Blue Gene/Pにおけるアプリケーション最適化と性能評価

土井 淳

Application Tuning and Performance Evaluation on the Blue Gene/P

Jun Doi

Blue Gene®/PはBlue Gene®/Lの後継として発表されたスーパーコンピューターであり, その最大計算性能はペタフロップスを超える. この性能を活かすためにアプリケーションの最適化は必要不可欠である. 本論文は, アプリケーション最適化の事例としてBlue Gene/Lで前例のある, 格子QCDシミュレーションの最適化を行った. ピーク性能に対して30%を超える実効性能を得ることができ, Blue Gene/Pの優位性を示すことができた.

IBM has announced the Blue Gene®/P supercomputer that is to succeed Blue Gene®/L, and its performance exceeds one petaFLOPS. It is important to tune the application to utilize its performance as much as possible. In this paper, we describe an example of application tuning; the tuning lattice QCD simulation that we have performed on the Blue Gene/L. We finally obtained over 30% sustained performance against the peak performance and were able to demonstrate the advantage of the Blue Gene/P.

Key Words & Phrases：スーパーコンピューター, Blue Gene, アプリケーション最適化, 性能評価, 格子
supercomputer, Blue Gene, application tuning, performance evaluation, lattice QCD

1. はじめに

IBM "System Blue Gene"/P Solution（以降Blue Gene®/P）[1]は, Blue Gene®/L [2] [3]の後継機種として発表されたスーパーコンピューターである. 基本的な設計概念や設計思想はBlue Gene/Lのものを受け継ぎ, 低消費電力, 省スペース設置ながら, 最大構成時のピーク計算性能は3PFLOPS（1秒間に3千兆回の浮動小数点数演算）まで引き上げられた.

しかしながら, Blue Gene/Pの性能を引き出すためには実行するアプリケーションをBlue Gene/Pに特化したような最適化を行うことが重要である. 十分な最適化が行われない状態で実行すると, 性能を活かしきれないばかりか, せっかくの低消費電力性を無駄にしてしまうことにもなりかねない.

我々は, Blue Gene/Lにおいて, 格子量子色力学（格子QCD）シミュレーションの最適化で良い成果[4]を残してきており, Blue Gene/Pにおいても良い成果が期待される. 格子QCDとは, 強い力の相互作用をコンピューターでシミュレーションする理論で, 非常に多くの計算量が必要なアプリケーションであり, Blue Geneのような超並列計算機において高い性能を出すことは非常に重要である.

本論文では, Blue Gene/Pの基本性能を知った上で, 格子QCDシミュレーションの最適化を行い, 性能を評価した. 2章では, Blue Gene/Pのアーキテクチャーについて, Blue Gene/Lとの比較とともに簡単に説明する. 3章では, メモリ性能, 通信性能についての性能評価結果について説明する. 4章では, アプリケーションの最適化の事例として, 格子QCDシミュレーションの最適化および性能評価について説明する.

表1. Blue Gene/LとBlue Gene/Pの比較

	Blue Gene/L	Blue Gene/P
Processor	PowerPC®440 700 MHz dual core	PowerPC®450 850 MHz quad core w/SMP
Main Memory	512MB/1GB DDR 5.6 GB/s	2GB DDR2 13.6GB/s
L3 cache	4MB	8MB
Peak performance	5.6 GFLOPS/node	13.6 GFLOPS/node
Torus network	175MB/s x12 = 2.1GB/s	425MB/s x12 = 5.1GB/s w/DMA
Tree network	350 MB/s x2 = 700 MB/s	850 MB/s x2 = 1.7GB/s

提出日：2007年8月29日　　再提出日：2007年12月11日

2. Blue Gene/Pのアーキテクチャー

Blue Gene/Pは, Blue Gene/Lのアーキテクチャーをもとに設計されており, 基本的なシステムの構成や筐体はそのままに, 性能を大きく向上させたシステムとなっている. **表1**にBlue Gene/LとBlue Gene/Pの主な違いについてまとめる.

最も大きな変更点は, プロセッサーコアがPowerPC® 440から1世代新しいPowerPC450になり, 1ノードあたり2コアから, 倍の4コアになり, 動作周波数が850MHzに引き上げられたことである. これによってノードあたり約2.4倍の性能向上がなされた. また, 新たにBlue Gene/Pでは, この4コアの共有メモリ並列化(SMP)に対応した.

Blue Gene/Lでは, 1つのノード内で1つのコアを計算に用いもう一方をMPI(Message Passing Interface)通信に用いるコプロセッサーモードと, 2つのコアを独立なプロセスとして動作させる仮想ノードモード(VNモード)があったが, Blue Gene/Pではコプロセッサーモードの代わりに1プロセスで4コアの共有メモリ並列を行うSMPモードと, 1ノードあたり4プロセスの仮想ノードモードが選択できる.

SMPモードを使用することにより, 1プロセスあたり2GBのメモリが使用できるようになり, 従来ではメモリ不足により動作しなかったプログラムを動作できる可能性が広がった. また, OpenMPによる共有メモリ並列化とMPI通信によるノード間並列化のハイブリッド並列化手法により, 従来よりも性能を引き出せるアプリケーションも出てくると期待できる.

また, ノードあたりのコア数が倍になったのに伴い, メインメモリおよびL3キャッシュのサイズが倍になり, メモリアクセス速度とネットワークの速度のクロックに対する比率も倍になっている. また, Blue Geneの通信網には, 三次元トーラスネットワークおよびツリーネットワークがある.

三次元トーラスネットワークは, 各ノードを三次元の格子状に並べ隣接するノード同士を相互に接続したネットワークで, 両端同士も接続される. 主にノード間のデータのやり取りに使用される. 三次元トーラスネットワークをコントロールするために, 新たにDMA(Direct Memory Access)エンジンが搭載された. これによりプロセッサーの負担なしで, 異なるノード間のメモリ間コピーを行うことができるようになった.

ツリーネットワークはノードを階層的な木構造で接続したネットワークで主に, データを配るブロードキャスト通信や, ある値の合計などを計算する集約通信に利用する.

3. Blue Gene/Pの基本性能評価

実際のアプリケーションの最適化を行う前に, Blue Gene/Pの基本性能を知るための簡単な性能評価プログラムを実施した. ここでは, そのプログラムと性能について説明する.

3.1 STREAMによるメモリアクセス性能

一般にメモリ速度を計測するベンチマークプログラムとしてSTREAMがある. STREAMベンチマークと同じ内容のプログラムを作成し, Blue Gene/Pの1ノードでの性能評価を行った. STREAMベンチマークの中から, triadベンチマークの性能評価について説明する. triadベンチマークは, 次のFORTRANプログラムのように2つの実数の配列のうち片方を実数倍して他方に加算し別の配列にコピーする処理である.

```
REAL*8 A(N), B(N), C(N), S
DO I=1,N
 A(I)=S*B(I)+ C(I)
ENDDO
```

この処理では, 配列の要素毎に乗算と加算の2つの演算を行うため, 配列サイズの2倍の演算回数となる. この演算回数を処理時間で割ることで1秒あたりの浮動小数点数演算回数(MFLOPS)を求め**図1**に横軸に配列のサイズ, 縦軸に1秒あたりの演算回数を示す.

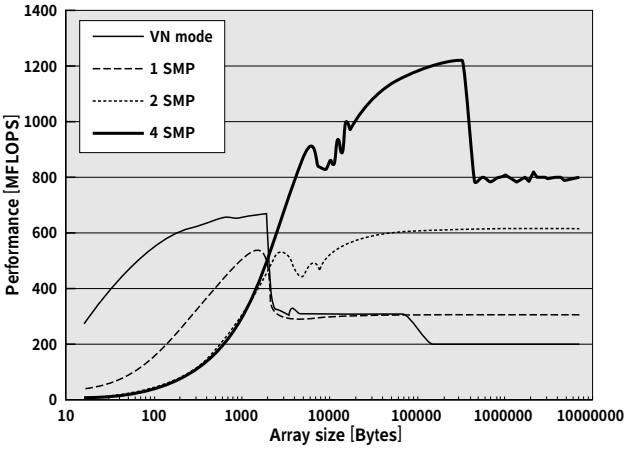

図1. triadベンチマークの測定結果

図1において, VN modeは仮想ノードモードを, n SMPは, SMPモードでn並列で実行した場合の結果を示す.

triadベンチマークはメモリ転送速度の影響を大きく受ける処理である. 仮想ノードモードでは, 大きく3つの区間に結果が分けられる. 一番左の部分はL1キャッシュに配列が収まる場合で, 中ほどで急激に性能が落ち

る部分から右がL3キャッシュに収まる部分，最後に落ち込む部分から右が，L3に収まらずにメモリからロードストアする部分である．1 SMPの結果と比較すると，L3のときの性能はほぼ同じであるが，L3に収まらない場合に，メモリ転送速度を4つのコアで分け合うために，速度が落ちているのが分かる．仮想ノードモードでは，L3キャッシュ以上を有効に利用するのが重要であることがよく分かる．

SMPモードでは，2並列以上の場合に並列スレッドを生成するタイミングでL1キャッシュが無効化されるため，L1キャッシュに収まる部分での性能があまりよくない．しかし，L3キャッシュを有効に利用できる長さの配列になると，並列度に応じて性能が上がってくる．SMPモードではL3キャッシュに収まるできるだけ長い配列を処理に用いるのが有効であることが分かる．

3.2 Ping-PongによるDMA転送速度の測定

三次元トーラスネットワークを介してデータ通信を行うために，Blue Gene/Pでは新たにDMAエンジンが搭載され，メモリ上のデータを直接別のノードのメモリ上にコピーできるようになった．ここでは，DMAを用いて2つのノード間のデータ転送速度を測定するためにPing-Pongベンチマークを作成して実行した．

Ping-Pongベンチマークでは，まず1つ目のノードが2つ目のノードにデータを送り，2つ目のノードはデータを受け取った後に，1つ目のノードにデータを送り返す．この往復の処理時間を測定し，データ転送速度を求める．

ここでは，三次元トーラスにおいてX方向に隣接する2つのノードの組を作り，その間を往復するデータ転送速度を測定した．**図2**にこのときの測定結果を示す．

仮想ノードモードの場合，DMAエンジン自身およびトーラスネットワークは4つのコアで共有されるため，転送速度がSMPモードに比べておおよそ4分の1になっているのが分かる．データ長が短い場合，メモリへのロードストアがキャッシュメモリの恩恵を受けるため，仮想ノードモードの場合に少し性能が上がっているのが分かる．アプリケーションの最適化では，十分にDMAの転送性能を引き出せる長さの配列を用いることが重要である．

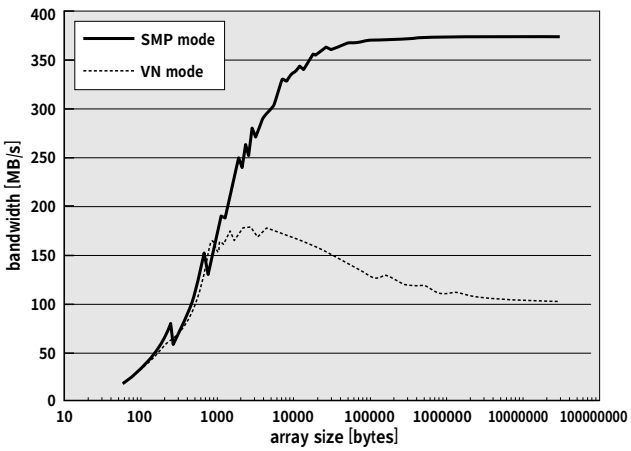

図2. Ping-PongによるDMA転送速度の測定

4. アプリケーション最適化と性能評価

Blue Gene/Pにおけるアプリケーション最適化の事例として，格子QCD（Lattice Quantum Chromo Dynamics：格子量子色力学）シミュレーションの最適化について述べる．

4.1 格子QCD

QCDは強い力の相互作用の基礎理論であり，原子を構成するハドロンと呼ばれる陽子や中性子などの構成要素であるクォークと呼ばれる素粒子とそれを結びつけるグルーオンのふるまいを記述する．QCDそのものをコンピューターのシミュレーションで解くことは問題の自由度が大きすぎて困難であり，QCDの自由度を減らすために四次元時空間を格子上に離散化した格子QCD［5］［6］が用いられる．格子QCDによるシミュレーションによって，物質の重さの起源［7］や，原子核内の強い力の起源［8］を解明することができている．これらの理論の解明には非常に大きな計算量が必要であり，Blue Geneのようなスーパーコンピューター上で効率よく演算を行う必要がある．

4.2 格子QCDシミュレーションの最適化

格子QCDシミュレーションのうち，最も多くの計算時間を要するのがWilsonディラック演算子［5］と呼ばれるクォークの伝播関数の演算である．Wilsonディラック演算子は，式（1）のような式であらわされ，四次元格子上の格子点に定義された4つの3自由度のクォーク$\delta(n, m)$が隣接する格子点のクォーク$\delta(n\pm\mu, m)$と，格子間に定義された3×3の複素行列で表現されたグルーオンU_μを介して相互に影響を及ぼしあう状態を記述したものである．ここでμは四次元空間の各軸に対応する1〜4の値である．

$$D(n,m)=\delta(n,m) -\kappa\sum_{\mu=1}^{4}\left\{(1-\gamma_\mu)U_\mu(n)\delta(n+\hat{\mu},m)+(1+\gamma_\mu)U_\mu^{\dagger}(n-\hat{\mu})\delta(n-\hat{\mu},m)\right\} \quad (1)$$

4.2.1 Wilsonディラック演算子の計算の最適化

Wilsonディラック演算子は，格子点に隣接する8方向

について, 4つの3成分の複素ベクトルと3×3複素行列の演算を行い, ガンマ行列(1±γ)を掛けた後に, 元の値にたし合わせる演算で構成される. これらの複素数の掛け算と足し算を効率よく行うことが重要である.

Blue Geneのダブル FPUを用いることで, 複素数の掛け算や足し算を非常に効率よく計算できる[4][9]. 複素数同士の掛け算は2つのダブルFPU命令で計算でき, 足し算は, 積和演算を用いることで掛け算と同時に計算することができ, 大変効率よく計算できる. 3成分の複素ベクトルと3x3の複素行列の掛け算は, 66個の実数演算からなるが, 18命令のダブルFPU命令で計算することができる. 最適な命令および命令のスケジューリングをインラインアセンブラを用いて記述し, 計算の最適化を行った.

4.2.2 Wilsonディラック演算子の並列化

Wilsonディラック演算子では隣接する格子間についてのみ影響を考慮する必要があるため, 分散メモリ並列化を行うのは比較的容易である. **図3**のように格子をある軸方向に分割し, 小さな格子に分割した場合, 分割した際にできた境界面同士でのみ, 相互に格子上のデータを参照することができれば, 計算することができる.

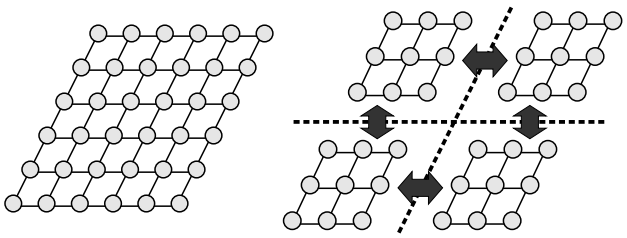

図3. 格子分割と境界面の通信

このときの境界面同士のデータの交換をなるべく短時間で行うことが重要である.

Blue Geneの各ノードは三次元トーラスネットワークによって相互に結合されている. このとき, 各ノードはXYZ正負の6方向に隣接するノードとのみ通信する場合, 最も効率よくデータの交換を行うことができる. また, 各ノード内のプロセッサーコア間のデータ交換を仮想的な四次元目のトーラスとみなすことができる. そこで, 四次元の格子を, 四次元のトーラスネットワークを構成する各軸方向のノード数で分割することで, 分割された格子間のデータ交換を, 隣接するノード間の通信に限定することが可能である[4].

ここで, Wilsonディラック演算子で用いるクォークWおよびグルーオンUの配列はFORTRANの記述で次のようなものである.

```
COMPLEX*16 W(3, 4, LX, LY, LZ, LT)
COMPLEX*16 U(3, 3, LX, LY, LZ, LT, μ)
```

LX, LY, LZ, LTは四次元格子の各方向の格子点数, μは四次元空間の各軸1~4に対応している.

Blue Gene/Pでは, 各ノードに4つのプロセッサーコアがあり, 仮想ノードモードでは4コア間で格子を分割し, 境界部分のデータを共有メモリを用いて交換する. SMPモードでは, 処理のループを共有メモリ並列化することでそのループの方向に4つに分割したのと同じ効果を得る. 上記のような並びの配列を使用する場合, プロセッサーコア間のデータ交換は高速であるため, 配列の一番内側であるX方向をコア間で分割するのが最も効率が良い. SMPモードにおいても, ループの一番内側を並列化しておけば, 仮想ノードモードと同じように各コアは常に同じメモリ領域をアクセスできるためキャッシュメモリ内のデータを効率よく再利用できる.

4.2.3 DMAによるデータ交換の最適化

DMAによるデータ転送は, 図2に示したように, ある程度の配列の長さがないと十分に性能を出すことができない. また, DMAによる転送では, 連続したデータのかたまりを一度に送受信することしかできないため, クォークの配列から直接境界部分となるデータを一度に送り出すことはできない. さらに, ガンマ行列の性質から, 隣のノードに送るデータは, 4×3自由度のクォークの半分の2×3成分で済むことが知られている[4][9]が, この性質を活かして通信量を減らすには直接送ることはできない.

そこで, Blue Gene/Pでは, 四次元格子のXYZT正負それぞれの方向について, 計8方向分の境界部分の隣に送信するデータをあらかじめ別々の配列に用意し, DMAを用いてそれぞれの配列について一度に隣のノードに送

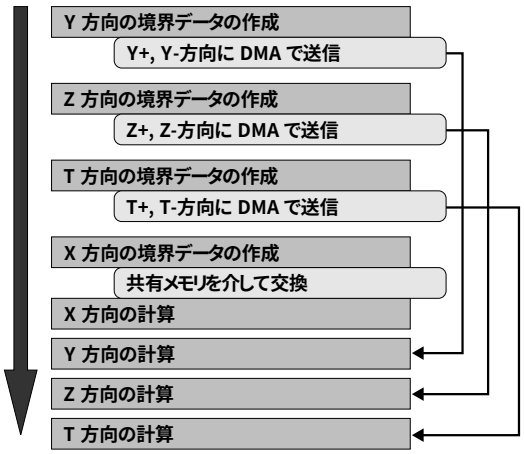

図4. 処理の重ね合わせによる最適化

るようにすることで転送速度を最適化した.

また, DMAがデータを送信している間は, 各プロセッサーコアは別の処理を実行することができ, DMA自身も同時に別の方向にデータを送受信することが可能となっている. そこで, **図4**のように各方向への送信のタイミングをずらすことで, データの送信中に別の方向の送信するデータを作成して送ったり, 受け取ったデータを使用して計算を行ったりすることで, 通信時間を短く見せかけることで最適化を行った.

4.3 Blue Gene/Pの格子QCDの性能評価

Blue Gene/Pにおける格子QCDの性能評価プログラムとして, 実際の格子QCDシミュレーションで多用する, 反復法の中でWilsonディラック演算子を繰り返し用いるようなプログラムを作成した. クォークおよびグルーオンに適当な初期値を与え, 反復法としてBiCGStab法を用い, 解が収束するまで処理を繰り返した. また, 実際の多くの格子QCDシミュレーションでは, 前処理によって, 格子点の座標(x, y, z, t)について$x+y+z+t$が奇数になる点と偶数になる点に分けて演算を行う, 前処理つき反復法を用いている. これによって収束までの反復回数を約半分に減らすことができるが, このような前処理つきの反復法に対応したWilsonディラック演算子についても最適化を行い, 評価をした. 本論文では, Blue Gene/Pの1/2筐体(512ノード)を用い, 仮想ノードモード, SMPモード, の両モードについて評価プログラムの最適化を行い, 性能測定をした.

4.3.1 Blue Gene/LとBlue Gene/Pの性能比較

仮想ノードモード1/2筐体におけるWilsonディラック演算子の実効性能(ピーク演算性能に対する単位時間あたり実測演算量の割合)のBlue Gene/LとBlue Gene/Pを, 前処理なしの場合について比較したものを**表2**に, 前処理ありの場合について比較したものを**表3**に示す. 表内の項目xBG/LはBlue Gene/Lに対するBlue Gene/Pのノードあたりの性能向上比を示す.

表2. 前処理なしの場合の実効性能比較

格子サイズ	Blue Gene/L	Blue Gene/P			
		仮想ノードモード	x BG/L	SMPモード	x BG/L
16x16x16x16	27.74 %	17.54 %	x 1.54	17.84 %	x 1.56
16x16x16x32	30.59 %	20.79 %	x 1.66	21.10 %	x 1.67
24x24x24x24	33.34 %	27.31 %	x 1.99	26.63 %	x 1.94
24x24x24x48	33.58 %	30.51 %	x 2.21	29.69 %	x 2.14
32x32x32x32	36.28 %	33.30 %	x 2.23	31.39 %	x 2.10
32x32x32x64	36.40 %	35.07 %	x 2.34	34.05 %	x 2.27

表3. 前処理ありの場合の実効性能比較

格子サイズ	Blue Gene/L	Blue Gene/P			
		仮想ノードモード	x BG/L	SMPモード	x BG/L
16x16x16x16	24.70 %	11.70 %	x 1.15	11.87 %	x 1.17
16x16x16x32	26.77 %	15.42 %	x 1.40	15.16 %	x 1.38
24x24x24x24	31.19 %	21.63 %	x 1.68	20.19 %	x 1.57
24x24x24x48	31.17 %	24.53 %	x 1.91	22.33 %	x 1.74
32x32x32x32	34.32 %	26.78 %	x 1.89	24.63 %	x 1.74
32x32x32x64	34.58 %	28.66 %	x 2.07	26.53 %	x 1.86

Blue Gene/L, Blue Gene/Pともに格子サイズが大きいほど性能がよく, 前処理なしの場合に性能が若干落ちている. 格子サイズが小さい場合にBlue Gene/Pの性能が落ち込んでいることが分かる. 理想的にはノードあたり約2.4倍の性能向上比が望ましいが, 実際には最も性能が出ているケースでも2.34倍であった. この原因については後述する.

また, Blue Gene/Pでは, SMPモードにおいても仮想ノードモードに近い性能が出ているが, 仮想ノードモードよりもDMAによる通信が速いにもかかわらず性能が若干悪い. これは, SMP並列化によるオーバーヘッドが関与していると考えられる.

Blue Gene/Lとの比較についてノードあたり2.4倍を下回ったのは2つの原因が考えられる.

1つ目は, L1キャッシュの方式の違いである. Blue Gene/Lでは, ライトバック方式(L1キャッシュにのみデータを書き込み, 後でメモリに反映させる方式)であり, 一方Blue Gene/Pはライトスルー方式(L1キャッシュとメモリの両方にデータを書き込む)である. 一般的に, ライトバック形式の方が短時間でデータを書き込めるため性能が出やすい. Blue Gene/Pでは長めの配列について同じ処理を行い, L3キャッシュやハードウェアプリフェッチを有効に使うことでライトスルー方式の性能の低下を補うような最適化を行うのが有効である. なお, Blue Gene/Lにおいて, ライトスルー方式を使用した場合, 格子QCDでは3割ほど性能が落ち, ライトスルー同士で比較すると, Blue Gene/Pの性能向上比は最大でノードあたり3.1倍になり, 理想値の2.4倍を上回った.

2つ目はノード間のデータ送受信の方法の違いである. Blue Gene/Lでは, トーラスネットワークを利用した通信をそれぞれのプロセッサーコアが処理しなければならなかったが, 制限は多いものの, レジスタからデータを直接送受信できるなどの利点があり[4], 境界部分のデータを一時的に別の配列に保存する必要がなかったため, これも小さいデータには有利であった. 一方Blue Gene/PではDMAで送受信するために一度境界

データを配列にストアする必要があり, この部分のオーバーヘッドがデータが小さい場合に比較的大きくなってしまっている. また, 前処理ありの場合には配列のサイズがなしの場合の半分になるため, 性能が落ちていると考えられる.

4.3.2 Blue Gene/Pにおけるスケーラビリティ

プロセッサーコア数と格子のサイズを変えて測定を行い, 最適化された格子QCD性能評価プログラムのスケーラビリティを評価した. ここでは, 仮想ノードモードを用い32ノード(128コア)から512ノード(2048コア)までを使用し, 前処理なしの場合について測定を行った.

まず, 全体の格子サイズを一定にし, 使用するノード数を変えた場合について実効性能を測定した. このときの測定結果を**図5**に示す.

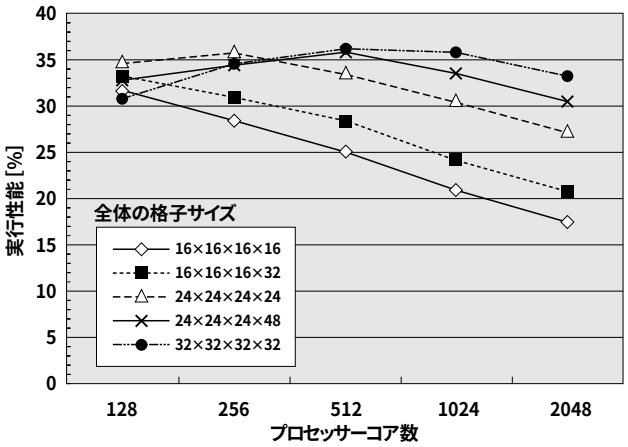

図5. 全体の格子サイズ一定の実効性能

図5では, 使用するプロセッサーコア数が多くなるのに反比例してコアあたりの格子サイズが小さくなる. 測定結果ではコア数が増えるのにつれて性能が落ちているのが分かる. これは, コアあたりの格子サイズが小さくなると, DMAによる通信の効率が悪くなり, また, コアあたりの計算時間よりも通信時間の割合が大きくなるためと考えられる.

ただし, 全体の格子サイズがある程度大きい場合は逆にノード数が小さいときにも性能が落ちている. これは, プロセッサーコアあたりの格子サイズが大きすぎてキャッシュの再利用率が落ちているためと考えられる. 続いて, プロセッサーコアあたりの格子サイズを一定にし, ノード数が増えると全体の格子サイズが比例して大きくなるような格子サイズを用意して実効性能を測定した. このときの測定結果を**図6**に示す.

図6に示す結果からは, コアあたりの格子サイズが同一であれば, 並列度を増やしても性能が落ちずにほぼ

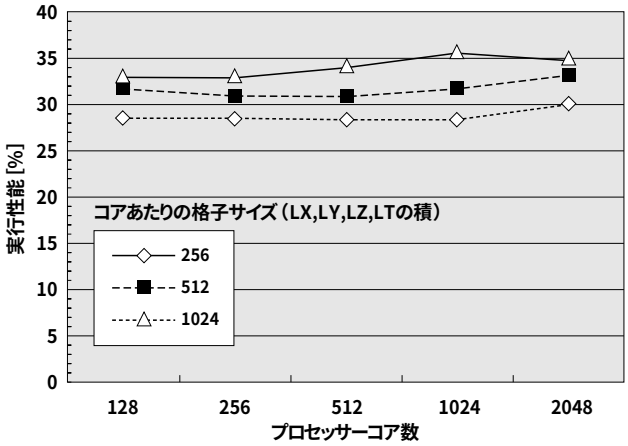

図6. コアあたり格子サイズ一定の実効性能

一定の性能を示すことが分かった.

図5および図6に示した結果は, 前処理ありの場合およびSMPモードにおいてもほぼ同じような結果を得ることができた. また, Wilsonディラック演算子では, 隣接ノードとの通信に限られるため, さらに1筐体(きょうたい)以上にノード数が増えても同じような性能の傾向を示すことが予想できる.

以上の結果から, 実際の格子QCDアプリケーションを運用する際には, 適切な格子サイズと使用するノード数を選択することで, 効率よく計算することが重要である. 一般的な格子QCDアプリケーションは, モンテカルロシミュレーションを用いて統計的にシミュレーションをするため, 解こうとする問題の格子サイズによっては, 大きな構成のシステムで1つの問題を解くよりも, 複数の小さな構成のシステムで複数の問題を解く方が効率よく計算できる場合があり, この結果を活用して効率よく実行することができる.

5. おわりに

Blue Gene/Pの性能を知る上で, 基本性能評価を実施し, 格子QCDシミュレーションの最適化および性能評価を行った. 格子QCDシミュレーションの性能は, Blue Gene/Lと比較すると, 理想値であるノードあたり2.4倍の性能にはやや劣るものの, L1キャッシュがライトスルー方式に変更になったにもかかわらず, 30%前後の実効性能を得ることができ, また, スケーラビリティに関しても良い結果が得られた. Blue Gene/Lの成果に続き, 格子QCDはBlue Gene/Pの優位性を示すことのできるアプリケーションである.

また, 最適化において, DMAが搭載されたことも大きいと感じた. 従来は, 計算と通信を重ねるために非常に複雑なプログラムを書く必要があったがDMAの

おかげで非常にシンプルなプログラムで良い性能を出せるようになった.

SMPによるノード内の並列化によって最適化の幅も大きく広がったと感じた. 格子QCDの最適化もまだ改善の余地があり, 今後さらなる性能の向上が見込まれる.

今後他のアプリケーションについても, DMAやSMPを用いて最適化を行い, Blue Gene/Pの優位性を示して行きたい. 特に, 格子QCDと同じように, 格子状に離散化された空間で問題を解くようなもの, 例えば, 流体シミュレーションや熱シミュレーションのようなアプリケーションには, 同じような最適化の手法が適用できると考えられる.

謝辞

性能評価にはIBM T. J. Watsonリサーチセンターの Blue Gene/Pを使用させて頂きました. Blue Gene/Pシステムへのアクセスおよび本研究に協力してくださったIBMリサーチのJames Sexton氏, 研究に関していつも助言をくださっているIBM東京基礎研究所ディープコンピューティングプロジェクトのメンバーの皆様に感謝いたします.

参考文献

[1] The IBM System Blue Gene Web Page : http://www.ibm.com/servers/deepcomputing/bluegene.html
[2] A. Gara and et al. : "Overview of the Blue Gene/L System Architecture," IBM Journal of Research and Development, Vol. 49, No. 2/3, pp.195-212 (2005).
[3] 清水 茂則, 寒川 光, 土井 淳 : "Blue Gene/L システム ースーパーコンピューティングへのグランドチャレンジー, " ProVISION, No.48, pp57-62 (2006).
[4] 土井 淳, 寒川 光, 松古 栄夫, 橋本 省二 : "Blue Geneに適した格子QCDプログラムの超並列化," 情報処理学会論文誌 : コンピューティングシステム, No. SIG 7 (ACS14), Vol.47, pp.114-123 (2006).
[5] K.G. Wilson : *New Phenomena in Subnuclear Physics*, edited by A. Zichichi, Plenum press New York, ISBN0-306-38181-8 (1977).
[6] 青木 慎也 : 格子上の場の理論, シュプリンガー・フェアラーク東京, ISBN4-431-71172-4 (2005).
[7] H. Fukaya, et al. : "Two-Flavor Lattice-QCD Simulation in the (Regime with Exact Chiral Symmetry," *Physical Review Letters*, vol. 98, number 17, p.172001 (2007).
[8] N. Ishii, S. Aoki and T. Hatsuda : "Nuclear Force from Lattice QCD," Physical Review Letters, vol. 99, number 2, p.022001 (2007).
[9] Pavlos Vranas, et al. : "The Blue Gene/L Supercomputer and Quantum ChromoDynamics," *SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, p.50 (2006).

日本アイ・ビー・エム株式会社
東京基礎研究所
研究員
土井 淳 Jun Doi

[プロフィール]

1999年日本IBM入社. 東京基礎研究所にて, CAD/CAE関連のプロジェクトや, スーパーコンピューター関連のプロジェクトに従事. 特にBlue Geneにおけるアプリケーション最適化に力を入れている.

doichan@jp.ibm.com