

## 2P02 OpenFMO における Fock 行列計算の GPGPU による 高速化への取り組み

○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 化を行なっている。

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。これらの困難に対し、ここまで我々はいくつかのアイデアを元に CUDA 実装を改良し高速化を実現してきた[3]。一方で GPU のアーキテクチャの進歩も急速に進んでおり、K20 等の Kepler 世代の GPU ではスレッドごとのレジスタ数制限の緩和やアトムック演算の高速化などが改善されている。前回発表では Fermi 世代の GPU である M2090 に対し性能評価を行なったが、今回は Kepler 世代の GPU である K20 を用いた性能評価を行なったのでそれを紹介する。また前回の実装で問題となった作業用配列増加に対応するため、より細粒度の並列化実装も行なったので当日はその結果も報告する予定である。

### 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”および TCA 部評価システム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。一方 HA-PACS TCA(Tightly Coupled Accelerator)部評価システムには Kepler 世代の GPGPU(NVIDIA K20 GPU)が装着されている。

グリシン 5 量体の HF/6-31G(d)計算をサンプルとして、GPU1 台で実行した場合の各積分タイプごとの性能をホスト CPU 1 コアからの性能向上として示した(図 1)。ここで凡例に表記した“K20(65,255)”は

「K20 GPU を利用しブロック数 65、スレッドごとの register 数を最大 255 まで」と設定した場合の性能である。また“K20(130,255)a”は“K20(133,255)”の条件のもとで Fock 行列の加算にアトミック演算を利用した場合になる。ブロック数やレジスタ数を十分な大きさに設定した場合には K20 の性能が M2090 を上回っていることがわかる。特に軌道量子数の大きい場合にこの性能向上が顕著であり、ローカルメモリに置かれた作業用配列へのアクセスがボトルネックになっていることが伺える。また K20 においてはアトミック加算を利用したことによるオーバーヘッドが多くの場合で小さくなっている。アトミック加算を露に利用することにより、他の積分タイプの実装作業において実装コストを減らすことも期待できる。

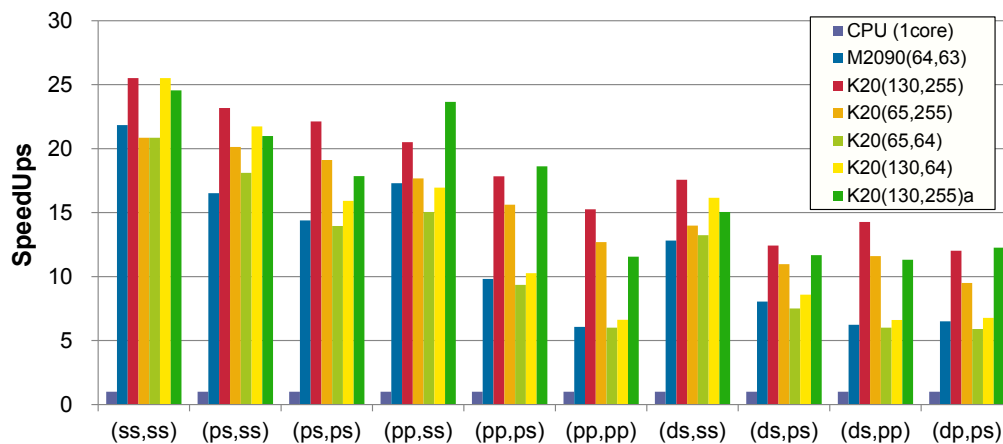


図 1 GPGPU による Fock 行列計算の性能向上

**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] 梅田宏明, 埴敏博, 庄司光男, 朴泰祐, 稲富 雄一, “フラグメント分子軌道法における Fock 行列計算の GPGPU 化”, 情報処理学会論文誌コンピューティングシステム(ACS), **44** (採録決定)