

2P11 OpenFMOにおけるFock行列計算ルーチンのGPGPU化について

○梅田宏明、埴敏博、庄司光男、朴泰祐

筑波大学 計算科学研究センター

序

高性能科学技術計算の急速な発展に対し、量子化学計算アプリケーションは必ずしも十分な対応が出来ていない。HPC向け並列計算機のうち汎用CPUを利用した並列計算機についてはOpenMP/MPIハイブリッド並列による対応も進みつつあるが、GPGPUのようなアクセラレータを用いた高性能計算機については限られた研究者が利用を検討するに留まっている。そこで我々はHF計算において最も計算量の多いFock行列計算についてGPGPUによる高速化を試みた。ターゲットプログラムとして超並列計算機向けのFMO計算プログラムであるOpenFMO[1]を取り上げ、そのフラグメント計算のHF計算部分だけを切り出したコードについてCUDA[2]によるGPGPU化を行なった。

OpenFMOは九州大学の稲富らが開発している超並列計算機向けフラグメント分子軌道(fragment molecular orbital, FMO)計算プログラムである。このプログラムはC言語で記述されており、CUDA実装でも扱いやすい。またOpenMP/MPIハイブリッド並列による超並列計算機向けの並列化だけでなく、RPC化などによる耐故障性の検討にも使われるなど次世代の量子化学計算アプリのプラットフォームとしても利用が期待されている。

GPGPU化実装

GPGPUを用いて高性能計算を実現するためには、大量の演算コアを有効に使うことが必要である。このために重要なのがメモリアクセスの効率化とプログラムのSIMD並列化となる。また特にFock行列計算においては各演算コアが計算した二電子積分をFock行列に加算する操作が存在するため、Fock行列の取り扱い手法についても工夫が必要である。本研究では1)行列加算手法、2)Schwarzの不等式によるスクリーニング、3)動的負荷分散、4)基底関数のソートの4つの改良を検討した。

1) 行列加算手法(Non-atomic)

GPGPU 1台あたり数百もの演算コアが利用可能であるのに対し、利用できるメモリは僅かに数GB程度しかない。このため多くの並列実装で行われているようなFock行列を演算コアが独立に保持する実装は現実的には不可能である。一方でFock行列を複数演算コアで共有する場合にはFock行列への加算にコストの高い排他制御が必要となり、性能が出ないことも知られている。そこで我々はFMO-MO計算のために開発した並列分散共有Fock行列計算アルゴリズム[3]をGPGPUに適用することでアトミック加算を回避する実装を行った。

2) Schwarzの不等式によるスクリーニング(Pre-screening)

ホストCPUと違いGPUはSIMD動作をするため、Schwarzスクリーニングによる積分計算の省略のための分岐が効果的に動作しない可能性がある。そこで我々はスクリーニング処理だけを積分計算の前に実行するよう改良を行い、効果的なFock行列計算を試みた。

3) 動的負荷分散(Dynamic LB)

GPGPUでは演算コアが多いため、静的な負荷分散では十分な負荷バランスが得られない可能性がある[4]。そこでGPUボード内についての動的負荷分散コードを実装し、効率の良い並列化を目指した。

4) 基底関数のソート(Sorting)

積分計算そのものの実装についても SIMD 動作を考える必要がある。OpenFMO で利用している小原積分では積分計算において原始シェルペアについての二重ループが存在しており、効果的な SIMD 動作のためにはこのループ長を揃えることが望まれる。我々は縮約シェルペアを原始シェルペア長でソートすることにより、これを実現した。

ベンチマーク

ベンチマーク計算は筑波大に導入されている GPGPU クラスタシステム”HA-PACS”を利用して実行した。HA-PACS の 268 台ある計算ノードには、それぞれ 2 台の 8 コア Intel E5 CPU(Sandy Bridge-EP, 2.6GHz)と 4 台の Fermi 世代の GPGPU(NVIDIA M2090 GPU)が装着されており、InfiniBand により接続されている。

グリシン 5 量体の HF/6-31G(d)計算をサンプルとして、HA-PACS の GPU1 台で実行した場合の性能をホスト CPU 1 コアからの性能向上として示した。図 1 では (ps,ps)タイプまでの積分についての性能向上を示している。ナイーブ実装から各ステップで性能が向上し、スレッドブロック数・スレッド数のチューニング後(Tuning)の性能としては CPU 1 コア比で 13~22 倍という性能向上が得られている。

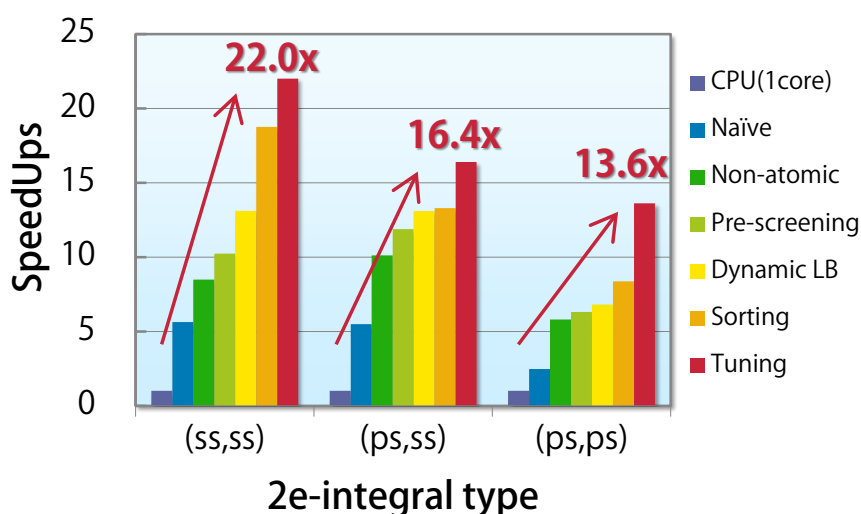


図 1 GPGPU 化による Fock 行列計算の速度向上

謝辞: 本研究で使用した HF 計算コードは九州大学の稲富らによる OpenFMO プログラムの一部を抜粋して提供していただいている。また本研究の一部は文部科学省特別経費「エクサスケール計算技術開拓による先端学際計算科学教育研究拠点の充実」事業、および JST-CREST 研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」、研究課題「ポストペタスケール時代に向けた演算加速機構・通信機構統合環境の研究開発」による。

[1] OpenFMO; <http://www.openfmo.org/OpenFMO/index.html>.

[2] NVIDIA: CUDA C Programming Guide Version 4.2; <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>

[3] Umeda, H., Inadomi, Y., Watanabe, T., Yagi, T., Ishimoto, T., Ikegami, T., Tadano, H., Sakurai, T. and Nagashima, U., *J Comput. Chem.*, **31**, 2381-2388(2010).

[4] Umeda, H., Inadomi, Y., Honda, H. and Nagashima, U., *J Comput. Chem.*, **30**, 826-831(2009).