシュ律速あり、L1 キャッシュ律速なし、演算律速なしの場合の実行時間は  $t_{L2}$  である。
- ケース 3: メモリ律速なし、L2 キャッシュ律速なし、L1 キャッシュ律速あり、演算律速なしの場合の実行時間は  $t_{L1}$  である。
- ケース 4: メモリ律速なし、L1 および L2 キャッシュ律速なし、演算律速ありの場合の実行時間は  $t_C$  である。

上記の仮定に従うとプログラムの実行時間  $t_E$  は、4 つの構成要素により律速される実行時間のうち、最大値になるものと考えられ、次の式で与えられる。

$$t_E = \max\{t_M, t_{L2}, t_{L1}, t_C\} \tag{4}$$

「京」において、性能に問題のあるプログラムで、何もチューニングされていない場合は、以下の状態となっていることが多い。

- ソフトウェアパイプラインニングや SIMD 化等の命令スケジューリングの最適化が不十分である。
- キャッシュスラッシングを原因としたキャッシュミスが頻発している。
- TLB スラッシングを原因とした TLB ミスが頻発している。

- 演算器が最適に利用できていない。

このような性能問題をかかえたプログラムは、限界性能を達成することが不可能であり、式 (4) で示した実行時間を実現できず、性能限界値を実現するためには、対象のプログラムをチューニングする必要がある。したがって、性能限界値を求めるためには、律速要因となる構成要素、律速時の実行時間を考えることが必要である。

本研究では、メモリバンド幅の実効値により律速されるメモリ律速、L2 キャッシュバンド幅の実効値により律速される L2 キャッシュ律速、および演算速度の実効値により律速される演算器律速の 3 つを考慮し、L1 キャッシュによる律速条件は考慮しない。L1 キャッシュは、データアクセス機構の差異により、メモリや L2 キャッシュと以下の点で違いがあるためである。

- SIMD ロードと非 SIMD ロードによりデータ転送速度が大きく異なる。
- キャッシュミスのパナルティ、特にストア時のパナルティが大きい。
- L2 キャッシュから L1 キャッシュへのアクセスはキャッシュライン単位であるが L1 キャッシュからレジスタへのアクセスはキャッシュライン単位ではない。

このため、L1 キャッシュの動作は複雑であり、 $t_{L1}$  を予測することが困難である。

そこで、式 (4) の中から、L1 キャッシュの項を取り除き、メモリアクセスと L2 キャッシュアクセスが混合している場合の限界性能予測モデルを提案する。L1 キャッシュについては、6.1 節において、それが律速となるかどうかの判定条件を示し、モデルの適用条件を明らかにする。

### 5.2 限界性能予測モデル

実効性能は、プログラムで実行される演算量を実行時間で除いた値であるので、実行時間の予測ができれば限界性能の予測が可能となる。

メモリ、L2 キャッシュ、演算に対して、メモリデータ転送量、実効メモリバンド幅、L2 キャッシュデータ転送量、実効 L2 キャッシュバンド幅、プログラム演算量、演算実効性能を、それぞれ  $D_M$ ,  $B_M$ ,  $D_{L2}$ ,  $B_{L2}$ ,  $N_c$ ,  $P_{eff}$  とすると、メモリ律速時の実行時間  $t_M$ 、L2 キャッシュ律速時の実行時間  $t_{L2}$ 、演算器律速時の実行時間  $t_C$  は、以下の式で計算できる。

$$t_M = D_M/B_M \tag{5}$$

$$t_{L2} = D_{L2}/B_{L2} \tag{6}$$

$$t_C = N_c/P_{eff} \tag{7}$$

したがって、プログラムの実行時間  $t_E$ 、ピーク性能比  $C_P$  は、以下で計算される。

$$t_E = \max\{t_M, t_{L2}, t_C\} \tag{8}$$

$$C_P = N_c/(\max\{t_M, t_{L2}, t_C\} \times P_{peak}) \tag{9}$$

ループ内のメモリアクセスする倍精度配列変数の個数を  $m$ 、ループ内の L2 キャッシュアクセスする倍精度配列変数の個数を  $n$ 、ループ内の演算の個数を  $l$  とすると、アプリケーションのピーク性能比  $C_P$  は、メモリ律速領域 ( $t_M > t_{L2}$  の領域)、L2 キャッシュ律速領域 ( $t_M < t_{L2}$  の領域) それぞれの場合に以下の式で求められる。

(1) メモリ律速:  $t_M > t_{L2}$  の領域

$$C_P = \left( \frac{B_M}{P_{peak}} \right) / \left( \frac{8m}{l} \right) \tag{10}$$

(2) L2 キャッシュ律速:  $t_M < t_{L2}$  の領域

$$C_P = \left( \frac{B_{L2}}{P_{peak}} \right) / \left( \frac{8(m+n)}{l} \right) \tag{11}$$

ただし、L2 キャッシュ律速の場合には、メモリアクセスするデータは L2 キャッシュにもアクセスするため、L2 キャッシュアクセスする変数の個数を  $(m+n)$  とした。

### 5.3 「京」での限界性能予測モデルの検証

混合性能テストとして、配列がメモリと L2 キャッシュの両方に載った混合状態のテストプログラムを作成し、メモリと L2 キャッシュに載る配列のサイズ、演算数を変化させ、各々のプログラムを 60 回実行し実行時間を測定した。また、 $k$  のループについては、ブロック分割による 8 スレッド並列化を行った。測定に用いたプログラムを図 5 に示す。

図 5 のプログラムにおいて、配列  $c(N1, N2, N3)$ 、 $N1=4000, N2=60, N3=80$  とすると、 $c(i, j-1, k)$ ,  $c(i, j, k)$ ,  $c(i, j+1, k)$  のうち 1 つはメモリからのロードとなるが、他の 2 つについてはキャッシュに載っている。配列  $c$  の 1 次元目の宣言が、1 から 4000 であるので、配列  $c$  の第 2 添え字に関しては、 $4000 \times 8 = 32\text{kB}$  離れており、ループ内に  $a$  と  $c$  の 2 つの配列アクセスが存在することを考えると、 $c(i, j+1, k)$  は L1 キャッシュには載りきらず L2 キャッシュアクセスとなる。

メモリ上の  $a(i, j, k)$  へのアクセス数を 2 とすると、メモリへの要求バイト数  $3 \times 8 = 24$  (B)、L2 キャッシュへの要求バイト数  $2 \times 8 = 16$  (B)、演算数は 2 となるので、メモリへの要求  $b/f$  値は 12、L2 キャッシュへの要求  $b/f$

```
do k = 1, N3
  do j = 1, N2
    do i = 1, N1
      a(i, j, k) = c(i, j-1, k) + c(i, j, k) * c(i, j+1, k)
    enddo
  enddo
enddo
```

図 5 メモリ・L2 キャッシュ混合性能テストプログラム  
Fig. 5 Performance test program for mixed access to both memory and L2 cache.

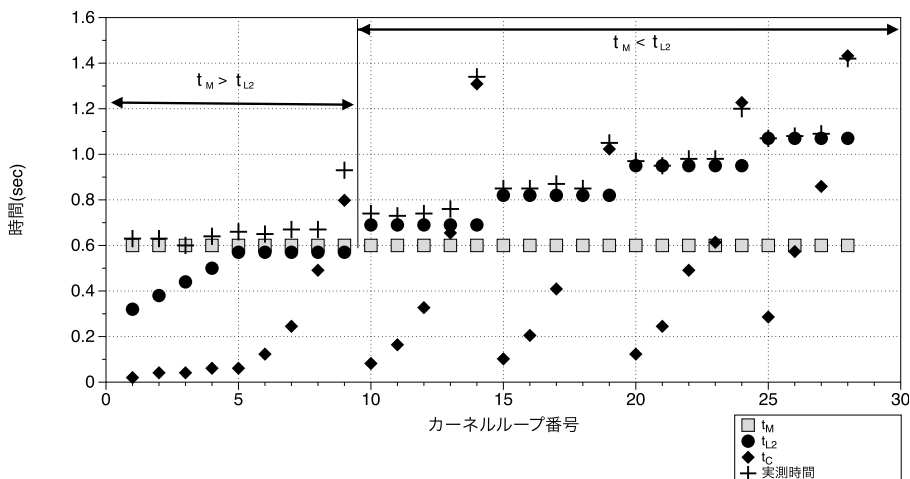


図 6 メモリ・L2 キャッシュ混合アクセス性能テスト結果

Fig. 6 Result of memory and L2 cache mix access performance test.

表 3 メモリ・L2 キャッシュ混合アクセス性能テストケース

Table 3 Result of memory and L2 cache mix access performance test.

カーネル ループ番号	カーネル ループ名	カーネル ループ番号	カーネル ループ名
1	3M-2L2-2F	15	3M-10L2-10F
2	3M-3L2-4F	16	3M-10L2-20F
3	3M-4L2-4F	17	3M-10L2-40F
4	3M-5L2-6F	18	3M-10L2-80F
5	3M-6L2-6F	19	3M-10L2-100F
6	3M-6L2-12F	20	3M-12L2-12F
7	3M-6L2-24F	21	3M-12L2-24F
8	3M-6L2-48F	22	3M-12L2-48F
9	3M-6L2-78F	23	3M-12L2-60F
10	3M-8L2-8F	24	3M-12L2-120F
11	3M-8L2-16F	25	3M-14L2-28F
12	3M-8L2-32F	26	3M-14L2-56F
13	3M-8L2-64F	27	3M-14L2-84F
14	3M-8L2-128F	28	3M-14L2-140F

値は 8 となる。3 変数のメモリへのアクセス, 2 変数の L2 キャッシュへのアクセス, 演算が 2 であるので, カーネルループ名を 3M-2L2-2F と呼ぶこととし, このプログラムをベースプログラムとした。

図 5 に対し, 4 行目の代入文の右辺の式に, (右辺)\*c(i, j+2, k)+c(i, j-2, k) のように項を追加することにより, L2 キャッシュのアクセスと演算数を増加させた。メモリアクセス数, L2 キャッシュアクセス数, 演算数の比を m : n : 1 とした場合のカーネルループを mM-nL2-1F と呼ぶことにする。

測定に用いたカーネルループ一覧を表 3 に示す。これらは, メモリへのアクセス数を固定し, L2 キャッシュへのアクセス数, 演算数を変化させたものである。カーネルループ番号の増加とともに, L2 キャッシュアクセス量および演算量が増加している。ループ番号 5-9, 10-14, 15-19,

20-24, 25-28 は, それぞれ L2 キャッシュへのアクセス量を 6, 8, 10, 12, 14 とした。

図 6 において, 実効バンド幅  $B_M$ ,  $B_{L2}$  は, それぞれ 46 GB/s, 146 GB/s とした。これらは, 各ループのバンド幅の最高値である。 $B_{L2}$  は, 4.2 節の L2 キャッシュ基礎テストで得られた 159 GB/s より小さい。これは, 「京」では, メモリと L2 キャッシュについて同一のハードウェアでデータ転送を受け持つようになっており, 互いのバンド幅が影響を受けるため, メモリアクセスをしない L2 キャッシュ基礎テストの値より小さくなるためと考えられる。また実効演算性能  $P_{eff}$  は, 表 2 の b/f 値 0.5 で得られた実測性能 (peak 比) 0.88 を用い,  $P_{eff} = P_{peak} \times 0.88$  で計算した。

図 6 から,  $t_M > t_{L2}$  となっているカーネルループ 1-9 では, 実測時間はほぼ  $t_M$  に一致し, また,  $t_M < t_{L2}$  となっているカーネルループ 10-28 では, 実測時間は, ループ 9, 14, 19, 24, 28 を除き, ほぼ  $t_{L2}$  に一致していることが分かる。ループ 9, 14, 19, 24, 28 は,  $t_C > t_{L2}$  となっており, 実測時間は  $t_C$  に近い。

ここで,  $t_M > t_{L2}$  から  $t_M \leq t_{L2}$  に切り替わる点は, 以下の式を満たす。

$$\frac{m \times (\text{ループ回転数})}{B_M} = \frac{(m+n) \times (\text{ループ回転数})}{B_{L2}} \quad (12)$$

これを  $n$  について解くと,

$$n = \left( \frac{B_{L2}}{B_M} - 1 \right) \times m = 2.17m \quad (13)$$

となる。 $m$  は 3 であるので,  $n$  は 6.34 となり, 表 3 から, この境目はループ 9 とループ 10 の間である。図 6 の実測値とも一致している。

次に, 従来のルーフラインモデルと提案した予測モデルの比較を行う。図 7 は, ピーク性能比のルーフラインモデルによる予測値と実測値をプロットしたものである。横軸

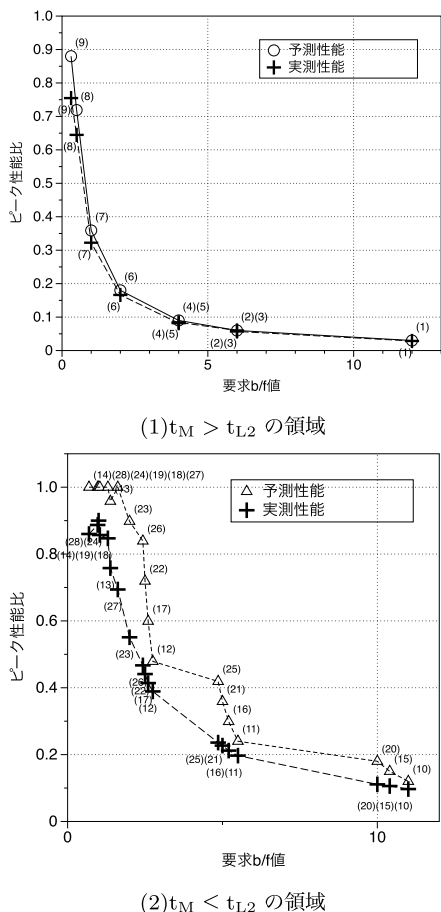


図 7 メモリ・L2 キャッシュ混合アクセステスト結果 (従来予測モデル)  
**Fig. 7** Result of memory and L2 performance test (Traditional predictive model).

の  $b/f$  値は、メモリアクセス量と演算量の比  $8 \times m/l$  とした。また、図中の括弧つきの数字は、カーネルループ番号である。図 7(1) は、メモリバンド幅律速の場合 ( $t_M > t_{L2}$ ) の場合であるが、予測値と実測値がよく一致していることが分かる。一方、図 7(2) は、カーネルループ 10-28 についてのものであるが、 $t_M \leq t_{L2}$  の領域であるため、従来のループラインモデルによる予測値と実測値の乖離があることが分かる。

図 8 は、 $t_M \leq t_{L2}$  の領域に提案した予測モデルを適用した結果を示す。横軸の  $b/f$  値は、L2 キャッシュアクセス量と演算量の比  $8 \times (m+n)/l$  とした。また、図中の括弧つきの数字は、カーネルループ番号である。図 8 は、予測値と実測値がよく一致していることを示している。

図 7, 図 8 において、予測性能値が 80% を超える領域については、演算器律速になる領域であるため、予測性能の精度は下がっている。

5.4 Xeon プロセッサに対する性能モデルの評価

Intel Xeon を用いて提案した性能予測モデルの適用可能性について検討した。計測に用いた環境を表 4 に示す。

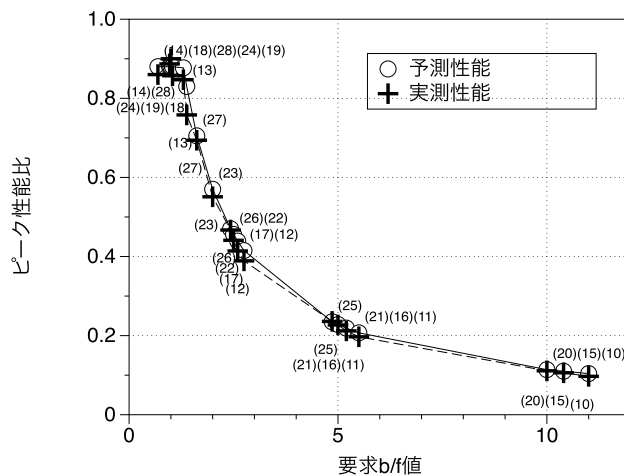


図 8 メモリ・L2 キャッシュ混合アクセス性能テスト結果  
**Fig. 8** Result of memory and L2 performance test (Proposed predictive model).

表 4 Intel Xeon (Haswell) の仕様  
**Table 4** Specification of Intel Xeon (Haswell).

周波数	2.3 GHz	SIMD 幅	256 bit
コア数	12	ソケット性能	441.6 Gflops
演算器タイプ	Multi and Add 2 バイブ	メモリバンド幅	68 GB/s
L1D cache サイズ	32 KB	L1D cache レイテンシ	4cycle
L1D cache バンド幅 (load)	64 B/cycle	L1D cache バンド幅 (store)	32 B/cycle
L2D cache サイズ	256 KB	L2D cache レイテンシ	11cycle
L2D cache バンド幅	64 B/cycle	L3 cache サイズ	30 MB

Haswell では、L1 キャッシュ、L2 キャッシュ、L3 キャッシュがある。L1 キャッシュと L2 キャッシュはコアごとに装備されており、30 MB の L3 キャッシュは、2.5 MB ずつ 12 個のコアに割り当てられ、それらがリング状の構成で共有されている [17]。メモリに近い共有キャッシュとして、L3 キャッシュが「京」の L2 キャッシュと同様の位置づけと考え、Haswell では、L3 キャッシュに対する評価を行った。

まず、メモリおよび L3 キャッシュの基礎性能テストにより、実効メモリバンド幅は 49 GB/s が得られた。これは理論バンド幅の約 72% である。また、要求  $b/f$  値 2.0 以上については、予測どおりの演算性能が得られた。要求  $b/f$  値 0.250 についての予測性能と実測性能は、それぞれ 100%、72.0% であった。要求  $b/f$  値 0.5 については、それぞれ 100%、72.5%、要求  $b/f$  値 1.0 については、それぞれ 75.1%、58.7% であった。実効 L3 キャッシュバンド幅は、331.8 GB/s が得られた。

次に、L3 キャッシュアクセス量と L3 バンド幅の関係を測定した。図 5 のプログラムに対し、 $N1=12800$ ,  $N2=60$ ,



N3=36 とし、最外側ループ k をブロック分割により 12 コアでスレッド並列化した。各スレッドの担当分の k のサイズは 3 である。N1=12800 なので、第 2 次元の j を 1 増やすと、102.4KB の L3 キャッシュを使用することとなる。

測定したテストケースの一覧を表 5 に示す。各カーネ

表 5 メモリ・L2 キャッシュ混合アクセス性能テストケース  
Table 5 Result of memory and L2 cache mix access performance test.

カーネル ループ番号	カーネル ループ名	カーネル ループ番号	カーネル ループ名
1	3M-2L3-2F	11	3M-18L3-18F
2	3M-3L3-4F	12	3M-20L3-20F
3	3M-4L3-4F	13	3M-22L3-22F
4	3M-5L3-6F	14	3M-24L3-24F
5	3M-6L3-6F	15	3M-26L3-26F
6	3M-8L3-8F	16	3M-28L3-28F
7	3M-10L3-10F	17	3M-30L3-30F
8	3M-12L3-12F	18	3M-32L3-32F
9	3M-14L3-14F	19	3M-34L3-34F
10	3M-16L3-16F	20	3M-36L3-36F
		21	3M-38L3-38F

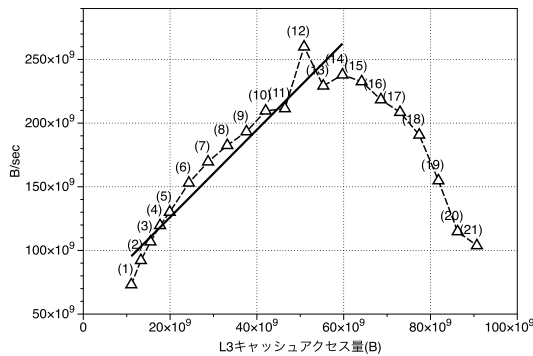


図 9 Haswell のメモリ・L3 キャッシュ混合時の L3 キャッシュバンド幅

Fig. 9 L3 band width of L3 cache and memory mix access test on Haswell.

ループは、L3 キャッシュのアクセス量が異なっており、L3 キャッシュのアクセス数 × 102.4kB が L3 キャッシュアクセス量となる。カーネルループ 14 では、102.4kB × 24 = 2.46 MB となり、各コアの L3 キャッシュ容量が一杯となる点と考えられる。

図 9 に、各ケースごとの L3 キャッシュバンド幅の測定結果を示す。「京」では L2 キャッシュアクセス量によらず L2 キャッシュバンド幅はほぼ一定であったが、Haswell では、L3 キャッシュが一杯になるループ 14 まで L3 キャッシュアクセス量に応じて L3 キャッシュバンド幅が増大している。さらに L3 キャッシュアクセス量が増加すると、L3 キャッシュの溢れを起こすために L3 キャッシュバンド幅は低下している。

以下では、カーネルループ 1-14 について、L3 キャッシュアクセス量と L3 キャッシュバンド幅の関係が線形関係にあると仮定した。線形フィッティングにより、実効 L3 キャッシュバンド幅  $B_{L3}$  は、

$$B_{L3} = 3.332 \times D_{L3} + 5.745 \times 10^{10} \quad (14)$$

となった。ここで、 $D_{L3}$  は L3 キャッシュアクセス量である。この直線も図 9 にプロットした。

式 (14) を使用し、L3 キャッシュバンド幅を L3 キャッシュアクセス量から計算し、5.2 節と同様の方法で、 $t_M$ 、 $t_{L3}$ 、 $t_C$  を求めた。図 10 にこれらの値をプロットした。 $t_C$  の計算に用いた  $P_{eff}$  は、L3 キャッシュ基礎性能テスト時に得られたピーク演算性能比 72% を使用した。

図 10 から、L2 キャッシュと L3 キャッシュの違いがあるものの、L3 キャッシュバンド幅を L3 キャッシュアクセス量に対する線形関係とすることにより、Haswell においても提案モデルにより性能予測が可能であることが分かった。

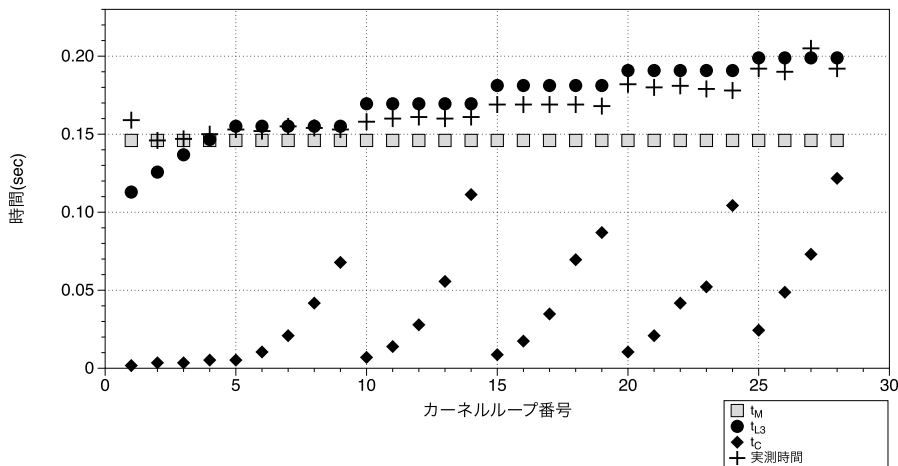


図 10 メモリ・L3 キャッシュ混合アクセス性能テスト結果 (Haswell)

Fig. 10 Result of memory and L3 cache mix access performance test (Haswell).

```

do k = 1,N3
  do j = 1,N2
    do i = 1,N1
      a(i,j,k) = c(i-1,j,k)+c(i,j,k)*c(i+1,j,k)
    enddo
  enddo
enddo

```

図 11 L1 キャッシュ性能テストプログラム a)  
Fig. 11 L1 cache test code a).

```

do k = 1,N3
  do j = 1,N2
    do i = 1,N1
      a(i,j,k) = (c(i-1,j,k)+c(i,j,k)*c(i+1,j,k))* &
        c(i-12,j,k)+c(i+12,j,k)
    enddo
  enddo
enddo

```

図 12 L1 キャッシュ性能テストプログラム b)  
Fig. 12 L1 cache test code b).

```

do k = 1,N3
  do j = 1,N2
    do i = 1,N1
      a(i,j,k) = (((((((c(i+1,j,k)+c(i,j,k)*c(i-1,j,k))* &
        c(i+12,j,k)+c(i-12,j,k))* &
        c(i,j-1,k)+c(i,j+1,k))* &
        c(i,j-2,k)+c(i,j+2,k))* &
        c(i,j-3,k)+c(i,j+3,k))* &
        c(i,j-4,k)+c(i,j+4,k))* &
        c(i,j-5,k)+c(i,j+5,k))* &
        c(i,j-6,k)+c(i,j+6,k))* &
        c(i,j-7,k)+c(i,j+7,k)
    enddo
  enddo
enddo

```

図 13 L1 キャッシュ性能テストプログラム c)  
Fig. 13 L1 cache test code c).

## 6. 「京」における具体的な性能予測手法

### 6.1 L1 キャッシュ律速の判定条件

ここでは 5.1 節に述べたように L1 キャッシュについては、 $t_{L1}$  の推定が困難なため「京」についての L1 キャッシュ律速とならない条件について、特に本論文では、ステンスル計算を意識した場合について検討する。ステンスル計算によく見られるような以下の 3 種類のテストプログラムにより評価を行った。

- a) 図 5 ベースで 1 次元目を差分としたプログラム (図 11)
  - b) a) と同様であるが差分間隔が a) とは異なるプログラム (図 12)
  - c) b) と同様であるがメモリのほかに L2 キャッシュにアクセスするプログラム (図 13)
- a) のプログラムは L1 キャッシュのアクセスを、 $c(i+1,j,k)$ ,  $c(i+2,j,k)$  のように 1 つ単位で増加させており、差分間隔が short なプログラムである。b) のプログラムは L1 キャッシュのアクセスを、 $c(i+12,j,k)$ ,  $c(i+24,j,k)$  のように 12 単位で増加させており、差分間隔が long なプログラムである。c) のプログラムは b) と同様であるが、14 要素分の L2 キャッシュにアクセスするプ

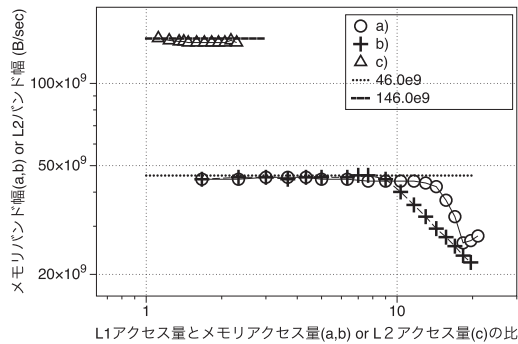


図 14 L1 キャッシュ律速条件の評価結果  
Fig. 14 estimation result of L1 cache bounding.

ログラムである。a) はメモリと short でアクセスする L1 キャッシュの混合状態、b) はメモリと long でアクセスする L1 キャッシュの混合状態、c) はメモリ、L2 キャッシュと long でアクセスする L1 キャッシュの混合状態における、L1 キャッシュの律速条件を検証するプログラムである。

図 14 に 3 つのテストプログラムの実行結果を示す。横軸は、プログラム a), b) では、メモリアクセス量と L1 キャッシュアクセス量の比を表し、プログラム c) では、L2 キャッシュアクセス量と L1 キャッシュアクセス量の比を表す。縦軸は、プログラム a), b) では、測定された実効メモリバンド幅、プログラム c) では、測定された実効 L2 キャッシュバンド幅を表す。

差分間隔が short なアクセス a) については、メモリのアクセス量に対し 10 倍以上まで、差分間隔が long なアクセス b) については、メモリのアクセス量に対し 8 倍程度まで、L1 キャッシュアクセスする配列を増やしても、メモリ律速状態でのルーフラインモデルによる性能予測が適用可能であることが確認された。また、差分間隔が long なアクセス、かつ L2 キャッシュのバンド幅を使いきっているプログラム c) についても、L2 キャッシュのアクセス量に対し 2 倍程度まで、L1 キャッシュアクセスする配列を増やしても、L2 キャッシュ律速状態でのルーフラインモデルによる性能予測が適用可能であることが確認された。

ここでの議論により、L1 キャッシュについて以下の条件を提示する。また、以下はすべて SIMD ロードが条件となっている。

- (1) メモリ律速の場合、a) の short タイプの差分については、メモリアクセス量に対し 10 倍程度まで L1 キャッシュアクセスする配列を増やしてもメモリアクセス律速でのルーフラインモデルの適用が可能である。
- (2) メモリ律速の場合、b) の long タイプの差分については、メモリアクセス量に対し 8 倍程度まで L1 キャッシュアクセスする配列を増やしてもメモリアクセス律速でのルーフラインモデルの適用が可能である。
- (3) L2 キャッシュ律速の場合、c) の long タイプの差分については、L2 キャッシュのアクセス量に対し 2 倍程

度まで L1 キャッシュアクセスする配列を増やしても L2 キャッシュアクセス律速でのルーフラインモデルの適用が可能である。

## 6.2 「京」における性能予測方法

前節までの議論をまとめると L1 キャッシュ律速条件を考慮した場合の性能上限値の予測方法は、以下のとおりである。ただし、個数は 8 バイト単位で計算する。

(1) プログラムのメモリ、L2 キャッシュ、L1 キャッシュアクセスの以下のデータ量および演算数を求める。

- a) メモリまでアクセスする配列の個数 ( $n_M$ )
- b) L2 キャッシュまでアクセスする配列の個数 ( $n_{L2}$ )
- c) L1 キャッシュまでアクセスする個数のうち short アクセスする個数 ( $n_{L1s}$ )
- d) L1 キャッシュまでアクセスする個数のうち long アクセスする個数 ( $n_{L1l}$ )
- e) 演算数 ( $l$ )

(2) メモリ律速か L2 キャッシュ律速かを判断する。

- a)  $n_{L2} \leq ((B_{L2}/B_M) - 1) \times n_M$  の場合、メモリ律速
- b)  $n_{L2} > ((B_{L2}/B_M) - 1) \times n_M$  の場合、L2 キャッシュ律速

(3) メモリ律速の場合、L2 キャッシュ律速の場合それぞれについて、L1 キャッシュアクセスの影響について、L1 キャッシュのアクセス量が限界まで達していないことを確認する。限界を超えていない場合は、 $n_{L1s}$ 、 $n_{L1l}$  の影響を考慮しなくてもよい。それぞれの場合について、ピーク性能比を予測する。

a) メモリ律速の場合

SIMD ロードで  $n_{L1s} < 10n_M$ 、かつ  $n_{L1l} < 8(n_M + n_{L2})$  であることを確認する。成立する場合は L1 キャッシュへのアクセスは考慮しない。成立しない場合は今回の予測モデルの対象外である。そのうえで、メモリアクセス律速での予測とし、ピーク性能比  $C_p = (B_M/P_{peak})/(n_M \times 8/l)$  とする。

b) L2 キャッシュ律速の場合

SIMD ロードで、 $n_{L1l} < n_M + n_{L2}$  であることを確認する。成立する場合は L1 キャッシュへのアクセスは考慮しない。成立しない場合は今回の予測モデルの対象外である。そのうえで、L2 キャッシュ律速での予測とし、ピーク性能比  $C_p = (B_{L2}/P_{peak})/((n_M + n_{L2}) \times 8/l)$  とする。

## 7. 実アプリケーション性能予測と評価

2 章に示したように、本論文では、ステンシル計算や疎行列ベクトル積といった一般的なプログラムに適用可能な性能予測手法の提案を目的としている。6.1 節に示した L1 キャッシュ律速の判定条件は、ステンシル計算を前提としたものである。したがって、本章ではステンシル計算を対

象として、「京」上で実アプリケーションの性能を向上させた。プロセッサの単体性能をできる限り向上させるチューニングの例として、図 4 のプログラム ((A) とする) を含む 4 つのプログラムに対し、本モデルによる性能予測と実測性能を基に、プログラムのチューニングを行った結果について述べる。今回の性能予測の目的は、限界性能を予測することであり、実際のアプリケーションで必ずしも限界性能値を得られるわけではない。そこで今までの経験に基づきチューニングによる性能向上の目標は、実測性能が予測性能の 85% 程度を達成することとした。性能予測と評価のサマリを表 6 に示す。性能予測のパラメータは、5.3 節で測定された値を使用した。

(1)  $n_{L2} \leq 2.17n_M$  の場合

$$C_p = (46/128)/(n_M \times 8/l) = 0.36/(n_M \times 8/l)$$

(2)  $n_{L2} > 2.17n_M$  の場合

$$C_p = (146/128)/((n_M + n_{L2}) \times 8/l) = 1.14/((n_M + n_{L2}) \times 8/l)$$

プログラム (A) は、 $n_{L2} > 2.17n_M$  が成り立つので L2 キャッシュ律速状態での予測となる。また、 $n_{L1l} < n_{L2} + n_M$  の条件と、SIMD ロードの条件を満たすので L1 のアクセスは考慮する必要はない。したがって、予測ピーク性能比は、 $C_p = 1.14/((5 + 21) \times 8/43) = 0.236$  (23.6%) となる。実際にこのプログラムを計測すると、ピーク性能比は 14.7% であり、予測値より低い値となっていた。性能劣化の調査を行ったところ、L1 キャッシュミス dm 率が 13.36% となっていることが分かった。L1 キャッシュミス dm 率は、3 つの L1 キャッシュミス\*2 のうちの、データアクセス要求 (dm) 時に発生する割合を示す数値であり、この値が大きい場合は、キャッシュラッシングの可能性が疑われる。このため、配列変数のマージを実施した [18], [19]。この結果、L1 キャッシュミス dm 率は 3.85% まで低下し、実測性能は 19.7% まで向上した。この値は予測性能との差が 15% 以内となったため、チューニングを終了した。本例は、L2 キャッシュ律速状態になると予測された。プログラムチューニング後は、L2 キャッシュバンド幅を、95.7 GB/s から 121.8 GB/s に向上させてあるので、L2 キャッシュバンド幅の限界値である 146 GB/s に近づいている一方、メモリバンド幅は、限界値の 46 GB/s に遠く達していない。このことは 5.1 節に述べた、式 (4) の  $t_{L2}$  の項が最大値になっていることを意味しており、5.1 節の議論を裏付けるものとなっている。

プログラム (B) は、 $n_{L2} \leq 2.17n_M$  が成り立つので、メモリ律速状態の予測となる。また、 $n_{L1s} < 10n_M$ 、および  $n_{L1l} < 8(n_{L2} + n_M)$  の条件と、SIMD ロードの条件を満たすので L1 のアクセスは考慮する必要はない。

\*2 L1 キャッシュミスは、データアクセス要求 (dm) 時、ハードウェアプリフェッチ時、ソフトウェアプリフェッチ時で 3 つの場合に発生する。

表 6 4つのプログラムの性能予測結果  
Table 6 Result of performance prediction of four sample codes.

	予測性能					tuning 前 (バンド幅: GB/s)			tuning 後 (バンド幅: GB/s)				
	$n_M$	$n_{L2}$	$n_{L1s}$	$n_{L1l}$	$l$	従来法 (peak 比)	提案法 (peak 比)	実測性能 (peak 比)	メモリ バンド幅	L2cache バンド幅	実測性能 (peak 比)	メモリ バンド幅	L2cache バンド幅
(A)	5	21	12	6	43	0.387	0.236	0.147	23.0	95.7	0.197	29.7	121.8
(B)	13	2	3	15	60	0.208	0.208	0.129	29.6	55.8	0.193	43.7	70.7
(C)	11	2	0	2	11	0.045	0.045	0.038	45.6	35.8	-	-	-
(D)	3	8	8	0	25	0.375	0.324	0.290	38.4	128.5	-	-	-

メモリ律速状態の予測式を使い、予測ピーク性能比は、 $C_p = 0.36 / (13 * 8 / 60) = 0.208$  (20.8%) となる。チューニング前の実測性能比は 12.9% であり、予測性能値に達していなかった。L1 キャッシュミス dm 率が 15.11% であったため、配列変数のマージのチューニングにより、L1 キャッシュミス dm 率も 8.81% まで低下し、実測性能が 19.3% と予測性能近くまで性能を向上させることができた。本例は、メモリ律速状態になると予測された。チューニング後は、メモリバンド幅を、29.6 GB/s から 43.7 GB/s に向上させてあるので、メモリバンド幅の限界値である 46 GB/s に近づいている一方、L2 キャッシュバンド幅は、限界値の 146 GB/s に遠く達していない。このことは 5.1 節に述べた、式 (4) の  $t_M$  の項が最大値になっていることを意味しており、5.1 節の議論を裏付けるものとなっている。

プログラム (C) は、 $n_{L2} \leq 2.17n_M$  が成り立つので、メモリアクセス律速状態である。 $n_{L1s} < 10n_M$ 、および  $n_{L1l} < 8(n_{L2} + n_M)$  の条件と SIMD ロードの条件を満たすので、L1 キャッシュのアクセスは考慮しない。予測ピーク性能比は、 $0.36 / (11 * 8 / 11) = 0.045$  (4.5%) となり、また実測性能値は 3.81% である。L1 キャッシュミス dm 率も 0.31% と低く、またメモリバンド幅を見ると、限界値である 46 GB/s にほとんど達しているため、ほぼ性能の限界値に達していると考えられる。メモリバンド幅と L2 キャッシュバンド幅の値をみると、先の 2 例と同様に、5.1 節の議論を裏付けるものとなっている。

プログラム (D) は、 $n_{L2} > 2.17n_M$  が成り立つので L2 キャッシュアクセス律速状態での予測となる。また、 $n_{L1l} < n_{L2} + n_M$  の条件と SIMD ロードの条件を満たすので、L1 キャッシュのアクセスは考慮しない。予測ピーク性能比は、 $1.14 / ((3 + 8) * 8 / 25) = 0.324$  (32.4%)、実測性能値は 29.0%、L2 キャッシュバンド幅は、128.5 GB/s となっており限界値に近い値である。予測値と実測値の差が 15% 以内であるため、チューニングは行わないこととした。メモリバンド幅と L2 キャッシュバンド幅の値をみると、先の 3 例と同様に、5.1 節の議論を裏付けるものとなっている。

疎行列ベクトル積については、節点のリオーダーリングにより、間接参照を使用したランダムアクセスを行うベクトルを、L1 キャッシュに載せる方法が可能である。これによ

り、従来のループラインモデルの理論メモリバンド幅の代わりに実効メモリバンド幅を使用すれば、アプリケーションの性能を、よく予測できることが確認できており、メモリ律速の性能予測手法が適用できることがすでに分かっている [4], [20]。

## 8. まとめ

本報告では、すべての変数が「京」の L2 キャッシュにある場合にも、ループラインモデルにより性能予測ができることを示した。また、ループラインモデルを拡張したメモリとキャッシュアクセスが混在している状態での性能予測モデルを提案し、「京」上で提案した性能予測モデルが成り立つことを確認した。最後に性能予測モデルを実際のプログラムに適用し、提案した性能予測モデルでプログラムの限界性能値を予測し、チューニング実施の判断基準にすることができることを示した。

プログラムをチューニングする場合、限界性能を予測することはきわめて重要である。提案モデルにより、キャッシュアクセスがある場合でも性能予測ができるようになり、チューニングの際の 1 つの指標として用いることができる。この性能予測モデルを他のアーキテクチャに適用してゆくことが必要と考えている。

謝辞 本報告に際しご討論いただき貴重な助言をいただいた、富士通株式会社の青木正樹氏、井上晃氏をはじめとした性能評価チームの皆様、理化学研究所 AICS 運用技術部門ソフトウェア技術チームの諸氏に感謝いたします。本報告の結果は、理化学研究所のスーパーコンピュータ「京」を利用して得られたものである。

## 参考文献

- [1] Frigo, M. and Strumpfen, V.: Cache Oblivious Stencil Computation, *ICS '05 Proceedings of the 19th annual international conference on Supercomputing*, pp.361-366, ACM New York, NY, USA (2005).
- [2] 近藤正章, 岩本 貢, 中村 宏: キャッシュラインを考慮した 3 次元 PDE solver の最適化手法, 情報処理学会研究報告, 計算機アーキテクチャ研究会報告, Vol.2001, No.22, pp.91-96 (2001).
- [3] Williams, S., Waterman, A. and Patterson, D.: Roofline: an insightful visual performance model for multicore architectures, *Comm. ACM*, 52:65-76 (2009).
- [4] 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, pp.23-31 (2012).

- [5] Datta, K., Murphy, M., Volkov, V., Williams, S., Carter, J., Olike, L., Patterson, D., Shalf, J. and Yelick, K.: *Stencil computation optimization and autotuning on state-of-the-art multicore architectures*, Conference, Austin, TX, Nov. 15-21, pp.1-12, IEEE Press, Piscataway, NJ (2008).
- [6] Williams, S., Carter, J., Olike, L., Shalf, J. and Yelick, K.: Lattice Boltzmann simulation optimization on leading multicore platforms, *Proc. IEEE International Symposium on Parallel and Distributed Processing Symposium*, Miami, FL, Apr. 14-18, pp.1-14 (2008).
- [7] Frigo, M., Leiserson, C.E., Prokop, H. and Ramachandran, S.: Cache-oblivious algorithms, *Proc. 40th Annual Symposium on Foundations of Computer Science*, New York, pp.285-297 (Oct. 1999).
- [8] Günther, F., Mehl, M., Pögl, M. and Zenger, C.: A Cache-Aware Algorithm for PDEs on Hierarchical Data Structures Based on Space-Filling Curves, *SIAM J. Sci. Comput.*, Vol.28, No.5, pp.1634-1650 (2006).
- [9] Rossinelli, D., Hejazialhosseini, B., Hadjidoukas, P., Bekas, C., Curioni, A., Bertsch, A., Futral, S., Schmidt, S.J., Adams, N.A. and Koumoutsakos, P.: 11 PFLOP/s Simulations of Cloud Cavitation Collapse, *SC13 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, November 17-21, Denver, CO, USA (2013).
- [10] 佐藤義永, 永岡龍一, 撫佐昭裕, 江川隆輔, 滝沢寛之, 岡部公起, 小林広明: ルーフラインモデルに基づくベクトルプロセッサ向けプログラム最適化戦略, 情報処理学会論文誌 コンピューティングシステム, Vol.4, No.3, pp.77-87 (2011).
- [11] available from (<https://www.nersc.gov/users/computational-systems/cori/nersc-8-procurement/trinity-nersc-8-rfp/nersc-8-trinity-benchmarks/stream/>).
- [12] 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).
- [13] 乱流音場解析ソフトウェア FrontFlow/Blue: 入手先 (<http://www.ciss.iis.u-tokyo.ac.jp/riss/project/device/>).
- [14] Ilic, A. Pratas, F. and Sousa, L.: Cache-aware Roofline model: Upgrading the loft, *IEEE Computer Architecture Letters*, Vol.13, No.1 (Jan.-June 2014).
- [15] FUJITSU 2012-5月号 (Vol.63, No.3) 特集: スーパーコンピュータ「京」(2012).
- [16] 片桐孝洋: 富士通 PRIMEHPC FX10 チューニング連載講座, 6. 数値計算ライブラリの利用, 入手先 ([http://www.cc.u-tokyo.ac.jp/support/press/news/VOL15/No6/08\\_201311-tuning-fx10-library.pdf](http://www.cc.u-tokyo.ac.jp/support/press/news/VOL15/No6/08_201311-tuning-fx10-library.pdf)).
- [17] Intel Xeon Processor E5 and E7 v3 Family Uncore Performance Monitoring Reference Manual.
- [18] 南 一生: 配信講義, CMSI 計算科学技術特論 B, アプリケーションの性能最適化 2 (CPU 単体性能最適化), 入手先 (<http://www.cms-initiative.jp/ja/events/2014-haishin>).
- [19] 青木正樹: プログラムのチューニング方法, 入手先 (<http://www2.itc.nagoya-u.ac.jp/riyou/tuning.pdf>).
- [20] Kumahata, K., Inoue, S. and Minami, K.: Kernel performance improvement for the FEM-based fluid analysis code on the K computer, *2013 International Con-*

*ference on Computational Science, Procedia Computer Science*, Vol.18, pp.2496-2499 (2013), available from ([www.sciencedirect.com](http://www.sciencedirect.com)).



南 一生 (正会員)

1981年日本大学理工学部物理学卒業, 同年富士通株式会社入社。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。



井上 俊介

1999年横浜国立大学教育学部卒業。同年(株)富士通長野システムエンジニアリング(現,(株)富士通システムズ・イースト)入社。2010年(独)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し, スーパーコンピュータ「京」のアプリケーション最適化に従事。2014年出向解除。現在(株)富士通システムズ・イーストにて, HPCアプリケーション最適化に従事。



千葉 修一 (正会員)

1973年生。1992年株式会社富士通神戸エンジニアリング入社。2003年富士通株式会社入社。スーパーコンピュータ「京」のソフトウェア開発のチームリーダーとしてプログラミング言語の研究・開発に従事。現在, 次期エクサスケールコンピュータに向けたソフトウェアの研究・開発を推進。



熊畑 清

2008年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了。同大産学官連携研究員、(株)富士通長野システムエンジニアリング(現、(株)富士通システムズ・イースト)を経て、2012年より理化学研究所計算科学研究機構にて「京」コンピュータの運用・高度化ならびにソフトウェアの性能改善に従事。博士(情報科学)。



横川 三津夫 (正会員)

1984年筑波大学大学院修士課程理工学研究科修了。同年日本原子力研究所入所。1997年地球シミュレータ研究開発センターにて地球シミュレータ開発に従事。2002年産業技術総合研究所グリッド研究センター。2006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て、スーパーコンピュータ「京」の開発に従事。2012年神戸大学大学院システム情報学研究科教授。2002年、2011年ゴードン・ベル賞受賞。工学博士。