4E01

GPUを用いた分割統治型電子状態計算の高速化

（早大先進理工[1], 早大理工研[2], JST-CREST[3], 京大ESICB[4]） 吉川 武司[1], 中井 浩巳[1-4]

Acceleration of divide-and-conquer calculation on GPU

(Advanced Science and Engineering, Waseda Univ.[1], RISE, Waseda Univ.[2], CREST, JST Agency[3], ESICB Kyoto Univ.[4])

Takeshi Yoshikawa[1], Hiromi Nakai[1-4]

【緒言】 近年 GPGPU の量子化学への適用は広範囲に行われている。Hartree-Fock (HF)から高精度な Cupled Cluster (CC)計算にまで幅広く対応しており、大幅な高速化に成功してきた。その中でも、GPGPU 専用の量子化学プログラムパッケージ『TeraChem』[1]は全てのコードを CUDA で書かれており、タンパク質等の大規模分子に対しても、HF 計算を数十倍から数百倍の高速化が可能である。しかし、大規模分子に対する高精度な電子相関理論の適用例は少ない。その大きなボトルネックとなっているのが GPU 上のメモリ量である。CPU のメモリに対して GPU 内部のメモリは最大で 5〜6 GB と比較的低量である。そのため、大規模分子に対して適用するためには、使用するメモリ量を少なくする必要がある。その解決策として当研究室で開発を行ってきた分割型計算理論の一つである分割統治(DC)法[2,3]の利用を提案する。DC 法等の分割型計算理論の特徴として、必要なメモリサイズを分割することで大幅なメモリ削減ができ、大規模分子への適用が可能となる。本研究では、GPGPU を用いた大規模高精度電子状態理論を可能とするために、DC 法のコードを GPU 化し、高速化を行ったので報告する。

【RI 近似を用いた DC-HF 法の高速化】 DC-HF 法は全系をいくつかの部分系に分けて対角化計算を行うことで計算コストを削減する方法である。部分系の周りのバッファ領域を含めた局在化領域で部分系αの軌道を構築することで、周囲の効果を取り込むことが可能となる。しかし、SCF サイクルにおいて、DC 法は全系の Fock 行列を生成する必要があり、その生成時間が DC-HF 法の大きなボトルネックとなっている。ただし、対角化とは異なり、Fock 行列の生成は並列化効率が高く、GPU を用いた高速化が可能である。さらに、resolution-of-identity (RI)近似[4]を適用することにより Fock 行列生成のコストを下げることも可能である。

DC-RI-HF法ではクーロン項**J**、交換項**K**をRI近似より(1), (2)式を用いて求めることができる。

$$J_{\mu\nu}^{\text{DC-RI-HF}} = \sum_{\alpha} \sum_{\lambda\sigma\in S(\alpha)} \sum_{PQ} (\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma) D_{\lambda\sigma}^{\alpha} \quad (1)$$

$$K_{\mu\nu}^{\text{DC-RI-HF}} = \frac{1}{2} \sum_{\alpha} \sum_{\lambda\sigma\in S(\alpha)} \sum_{PQR} (\mu\sigma|P)(P|R)^{-1/2}(R|Q)^{-1/2}(Q|\mu\sigma) D_{\lambda\sigma}^{\alpha} \quad (2)$$

ここで、$\mu, \nu, \lambda, \sigma$は原子基底を、$P, Q$ は補助基底関数を表す。ここで、**J**、**K** を求める部分を GPU で並列に計算を行った。

【数値検証: DC-RI-HF 法】 β-ストランド型オリゴグリシン$(gly)_{10}$ に対して、基底関数は cc-pVTZ、補助基底関数は cc-pVTZ を用いて DC-RI-HF 計算を行った。DC 法による計算では、部分系はグリシンモノマーからなるユニットとし、バッファは左右 2 ユニットとした。表 1 に、direct-SCF 計算と RI-SCF 計算における Fock 行列の構築と対角化の時間を示す。ただし、各項目の高速化率を括弧書きで示す。GPU を用いることによって Fock 行列の構築時間を 4.7

倍程度減少することが可能となる。さらに、RI 近似と GPU を融合させることによって、Fock 行列生成時間を 10.6 倍高速化することに成功した。一方で、並列化が困難である対角化の計算時間は DC 法を用いることによって高速化することが可能である。今回は Fock 行列生成の部分のみ GPU 化したため対角化の計算時間は GPU を用いた場合でも変化はないが、全てを GPU 化することによって対角化の計算時間も減少することが可能となる。

Table 1. The wall time [min] of construction of Fock matrix and diagonalization of SCF equation by direct-SCF and RI-SCF on CPU and GPU.

	Construction		Diagonalization		Total time	
Direct-HF(CPU)	124.2	-	3.8	-	128.5	-
Direct-HF(GPU)	26.7	(4.7)	3.8	(1.0)	30.7	(4.2)
RI-HF(GPU)	11.7	(10.6)	3.8	(1.0)	15.9	(8.1)
DC-RI-HF(GPU)	11.7	(10.6)	2.5	(1.5)	14.7	(8.8)

CPU: intel Xeon 3.47GHz (1core) GPU: Tesla K20m

【GPU を用いた DC-電子相関法の高速化】 DC 法に基づく 2 次 Møller-Plesset (MP2)法や CC 法等においては、各部分系の軌道を用いることで電子相関計算を行う。部分系の軌道はバッファ領域にも広がっているため、当研究室で開発されたエネルギー密度解析(EDA)[5]を用いて中央領域だけの相関エネルギーを求めることで、2 重カウントを防いでいる。これを MP2、CCSD 計算に適用すると、以下のようにして相関エネルギーを見積ることができる。

$$\Delta E_{\mathrm{corr.}}^{\mathrm{DC}} = \sum_{\alpha} \sum_{i,j}^{occ} \sum_{a,b}^{vir} \sum_{\mu \in S(\alpha)} C_{\mu i}^{\alpha *} \left\langle \mu j^{\alpha} \middle| a^{\alpha} b^{\alpha} \right\rangle \left[2\tilde{t}_{ij,ab}^{\alpha} - \tilde{t}_{ij,ba}^{\alpha} \right] \qquad (3)$$

MP2 の場合: $\tilde{t}_{ij,ab}^{\alpha} = -\dfrac{\left\langle a^{\alpha} b^{\alpha} \middle| i^{\alpha} j^{\alpha} \right\rangle}{\varepsilon_a^{\alpha} + \varepsilon_b^{\alpha} - \varepsilon_i^{\alpha} - \varepsilon_j^{\alpha}}$

CCSD の場合: $\tilde{t}_{ij,ab}^{\alpha} = t_{i,a}^{\alpha} t_{j,b}^{\alpha} - t_{i,b}^{\alpha} t_{j,a}^{\alpha} + t_{ij,ab}^{\alpha}$

ここで、ボトルネックとなっている MO 積分、T_1、T_2amplitude を求める部分を GPU で並列に処理を行うことによって高速化を目指す。

【数値検証: DC-CC 法】オリゴエン $C_{2n}H_{2n+2}$ に対して、基底関数は 6-311G**を用いて DC-CCSD 計算を行った。DC 法による計算では、部分系は炭素二個からなるユニット(C_2H_4 or C_2H_5)とし、バッファは左右 2 ユニットとした。図 2 に、通常法と DC 法を用いた場合の CCSD 計算に必要な時間と分子サイズとの関係を示す。ただし、CPU は intel Xeon 3.47GHz (1core)を、GPU は Tesla K20m を用いて計算を行っている。DC 法を用いることによってほぼ線形の計算時間を達成することに成功しているが、CPU を用いた場合のプレファクターは依然として大きい。しかし、GPU を用いることによって計算時間の大幅な削減に成功し、15.6 倍の高速化に成功した。

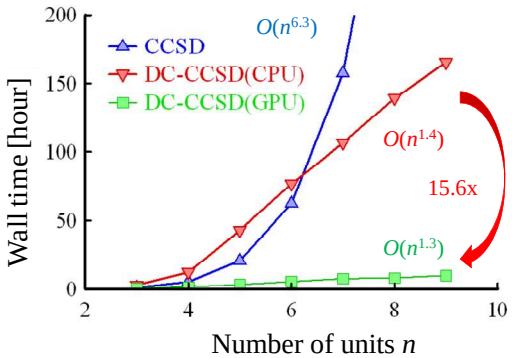

Fig. 1. System-size dependence of the wall times for the DC and conventional CCSD calculations of $C_{2n}H_{2n+2}$ on CPU and GPU.

[1] I. S. Ufimtsev and T. J. Martinez, *J. Chem. Theor. Comput.* **4**, 222 (2008).
[2] W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).
[3] M. Kobayashi and H. Nakai, in Linear-Scaling Techniques in Computational Chemistry and Physics: Methods and Applications (2011, Springer), pp. 97-127.
[4] J. L. Whitten, *J. Chem. Phys.* **58** (1973) 4496. [5] H. Nakai, *Chem. Phys. Lett.* **73** (2002) 363.