RA-003

キャッシュの効果を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

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

南一生[†1] 井上俊介[†2] 千葉修一[†3] 横川三津夫[†4]
Kazuo Minami Shunsuke Inoue Syuichi Chiba Mitsuo Yokokawa

1．まえがき

以前の研究者やプログラマは，高級言語とコンパイラにより，数理モデルの定式化・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なうことが一般的であった．しかし，現在では，計算機アーキテクチャの変化によりプログラミングに変化が生じている．ここでいう計算機アーキテクチャの変化とは，ひとつは，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．もうひとつは，メモリーウォール問題に対処するためにキャッシュ等を用いた，多層メモリ構造の採用である．ここに示した超並列アーキテクチャや多層メモリ構造に対応するための「超並列性を引き出すプログラミング」と「プロセッサの単体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 万個あまりに及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，ユーザ，研究者，プログラマが意識すべきプログラミング上の重要な点である．

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が，限られた計算資源の有効活用，また研究期間の短縮に資するものである．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しうる実効性能値，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，メモリバンド幅，CPU 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]．著者らも，「京」においてルーフラインモデルをベースに，プログラムのコーディングを精査することにより性能予測を実施し，その予測を元に CPU 単体性能チューニングを実施している[4]．しかし，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良く一致するが，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，及びチューニングを通して明らかになった事例をベースに，キャッシュアクセスのあるプログラムの性能限界値を予測するモデルを提案する．この予測モデルは，メモリバンド幅により性能律速となっている状態からキャッシュアクセスが増大した状態まで適用可能である．

2．「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[5]．一つの計算ノードは，一つの CPU（富士通製 SPARC64™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/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 22.7mm×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．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，したがってコアの理論性能は 16GFLOPS，CPU（8 コア）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．メモリの理論バンド幅は 64GB/秒，B/F 値は 0.5，L2 キャッシュの理論バンド幅は 256GB/秒，B/F 値は 2.0，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．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（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/秒であった．

3. 「京」でのルーフラインモデルの検証

3.1 ルーフラインモデル

ルーフラインモデルは，Williams の論文[3]中のステンシルの例[6]や疎行列・ベクトル積の例[7]でも議論されているようにキャッシュからのロードは考慮しないメモリアクセスに則った性能予測となっている．ハードウェアが持つ実効的なメモリバンド幅を B，ハードウェアの持つピーク性能を F，アプリケーションの演算強度を X=f/b（アプリケーションの要求フロップス値：f，アプリケーションの要求バイト値：b）とすると，アプリケーションの性能は min（F，B*X）で表される[3]．ここで BX を，

$$B*X=B*(f/b)=B*F*f/(b*F)=(B/F)/(b/f)*F$$

†1 独立行政法人 理化学研究所 計算科学研究機構 運用技術部門
†2 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
†3 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
†4 国立大学法人神戸大学 大学院システム情報学研究科

と変形すると，B*X は，ハードウェアの持つ B/F 値をアプリケーションの要求 b/f 値で除した値にピーク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を求めることができる．本論文では， B，F，b，f 用いてアプリケーションの性能を，

min(F, (B/F)/(b/f)*F)

アプリケーションのピーク性能比を

min(1.0 , (B/F)/(b/f))

で考える．

3.2 ルーフラインモデルの適用可能ケースと適用限界

地震波伝播アプリケーション Seism3D[8][9]と汎用流体解析コード FrontFlow/Blue[10] の「京」上のルーフラインモデルによる性能予測とチューニングの結果については，文献 [4]に報告されている．これらの事例では，メモリバンド幅に性能が支配されるような場合には，従来のルーフラインモデルで性能限界を良く予測することができる．

ここで， 図-1 に示す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．この例では，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，ストアするために 1 度ロードもするため配列アクセスは 2 と勘定すると計 5 個である．また cdiv の配列が L2 キャッシュに常時キープされるため，メモリアクセス対 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．このループにルーフラインモデルを適用すると，演算数 43，倍精度浮動小数点数（8 バイト）に対し，要求 b/f 値は 40/43＝0.93，ハードウェアの実効 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%となり，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るルーフラインモデルでは難しいことが分かった．

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

図-1 ルーフラインモデルが当てはまらないプログラムの例

3.3 ルーフラインモデルの検証

ルーフラインモデルは，メモリ性能に基づく性能予測手法であるが，メモリ性能律速の場合のみならず，全ての配列が L2 キャッシュにキープされた状態についても，ルーフラインモデルが適用可能であることを示す．

まず，全ての配列がメモリに載った状態（on メモリ），L2 キャッシュに載った状態（onL2）の 2 種類のテストプログラムを用意した，それぞれ配列サイズを変化させることにより，両方の状態を実現できる．メモリ及び L2 キャッシュアクセス性能テストのプログラムを図-2(a)に示す．

```
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-2 メモリ・L2 キャッシュ基礎性能テストプログラム

このプログラムを基本として，要求バイト数を変えないように，図-2(a)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．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，不要な性能低下要因を導入しないためである．j のループに対してはブロック分割による 8 スレッド並列化を実施している．またプログラムでは，変数が on メモリ，または onL2 となるように，ループの回転数 M，N の大きさを調整した．

メモリ性能テストの結果を表-1 に示す．メモリバンド幅の値の平均値より，メモリの実効バンド幅を 46.3GB/sec と測定した．この実行メモリバンド幅とルーフラインモデルを用いた性能予測値を表-1 の予測性能の欄に，予測性能と実測性能の比較グラフを図-3 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良く一致していることが確認された．実効 B/F 値は，実効メモリバンド幅/理論メモリバンド幅×理論メモリ B/F 値で求められ，B/F 値 0.5 の場合は，実効 B/F 値=46.3/64*0.5 =0.36 となる．

表-1 メモリ基礎性能テスト結果

b/f値	予測性能 (peak比)	実行時間 (sec)	実測性能 (peak比)	メモリバンド幅(GB/sec)
0.5	0.72	3.34	0.7186	45.92
1	0.36	3.29	0.3647	46.61
2	0.18	3.32	0.1807	46.22
3	0.12	3.28	0.1220	46.81
4	0.09	3.3	0.0909	46.55
6	0.06	3.34	0.0599	46.02
12	0.03	3.34	0.0299	45.98

次に，L2 キャッシュ性能テストの結果を表-2 に示す．表-2 の L2 キャッシュのバンド幅の平均値より，L2 の実効バンド幅は 159.2GB/sec とした．

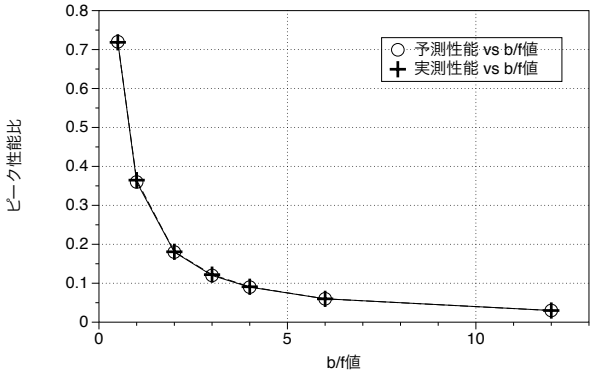


図-3 メモリ基礎性能テスト結果

この値は，後述する理由により B/F 値 0.5 及び 1 の場合に計測したバンド幅値を除いた平均値である．この L2 実効バンド幅とルーフラインモデルを用いた性能予測値を表-2 の予測性能の欄に，予測性能と実測性能のグラフを図-4 に示す．L2 の実効 B/F 値もメモリと同様に，理論バンド幅：256GB/sec と L2 の B/F 値：2.0 を使い，L2 の実効 B/F 値=159.2/256 *2.0 =1.24 と計算される．したがって L2 においては，B/F 値=1.24 以上では予測値と合致している．また B/F 値=1.24 以下では性能が 100%以上に予測されるので 1 が上限となる．ただし，通常，演算性能がピーク性能の 100%に達することはない．この B/F 値=1.24 以下の部分は，もっと演算器が装備されていれば，L2 のバンド幅を使い切ることができるが，実際には演算器の限界のために L2 のバンド幅が使い切れない部分である．そのため L2 バンド幅の測定値が低下している．この理由により， L2 バンド幅の平均値の計算から除外した．この結果から，L2 キャッシュにおいてもピーク性能付近でなければルーフラインモデルが適用可能であることが明らかとなった．

表-2 L2 キャッシュ基礎性能テスト結果

b/f値	予測性能 (peak比)	実行時間 (sec)	実測性能 (peak比)	L2バンド幅 (GB/sec)
0.5	1.00	10.97	0.8806	56.36
1	1.00	5.91	0.8173	104.61
2	0.63	3.91	0.6176	158.12
3	0.42	3.87	0.4160	159.75
4	0.31	3.82	0.3161	161.84
6	0.21	4.04	0.1993	153.03
12	0.10	3.78	0.1065	163.56

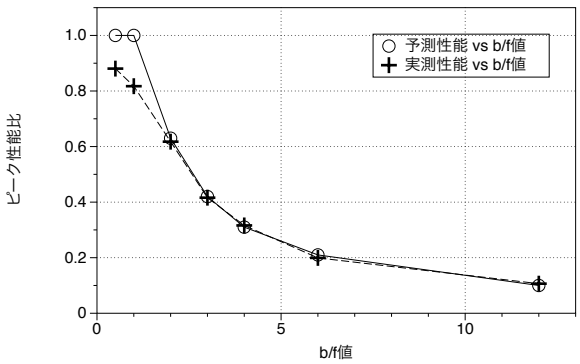


図-4 L2 キャッシュ基礎性能テストの結果

4. 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

4.1 限界性能予測モデル

ここではメモリアクセス律速状態から L2 キャッシュアクセスを増大させ，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について，ルーフラインモデルを拡張した限界性能の予測モデルを提案する．

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した値であり，演算量はプログラムの中の演算の数を勘定して求める．したがって，実行時間の予測ができれば性能予測が可能となるが，メモリアクセス律速状態から，L2 キャッシュアクセス律速状態，さらに演算律速と変化する場合の実行時間は，次の 3 つのケースとなると仮定した．L1 キャッシュに関する議論については後述する．

ケース１：メモリバンド幅律速，かつ演算律速なしの場合の 実行時間はメモリ上のデータの移動時間である．

ケース２：L2 バンド幅律速，かつ演算律速なしの場合の実行時間は L2 上のデータの移動時間である．

ケース３：メモリバンド幅律速，または L2 キャッシュバンド幅律速の状態から演算量が増えてくると演算律速の状態となる．その場合の実行時間は演算時間である．

演算律速の場合の演算時間は，「京」の場合，SIMD 演算が適用できるか，積和演算が適用できるか（積と和のバランス），コンパイラのスケジューリングができているか等によって予測値が異なるため，本論文では演算器が完全に理想的に動いていると仮定し，ピーク性能が得られると仮定した場合の実行時間を用いるものとした．演算律速の場合の精度の高い性能予測は，本論文では考慮しない．

ここで，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 データ転送量，実効 L2 バンド幅，プログラム演算量，演算ピーク性能を，それぞれ Mdata，Mband_peak，L2data，L2band_peak，Ca，Ppeak とする．このとき，メモリに載っているデータの移動にかかる時間：Mtime，L2 に載っているデータの移動にかかる時間：L2time，演算時間：Ctime は以下のように計算できる．

$$Mtime=Mdata/Mband\_peak$$
$$L2time=L2data/L2band\_peak$$
$$Ctime=Ca/Ppeak$$

ケース 1，ケース 2，ケース 3 のプログラムの実行時間は，それぞれ，Mtime，L2time，Ctime となるため，プログラムの実行時間：Ex_time，演算ピーク性能比 Cp は，以下で計算される．

$$Ex\_time=max(Mtime,L2time,Ctime)$$
$$Cp= Ca/(max(Mtime,L2time,Ctime)*Ppeak)$$

このモデルでは，メモリデータ転送律速の場合は，Mtime>L2time であるが，L2 の転送量が増えてくるとある時点で Mtime<L2time になると考えられるので，実行時間が Mtime から L2time へ変化する交差点が存在するはずである．この交差点は，Mtime=L2time なので，ループの回転数を Nloop とすると交差点の条件は以下のように求められる．この時，性能評価対象のプログラムのメモリアクセス，L2 アクセス，演算数の比をｍ：ｎ：ｋとする．

$$Nloop*m/Mband\_peak=Nloop*(m+n)/L2band\_peak$$
$$n=((L2band\_peak/Mband\_peak)-1)*m$$

メモリアクセスするデータは L2 にもアクセフするため L2data の計算において（m+n）の項が入っている．また 3.1 節に述べたように，アプリケーションの性能は，ハードウェアの持つ B/F 値をアプリケーションの要求 b/f 値で除した値にピーク性能を掛けた値となるため，配列要素が倍精度の場合に，Mtime>L2time の領域と，Mtime<L2time の領域では，それぞれ以下のように求められる．

(1) Mtime>L2time の領域：

$$Cp=(Mband\_peak/Ppeak)/(m*8/k)$$

(2) Mtime<L2time の領域：

$$Cp=(L2band\_peak/Ppeak)/((m+n)*8/k)$$

3.1 節に示した従来のルーフラインモデルの式を本節で使用している変数で記述すると以下のようになる．

$$Cp*Ppeak=min(Ppeak, Mband*(f/b))$$

この式を $f/b=Ca/Mdata$ を使用して変形すると下式が導出される．

$$Cp= Ca/(max(Mtime,Ctime)*Ppeak)$$
$$Ex\_time=max(Mtime,Ctime)$$

したがって従来のルーフラインモデルは，本節の提案モデルにおいて L2time の効果を考慮していないモデルであると言える．L2time を考慮し Mtime と L2time の切り替えを考慮するところが提案するルーフラインモデルの拡張部分である．

4.2 性能限界値予測モデルの検証

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，メモリとキャッシュに載る配列の量を変化させ性能を測定する．メモリ・L2 キャッシュアクセス混合性能テストのプログラムを図-5 に示す．図-5 のプログラムはメモリのみ考慮した時の要求バイト：B=3×8=24，要求演算数：F=2 であり要求 b/f 値：B/F=24/2=12 となる．本プログラムは，k のループに対しブロック分割による 8 スレッド並列化を実施している．N1=4000，N2=60，N3=80 を設定し全体を 60 回実行して計測し，配列の２つ目の添え字:j の差分一つ分は 4000×8 バイト=32KB は離れているので配列:c への差分アクセスは L2 アクセスとなる．図-5 の L2 に対する要求バイトは，2×8=16 となる．また演算数は 2 である．このプログラムは 3 つのメモリアクセス，2 つの L ２アクセス，演算が 2 であるので，3M-2L2-2F と呼ぶこととする．このプログラムをベースとして，（右辺)* c(i,j+2,k)+c(i,j-2,k)のように項を追加し L2 のアクセスと演算数を増やしている．一般に，性能評価対象のプログラムのメモリアクセス，L2 アクセス，演算数の比をｍ：ｎ：ｋとした場合のプログラムを mM-nL2-kF と呼ぶことにする．

```
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 キャッシュアクセス混合性能テストプログラム

これらのプログラムを用いて測定した実行時間を表-3 に示す．メモリのアクセス量を固定し，L2 のアクセス量を増加させている．同一の L2 アクセス量について演算量を複数パターン変化させているケースも用意した．図-6 は，横軸にメモリアクセス律速状態から，L2 キャッシュアクセス律速状態，さらに演算律速と変化する順番にプログラムを並べ，縦軸に実行時間を示したものである． Mtime，L2time，Ctime の計算のための各パラメタは，以下の通りである．

$$Mdata[Byte]＝m*ループの回転数$$
$$L2data[Byte]=(m+n)*ループの回転数$$
$$Mband[Byte/sec]=Mdata/各ケースの実行時間$$
$$L2band[Byte/sec]= L2data/各ケース実行時間$$
$$Ca=k*ループの回転数$$

表-3 メモリ・L2 混合性能テスト結果

ケースNo		実行時間 (sec)	実測性能 (peak比)	Cp(peak 比)
1	3M-2L2-2F	0.63	0.029	0.030
2	3M-3L2-4F	0.63	0.057	0.060
3	3M-4L2-4F	0.60	0.060	0.060
4	3M-5L2-6F	0.64	0.084	0.090
5	3M-6L2-6F	0.66	0.082	0.090
6	3M-6L-12F	0.65	0.166	0.180
7	3M-6L2-24F	0.67	0.322	0.359
8	3M-6L2-48F	0.67	0.645	0.719
9	3M-8L2-8F	0.74	0.097	0.104
10	3M-8L2-16F	0.73	0.197	0.207
11	3M-8L2-32F	0.74	0.389	0.415
12	3M-8L2-64F	0.76	0.758	0.830
13	3M-8L2-128F	1.34	0.860	1.000
14	3M-10L2-10F	0.85	0.106	0.110
15	3M-10L2-20F	0.85	0.212	0.219
16	3M-12L2-12F	0.97	0.111	0.114
17	3M-12L2-24F	0.95	0.227	0.228
18	3M-14L2-28F	1.07	0.236	0.235

計測の結果，Mband の最高値は 46GB/sec であり，この値を Mband_peak とする．また L2band の最高値は 146GB/sec であり，この値を L2band_peak とする．L2band_peak は，3.4 節 の L2 基礎テストの章で得られた 159GB/sec より低めに出ているが，「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 基礎テストの値より少し小さな値となっている．図-6 では，それぞれ，Mtime=Mdata/Mband_peak，L2time=L2data/L2band_peak，Ctime=Ca/Ppeak で計算した．単位はそれぞれ sec である．

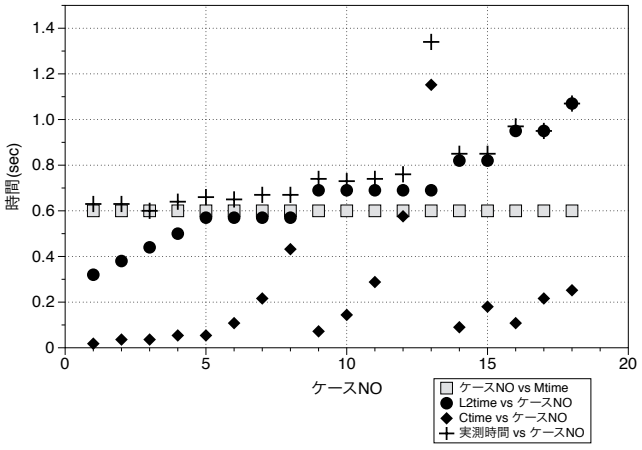
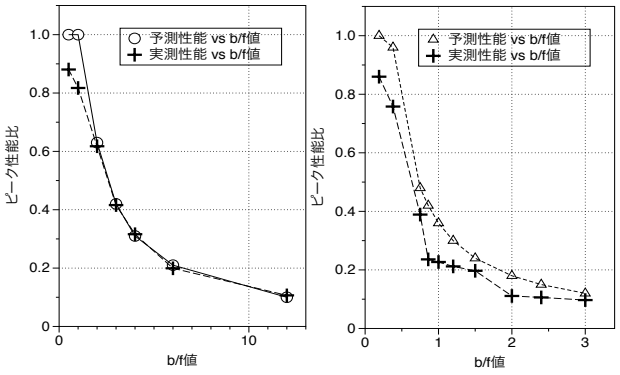


図-6 メモリ・L2 混合性能テスト結果

図-6 を見ると Mtime>L2time の領域では，実測時間はほぼ Mtime に一致していることがわかる．また，Mtime<L2time の領域では，実測時間は，ケース 13 を除きほぼ L2time に一致していることがわかる．ケース 13 は Ctime>L2time となる領域であり，実測時間は Ctime に近い値となっている．他のケースと比べると差異が大きいように見えるが，この理由は Ctime の予測のために，プログラムがピーク性能比 100%の性能が出ることを仮定しているためである．通常，ピーク性能に近い性能の達成は，DEGEMM 等の特別にチューニングされたプログラムに限られる．したがって Ctime>L2time の領域で，高い性能が予測される場合の Ctime の予測は誤差が大きくなる．

Mtime>L2time から Mtime<L2time に切り替わる交差点の条件は，n=((L2band/Mband)-1)*m であるので，L2band を L2band_peak，Mband を Mband_peak の値を使用すると，交差点の条件は，n=2.17m である．この条件をテストプログラムの名称で表わすと ，3M-6L2 と 3M-8L2 の間となり．ケース番号 8 と 9 の間と予測されるが，図-6 の実測値とも一致している．図-6 を見ると 4 章の性能モデルが実測値と一致していることがわかる．



(1)Mtime>L2time の領域　　(2) Mtime<L2time の領域

図-7 メモリ・L2 混合性能テスト結果(従来と提案モデルの比較)

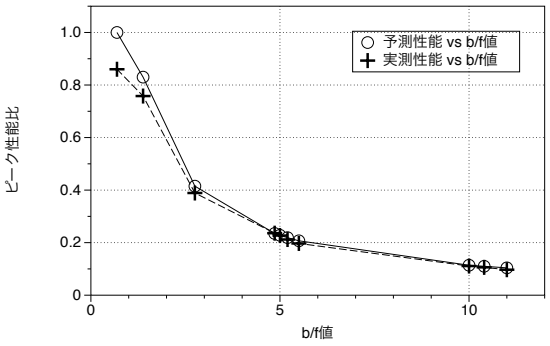


図-8 メモリ・L2 混合性能テスト結果(Mtime<L2time の領域)

ここで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(1)は，Mtime>L2time の領域であり従来のルーフラインモデルと提案した予測モデルが同じ予測値を与える領域であるり，予測値と実測値がよく一致していることを示している．Mtime<L2time の領域は提案した予測モデルを適用すべき領域であるが，この領域に従来のルーフラインモデルを適用した結果を図-7(2)に示す．図-7(2)を見ると予測値と実測値の乖離があることがわかる．図-7(1)(2)共に横軸の b/f 値は，メモリアクセス量と演算量の比から 8*m/k で計算している．図-8 は，Mtime<L2time の領域に提案した予測モデルを適用した結果を示す．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-8 の横軸の b/f 値は，L2 アクセス量と演算量の比から 8*(m+n)/k で計算している．図 7，8 共に Ca は k*ループの回転数で計算し，Cp は，4.1 節に示した式で計算した．ただしピーク性能比 1.0 を超える事はないので Cp が 1.0 以上の値は 1.0 を上限としている．図 7，8 共に，予測性能値が 80%を超える領域については，演算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. プログラムの性能予測方法

5.1 L1 キャッシュの影響の評価

4 節で提案したモデルを用いてプログラムの性能予測が可能となると考えられるが， L1 キャッシュの影響に気をつけなければならない．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．

- SIMD と noSIMD アクセスによりデータ転送速度が大きく異なる．
- キャッシュミスのペナルティ，特にストア時のペナルティが大きい．
- L2 から L1 へのアクセスはキャッシュライン単位であるが L1 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．

そこで，L2 アクセスとの差異があるかどうかを確認するために，以下の 4 種類のテストプログラムにより，データアクセス性能の評価を実施した．

a) 図-2 ベースの on L1 キャッシュで配列の数を一定にしたプログラム(図-9)

b) a) のプログラムで on L1 キャッシュの配列の数を増やしたプログラム(図-10)

c) 図-5 ベースの on L1 キャッシュで配列の数を増やしたプログラム(図-11)

d) c)と同様であるが差分間隔が c)とは異なるプログラム(図-12)

a)のプログラムは，図-2 と比較すると j の添字を取り除いているが，最内ループ長を確保しコンパイラの最適化の状態を L2 キャッシュのテストと同じ状態にして評価を実施するためである．c)のプログラムは L1 のアクセスを，c(i+1,j,k)，c(i+2,j,k)のように 1 つ単位で増減させており，差分間隔が short であると定義する．d)のプログラムは L1 のアクセスを，c(i+12,j,k)，c(i+24,j,k)のように 12 単位で増減させており，差分間隔が long であると定義する．

```
      do j = 1,N
        do i = 1,M
         c10(i) = z+c1(i)*x
        enddo
      enddo
```

図-9 L1 キャッシュ性能テストプログラム a)

```
     do j = 1,N
       do i = 1,k
         c10(i) = (((z+c1(i)*x)* &
                              c1(  k+i)+z)*  &
                              c1(2*k+i)+z)*  &
                              c1(3*k+i)+z
       enddo
     enddo
```

図-10 L1 キャッシュ性能テストプログラム 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)
        enddo
      enddo
    enddo
```

図-11 L1 キャッシュ性能テストプログラム c)

```
    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 d)

測定結果は，a)は L1 キャッシュのバンド幅が 300GB/sec と測定された．また b)のプログラムは， 350GB/sec，d)のプログラムは，430GB/sec と測定された．また c)のプログラムは，コンパイラの最適化のために正確な L1 バンド幅は測定できなかったが，メモリのアクセス量に対し 10 倍程度 まで L1 アクセスする配列を増やしても，メモリアクセス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った．

L2 キャッシュに関する 3.4 節と 5.2 節のバンド幅の結果を見ると，3.4 節のバント幅が 5.2 節のバンド幅より大きい．L1 キャッシュのテストでは，3.4 節に対応するプログラムが a)であり 5.2 節に対応するプログラムが d)であるが，a)より d)のバンド幅の方が大きくなっており，L2 キャッシュとは逆の傾向である．また本来であれば，a)と b)のバンド幅は，同じ値となるはずである．

以上の結果より L1 キャッシュは，その機構の複雑さから，メモリや L2 キャッシュと同等のバンド幅によるルーフラインモデルの適用はできないものと考える．しかし，ここでの議論により，L1 キャッシュについて以下の条件が提示できるものと考える．

(1) c)の short タイプの差分については，メモリのアクセス量に対し 10 倍程度 まで L1 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) a)b)d)タイプの L1 アクセスにおいて SIMD アクセスでは，300GB/sec 程度のバンド幅は確保できることがわかった．L2 のバンド幅が 145GB/sec ほどであるため，L2 アクセスと同程度の L1 アクセスがあっても，L2 ベースのルーフラインモデルの適用が可能である．

5.2 L1 アクセスを考慮した性能予測モデル

L1 のアクセス量が限界まで達していない性能上限値の予測方法は，以下のとおりである．まず，プログラムのメモリ，L2 キャッシュ，L 1 キャッシュアクセスバイトを計算する．計算する項目は以下の通りとする．

a) メモリまでアクセスする配列のバイト数(m)

b) L2 までアクセスする配列のバイト数(nL2)

c) L1 までアクセスするバイト数のうち short アクセスするバイト数(nL1_S)

d) L1 までアクセスするバイト数のうち long アクセスするバイト数(nL1_L)

次に L1 アクセスの影響について，SIMD ロードアクセスを前提として，L 1 のアクセス量が限界まで達していない事を確認する．限界を超えていない場合は，メモリアクセス性能を基礎とした予測，または L2 アクセス性能を基礎とした予測ができるので， nL1_S および nL1_L の影響は考慮しなくても良い．

さらに SIMD ロードアクセスで nL1_S<10m かつ nL1_L<nL2+m である事を確認する．成立する場合は L1 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．これらが満たされた条件で以下のように性能予測を行う(以下倍精度データの例)．

(1) nL2≦((L2band/Mband)-1)*m の場合
・メモリアクセス律速での予測とする
Cp=(Mband_peak/Ppeak) /(m*8/k)

(2) nL2>((L2band/Mband)-1)*m の場合
・L2 アクセス律速での予測とする
Cp=(L2band_peak/Ppeak) /((m+nL2)*8/k)

6. 実アプリケーション性能予測と評価

6.1 性能予測とチューニング

本章では，性能予測とチューニングの具体例を示す．最初の性能実測で予測性能まで性能が達していない場合は，なんらかの性能問題が発生していると考えられる．このような場合は，問題を取り除くためのチューニングの必要性が明らかとなる．本章の例では，性能予測後の実測性能と予測性能の差の評価，問題の究明，チューニングの実施例について述べ，提案した性能モデルによる性能予測が実際に有効であることを示す．実測性能と予測性能がどれくらい一致すれば良いかについては，差異 15%を目安とした．

6.2 性能予測に基づいた性能改善

ここでは，3.2 節図-1 に示されたプログラム(A)及び別途用意した 3 つのカーネルプログラム(B)(C)(D)の 4 個のプログラムに対し，性能予測モデルによる性能推定値と実測性能との比較から，プログラムに対しチューニングを適用し，予測値まで性能を上げた例について述べる．ここで 5.2 節の予測式に 4.2 節で測定された値を使用することで下式を使用して予測する．

(1) nL2≦2.17m の場合
Cp=(46/128)/(m*8/k)=0.36/(m*8/k)

(2) nL2 >2.17m の場合
Cp=(146/128)/((m+nL2)*8/k)=1.14/((m+nL2)*8/k)

まずプログラム(A) の性能評価を行った．このプログラムは， m=5，nL2=21，nL1_S=6，nL1_L=12，k=43 である．nL1_S<10m および nL1_L<nL2+m の条件を満たすので，また SIMD アクセスの条件を満たすので L1 のアクセスは考慮しない． nL2>2.17m が成り立つので L2 アクセス律速状態での予測となる．L2 アクセス律速状態での予測値は，1.14/((5+21)*8/43)=1.14/4.83=0.236 (23.6%) となる．従来のルーフラインモデルでの予測性能は，38.7%となる．

このプログラムの実測値は，14.7%であり予測値より大

分低い値となっていたため，性能劣化の調査を行ったところ， L1 ミス dm 率が 13.36%であった．L1 ミスは，データアクセス要求(dm)時に発生する，ハードウェアプリフェッチ時に発生する，ソフトウェアプリフッチ時に発生する，の３種類あるが，要求(dm)時に発生する L1 ミスが少ない方が良い．L1 ミス dm 率は，L1 ミスのうち，要求(dm)時に発生する L1 ミスの割合である．この値が大きめの場合は，キャッシュスラッシング[11][12]の可能性が疑われたため，配列のマージ[11][12]を実施した．その結果，実測性能は 19.7%まで向上し，予測性能まで近づくことができた．また L1 ミス dm 率は 3.85%まで低下した．

プログラム(B)のパラメタは，m=13，nL2=2，nL1_S=12，nL1_L=8，k=60 である．nL1_S<10m および nL1_L<nL2+m の条件を満たすので，また SIMD アクセスの条件を満たすので L1 のアクセスは考慮しない． nL2>2.17m が成り立たないのでメモリアクセス律速状態での予測となる．したがって従来のルーフラインモデルの予測値と同様となる．メモリアクセス律速状態での予測値は，0.36/(13*8/60)= 0.36/1.73=0.208（20.8%）となる．最初の実測値は，12.9%であり予測性能値に達していなかった． L1 ミス dm 率を確認すると 15.11%であった．ここでも配列マージのチューニングを実施することにより，実測性能が 19.3%まで向上し，L1 ミス dm 率も 8.81%まで低下し，予測性能近くまで性能向上できた．

プログラム(C)のパラメタは，m=11，nL2=2，nL1_S=0，nL1_L=2，k=11 である．nL1_S<10m および nL1_L<nL2+m の条件を満たすので，L1 のアクセスは考慮しない．nL2>2.17m が成り立たないのでメモリアクセス律速状態となる．したがって従来のルーフラインモデルの予測値も同様となる．メモリアクセス律速状態での予測値は，0.36/(11*8/11)=0.36/8=0.045（4.5%）となる．実測値は，3.81%である．この例は，L1 ミス dm 率も 0.31%と低く，予測値と実測値の差が 15%程度になっているため，これ以上の性能向上はできないと判断した．

プログラム(D)のパラメタは，m=3，nL2=8，nL1_S=8，nL1_L=0，k=25 である．nL1_S<10m および nL1_L<nL2+m の条件を満たすので，L1 のアクセスは考慮しない．nL2>2.17m が成り立つので L2 アクセス律速状態となる．L2 アクセス律速状態での予測値は，1.14/((3+8)*8/25)= 1.14/4.83=0.324（32.4%）となる．従来のルーフラインモデルでの予測性能は，38.7%となる．実測値は 29.0%であり，予測値と実測値の差が 15%以内になったため，これ以上のチューニングは行わないこととした．ここに述べた実例のサマリを表-4 に記載する．

表-4 4 つのプログラムの性能予測結果

	m	nL2	nL1_S	nL1_L	演算数	予測性能従来法（peak比）	予測性能提案法（peak比）	実測性能（peak比）	tune後実測性能（peak比）
(A)	5	21	12	6	43	0.387	0.236	0.147	0.197
(B)	13	2	12	8	60	0.208	0.208	0.129	0.193
(C)	11	2	0	2	11	0.045	0.045	0.038	—
(D)	3	8	8	0	25	0.375	0.324	0.290	—

7. まとめ

プログラムの限界性能を予測するためにルーフラインモデルが提唱されている．本報告では，「京」上で，ルーフラインモデルが，onL2 キャッシュの場合にも成り立つ事を確認した．また，ルーフ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．

今後は，このような性能予測モデルを他のアーキテクチャに適用してことが必要と考えている．

謝辞

本報告に際しご討論頂き貴重な助言を頂いた，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す．

[1] Matteo Frigo,Volker Strumpen：Cache Oblivious Stencil Computation,ICS '05 Proceedings of the 19th annual international conference on Supercomputing,Pages 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 2001(22), 91-96, 2001-03-08
[3] S. Williams, A. Waterman, and D. Patterson: Roofline: an insightful visual performance model for multicore architectures. Commun. 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] 雑誌 FUJITSU2012-5 月号 (VOL.63，NO.3)特集：スーパーコンピュータ「京」
[6] Datta, K., Murphy, M., Volkov, V., Williams, S., Carter, J., Oliker, 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). IEEE Press, Piscataway, NJ, 2008, 1-12.
[7] Williams, S., Carter, J., Oliker, L., Shalf, J., and Yelick, K. Lattice Boltzmann simulation optimization on leading multicore platforms. In Proceedings of the IEEE International Symposium on Parallel and Distributed Processing Symposium (Miami, FL, Apr. 14–18, 2008), 1–14.
[8] 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.
[9]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.
[10]「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/
[11]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），http://www.cms-initiative.jp/ja/events/2014-haishin
[12]青木正樹,プログラムのチューニング方法, http://www2.itc.nagoya-u.ac.jp/riyou/tuning.pdf