

2P115

OpenFMO プログラムの GPGPU 化

(筑波大¹, 東大²) ○梅田宏明¹, 埴敏博², 庄司光男¹, 朴泰祐¹, 重田育照¹

Development of GPU-accelerated OpenFMO program

(Univ. Tsukuba¹, Univ. Tokyo²) ○Hiroaki Umeda¹, Toshihiro Hanawa², Mitsuo Shoji¹,
Taisuke Boku¹, Yasuteru Shigeta¹

序

GPU 等の演算加速装置を用いた高性能科学技術計算システムへの対応は、量子化学計算においても重要なトピックとなっている。これまでに我々は Hartree-Fock(HF)計算のホットスポットである Fock 行列計算の GPGPU 化[1]を行ってきた。一方で大規模分子軌道計算を実現するには、大規模計算に適した計算手法を利用することも重要である。フラグメント分子軌道(FMO)法[2]はそのような大規模計算向け分子軌道計算手法の一つであり、フラグメント間の相互作用計算が可能である等の化学的な特徴も合わせ実用的にも広く利用されるようになってきている。我々は大規模分子軌道計算の実現に向けて FMO 計算プログラム実装の一つである OpenFMO [3]の GPGPU 化に向けた開発を行っており[4]、本発表ではその概要と現時点での性能評価について発表する。

実装

FMO 計算でボトルネックとなりうるのは Fock 行列計算とフラグメント間の静電相互作用(ESP)計算になる。このうち Fock 行列計算については、これまで我々が行ってきた HF 計算における Fock 行列計算の GPGPU 化コードを導入した。ESP 計算のうち特に二つの近接フラグメント間の静電相互作用を計算する 4 中心フラグメント間クーロン相互作用計算(4C-IFC)のアルゴリズム構成は Fock 行列計算と類似しており、Fock 行列計算の GPGPU 化で行なった手法の多くを活用できる。

4C-IFC 計算の具体的な実装方針としては、フラグメント A のシェルペアについてのループを GPU のブロックに配分し、フラグメント B についてのシェルペアループをブロック内のスレッドに配分する並列化を行った。この並列化アルゴリズムの場合にはスレッドブロック内でのリダクション処理さえできれば Fock 行列計算の GPGPU 化で問題となった行列加算における排他制御が必要なくなるため、比較的簡単な実装が可能である。また Fock 行列計算の GPGPU 化と同様に、シェルペアの並べ替えや Schwarz 不等式によるスクリーニングプロセスの分離、さらにはブロック間での動的負荷分散なども適用している。

性能評価

実装した GPGPU 化 OpenFMO の性能評価は筑波大学の HA-PACS ベースクラスタ[5]を用いて行った。HA-PACS ベースクラスタの計算ノードには 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)、および 128GB のメモリが搭載されており、それらが InfiniBand QDR2 ポートにより接続されている。また複数 GPU を活用するためノードごとに 4MPI プロセスを起動し、それぞれのプロセスが OpenMP 並列で 4 CPU コアと 1 台の GPU を利用することとした。コンパイルや実行には Intel コンパイラ 15.0.2, CUDA 6.5.14, IntelMPI5.0 をそれぞれ利用した。

性能評価としてクランビン分子(642 原子、20 フラグメント)の FMO-HF/6-31G(d)計算を HA-PACS の 8 ノードを利用して実行した。比較のため CPU による直接計算の他、SCF 中に積分を保存しておく in-core 手法も行っている。Table には全実行時間[s]に加え FMO 計算の主たる計算過程であるモノマー SCC、ダイマーSCF および遠隔ダイマー計算(ES dimer)の実行時間[s]を示した。また CPU による直接計算からの GPGPU 化による性能向上(speedup)も示している。GPU 化により全ての計算過程で 3~4 倍の高速化となっており、計算全体としては 3.3 倍の高速化を実現している。

Table 1 Elapsed time[s] and speedups for FMO calculation of Crambin (642 atom, 20 fragments) with/without GPU-acceleration on HA-PACS base cluster.

Crambin				
	CPU	CPU	CPU+GPU	Speedups
	Direct	In-core	Direct	Direct
#node	8	8	8	8
SCC [s]	629.8	632.2	207.1	3.0
Dimer SCF [s]	1,266.2	857.3	345.5	3.7
ES Dimer [s]	43.3	42.5	10.5	4.1
Total [s]	1,961.1	1,586.0	590.8	3.3

謝辞: 本研究の一部は 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。

参考文献

- [1] 梅田宏明, 埴敏博, 庄司光男, 朴泰祐, 稲富雄一, 情報処理学会論文誌コンピューティングシステム(ACS), **6**, 26(2013).
- [2] K. Kitaura et al., Chem. Phys. Lett., **312**, 319(1999).
- [3] OpenFMO; <http://www.openfmo.org/>
- [4] 梅田宏明, 埴敏博, 庄司光男, 朴泰祐, 重田育照, J. Comp. Chem. Japan, **13**, 323(2015).
- [5] HA-PACS ベースクラスター;
http://www.ccs.tsukuba.ac.jp/research/research_promotion/project/ha-pacs/cluster