平成 28 年度　利用報告書

課題名（タイトル）：

格子量子色力学クォークソルバーの FX100 向け最適化

利用者氏名：石川健一

所属：計算科学研究機構　連続系場の理論研究チーム

1．本課題の研究の背景、目的、関係するプロジェクトとの関係

格子量子色力学(格子 QCD)は素粒子理論の強い力に関する力学（量子色力学 QCD）を第一原理から計算することのできる理論である。強い力は陽子や中性子、中間子などの性質とともに素粒子標準模型のクォークセクターの力学も記述している。格子量子色力学では連続的な 4 次元時空を格子に離散化することで量子力学の経路積分をモンテカルロ法により数値計算することが可能な体系となっている。物理量の期待値の計算のためのこのモンテカルロ法ではクォークの運動方程式(離散化された時空上の差分方程式)を解くところ(クォークソルバー)が最も時間のかかるところとなっている。格子量子色力学全体の数値計算がうまく進むためには、この運動方程式を効率よく高速に解くことが必要である。クォークソルバーの最適化・高速化は新しい計算機が登場するたびに行わなくてはならない。現在、主流の格子 QCD 計算の計算資源量は 100～300 TFlops*Years のものでスーパーコンピュータを用いて行われている。

現在国内では理研も含めて FX100 世代の計算機の導入が進んできているため、FX100 向けの高効率コードを提供することで分野全体の計算機資源の有効利用に寄与することができる。

本課題では、京コンピュータ用に開発された計算コードに FX100 のアーキテクチャに即した最適化を行うことを目的とする。

前年度(平成 27 年度)に行った最適化では Fortran95 言語を用いた FX100 向けの最適化を施したが、理想的なメモリ・キャッシュビジー率を達成できていなかった。本年度（平成 28 年度）は、キャッシュ利用効率上昇による性能向上をめざし、引き続き最適化を行った。

2．具体的な利用内容、計算方法

最適化対象のクォークソルバーは、格子 QCD の Wilson/Clover フェルミオン運動方程式から導出される 4 次元時空格子に対する大規模な連立方程式を解くプログラムである。連立方程式の係数行列は大規模疎行列であり、いわゆるステンシル系の問題である。解法としては Krylov 部分空間反復法の BiCGStab 法を用いており、数値精度はメモリバンド幅やキャッシュ効率を有効利用できるように単精度を用いている。前処理としては領域分割前処理法を用いており、差分のステンシル計算を 2 色のブロックに分割し、MPI プロセス内には色の異なる 2 つのブロックが収まるようにブロック分割を行っている。ブロック内演算とブロック間演算が分離でき、通信はブロック間演算の時に必要となる。

京向けの計算コードでのクォークソルバーは、京の 8 コアの有効利用と C 言語の SIMD 組み込み関数(intrinsics)を用いた倍精度 2 成分の SIMD 最適化が施してある。その性能はノードあたり 4 次元格子サイズ 6^3x12 の時、疎行列ベクトル積演算部の演算効率は約 50%、ソルバー全体の演算効率は約 26%である。

一方 FX100 では、コア数が 16 コアをひとまとめにしたもの(CMG)が 2 つ実装された 32 コア(2CMG)であり、1CMG あたりに一つのアシスタントコアが設けられている。さらに、SIMD 長さは倍精度が長さ 4、単精度が長さ 8 に拡張されている。

昨年度までにこのクォークソルバーに FX100 アーキテクチャに適した最適化を Fortran95 言語を用いて実施した。具体的にはメモリ上のデータレイアウトの変更による SIMD 8 への最適化と OpenMP 16 スレッドへの最適化であった。この結果、MPI プロセスあたり格子サイズが 12^3x24 、MPI プロセス数が 2x2x2x2=16 プロセス、8 ノード実行の時、 MPI プロセスあたりの単精度ソルバー全体の効率は約 11.1% となり L1D, L2D, メモリビジー率はそれぞれ 37%, 21%, 32 % であった。最も計算量が多いブロック内差分演算では、効率は 24.3%, ビジー率はそれぞれ 65%, 26%, 39% であった（効率は倍精度ピーク性能に対するものであるので単精度演算の場合はこの半分が真の効率であるが、PA 情報エクセル表は倍精度ピーク性能に対する比を算出するので以下でもその値をそのまま記載する）。メモリビジ

一率や L2D ビジー率に問題がある様であった。

昨年度の結果からメモリビジー率を上げるようなプリフェッチの挿入を入れる必要があると考え、本年度はプリフェッチを入れて性能を上げることを行った。さらに、本年度から C/C++ 言語において SIMD 組み込み関数がサポートされ、 SIMD 演算を直接記述できるようになった。 さらに、C/C++言語においては、ソフトウェアプリフェチ(_builtin_prefetch)がサポートされている。そこで前年度作成した Fortran95 言語版のプログラムを再び C 言語に変更し、SIMD 組み込み関数を用いた最適化を施した。

最適化は精密 PA 情報を基に繰り返した。演算量の少ないブロック間差分演算においてスレッドロードインバランスが生じていることが分かったので、領域の袖部分にあたるデータを格納する専用の配列を用意し格子点をめぐるループ構造を変更しスレッドインバランスを解消した。

3．　結果

区間	性能 [GFlops/MPI] (効率)	L1D ビジー率	L2D ビジー率	メモリビジー率
単精度ソルバー全体	91.4 (18.1%)	47%	36%	61%
ブロック内差分演算	167.9 (33.2%)	68%	39%	68%
ブロック間差分前処理演算	86.8 (17.2%)	40%	47%	71%
ブロック間差分後処理演算	46.7 (9.24%)	29%	41%	63%

表 1 ソルバーの各区間の性能とキャッシュ・メモリビジー率

MPI ランクあたりの格子サイズを 12^3x24x48 とした。これは最内側ループ（4 次元格子の T 方向）長を長く取る（=48）ことと、T 方向の再内ループと内側から 2 番目のループ（4 次元格子の Z 方向）のループを転置して T 方向の並列分割による袖領域の計算を Z 方向のループ(長さ=24)で SIMD ベクトル化するためにこのように格子サイズを変更した。

表1に、ソルバーの各区間の性能とキャシュ・メモリビジー率を示す。2 段目のブロック内差分演算の性能は効率約 33%を達成できた（昨年版 Fortran95 は約 24.3%）。この性能向上は、SIMD 組み込み関数による SIMD 化率の向上（Fortran95 版:約 74%、C 言語組み込み版：約 86%）によるものとプリフェッチ挿入によるメモリビジー率の向上による。FX100 の 16 コアの性能は倍精度で 505.6GFlops であり、1CMG（=16 コア）あたりのメモリバンド幅は 120GB/s と考えられるが、ブロック内差分の演算密度は約 0.89Byte/sec （単精度）であるのでメモリバンド幅律速である。メモリバンド幅のみから予測される性能は 120/0.89=134GFlops であるので、測定された値はキャッシュの効果を含んでいるものと考えられる。

ブロック間差分演算については 4 次元格子のブロックの袖領域を OpenMP 16 スレッドに最大限に均等に割り振るようにスレッド番号毎に担当袖領域を直接実行するように変更した。その結果、ブロック間差分前処理部分（領域袖データの MPI 送出配列へのパッキング）の効率は Fortran95 版で約 1%だったものが、C 言語組み込み版で約 17%となり、後処理部分（MPI 受取配列から隣接データを取り出し、領域表面での差分計算を行なう）においては Fortran95 版で約 3.6%だったものが C 言語組み込み版では 9.24%に性能向上ができた。L2 ビジー率が L1 ビジー率に比べて低くなっている傾向は昨年度の Fortran95 版でも同様であるが、全体的にビジー率は上がっている。さらなるビジー率の向上を目指したが、これ以上のプリフェッチ挿入は却って演算性能を落とす結果となった。単精度ソルバー全体の性能としては効率約 18.1%を達成できた。昨年度の Fortran95 版の MPI あたりの格子サイズは 12^3x48 であり、本年度の計測したプログラムと同一ではなく、むしろサイズが増えているが効率の向上により、少ないノード数で効率よい計算ができるようになったと考えている。

4．　まとめ

本課題では、格子 QCD 計算で最も時間がかかるクォークソルバーの FX100 への最適化を行った。昨年度は C 言語での SIMD 組み込み関数の正式サポートができていなかったが、本年度初めに SIMD 組み込み関数がサポートされ、SIMD 効率を上げたコードを作成することができた。加えて ソフトウェアプリフェッチを使用することでキャッシュ・メモリビジー率の向上ができた。本年度実

施した最適化により単精度ソルバー全体の効率は 18.1％（性能は 91.4GFlop/s）となり、昨年度実施した版に比べ 1.63 倍性能が向上した。

５．　今後の計画・展望

最適化は終了したと考える。コードの整理を行いユーザに提供していきたい。