

# MPI/OpenMP ハイブリッド並列化大規模量子化学計算アルゴリズムの開発

石村和也<sup>1</sup>、倉本圭<sup>2</sup>、生田靖弘<sup>1</sup>、兵頭志明<sup>1</sup>

<sup>1</sup>豊田中央研究所、<sup>2</sup>兵庫県立大学

e1502@mosk.tytlabs.co.jp

**概要：**量子化学計算は、化学反応の予測・制御、スペクトルの解析など化学の様々な面で重要な役割を果たし、取り扱われる分子サイズは年々大きくなっている。そこで、量子化学の基礎理論である Hartree-Fock 法について、計算機ノード間を MPI、計算機ノード内を OpenMP で並列化する MPI/OpenMP ハイブリッド並列化アルゴリズムを開発した。最も計算時間のかかる Fock 行列計算だけでなく、初期軌道計算など計算全体の並列化を行った。2048CPU コアまでの計算時間、並列加速率を報告する。

## 1 はじめに

物質が変化する化学反応の研究において、量子化学計算は反応の解析や制御、化合物の安定性や物性など様々な面で用いられており、計算に対する要求は高まる一方である。計算対象分子は年々大きくなり、大きな置換基を付けることによる立体効果や、分散力など非共有結合を多用することで、新たな物質・反応の設計が期待される。通常用いられる計算方法は、原子数のおおよそ 3 乗に比例して計算量が増加することから、ナノサイズ分子では計算コストは膨大となる。今後の計算機システムで高速に計算するためには、並列化効率の良いプログラムの開発が求められている。本研究では、量子化学の基礎理論である Hartree-Fock 法について、計算機ノード間を MPI、計算機ノード内を OpenMP で並列化した MPI/OpenMP ハイブリッド並列化アルゴリズムを開発・実装し、2048CPU コアまでのベンチマーク計算を行った。

## 2 並列化アルゴリズム

Hartree-Fock 計算で最も時間を要する Fock 行列(2電子反発積分)計算は4重ループの中で行われる。図1のように、第1ループを OpenMP で、第3ループを MPI のランクにより負荷を分散させるアルゴリズムを開発した。OpenMP での分散を最外ループで行うことにより、OpenMP のオーバーヘッドを大幅に減らし、さらに動的に分散させることにより、ノード内の負荷をほぼ均等に分散させることができる。

Fock 行列計算以外にも、1電子積分、初期軌道計算のハイブリッド並列化を行った。初期軌道の射影計算では、大半が行列積演算になる次式[1]を

用いることにより、高速化も行った。

$$\mathbf{C}_1 = \mathbf{S}_{11}^{-1} \mathbf{S}_{12} \mathbf{C}_2 \left[ \mathbf{C}_2' \mathbf{S}_{12}' \mathbf{S}_{11}^{-1} \mathbf{S}_{12} \mathbf{C}_2 \right]^{1/2}$$

$\mathbf{C}_1$ : 初期軌道係数

$\mathbf{C}_2$ : 拡張 Huckel 計算による軌道係数

$\mathbf{S}_{11}$ : 実計算の基底の重なり積分

$\mathbf{S}_{12}$ : 拡張 Huckel と実計算との基底の重なり積分

```
$OMP parallel do schedule(dynamic)
do  $\mu = 1, nbasis$ 
  do  $v = 1, \mu$ 
    do  $\lambda = 1, \mu$ 
       $\mu\nu\lambda = \mu\nu\lambda + 1$ 
      if(mod( $\mu\nu\lambda, nproc$ ).ne.mpi_rank) cycle
      do  $\sigma = 1, \lambda$ 
        2 電子反発積分( $\mu\nu\lambda\sigma$ )計算
        ( $\mu\nu\lambda\sigma$ )を Fock 行列に足し込み
      enddo
    enddo
  enddo
enddo
call mpi_allreduce(Fock matrix)
```

図1 Fock 行列計算の MPI/OpenMP ハイブリッド並列化アルゴリズム

## 3 計算結果

開発したアルゴリズムを量子化学計算プログラムパッケージ GAMESS[2]に実装した。オリジナルの GAMESS は MPI(もしくは Socket 通信)のみで並列化されている。TiO<sub>2</sub> クラスタ (Ti<sub>35</sub>O<sub>70</sub>, 6-31G 基底, 1645 次元, 30SCF サイクル) を用いて、Cray XT5 (Quad-core Opteron 2.4GHz, 8CPU コア/ノード) 2048CPU コアのベンチマーク計算を行った。ハイブリッド並列計算は、8 スレッド/プロセスで行った。

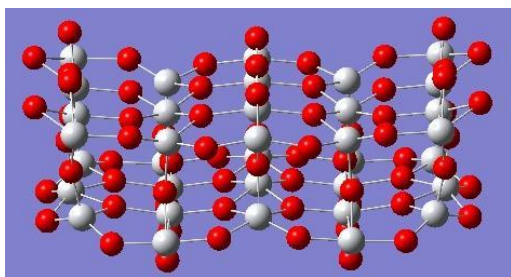


図 2 TiO<sub>2</sub> クラスタ

TiO<sub>2</sub> クラスタ全計算の並列加速率を図 3 に示す。2048CPU コアの並列加速率は、オリジナルの GAMESS では 748 倍であったが、初期軌道計算変更により 1055 倍(改良版(Flat MPI))に向上し、さらに MPI/OpenMP ハイブリッド並列化により 1238 倍(改良版(MPI/OpenMP))に向上した。

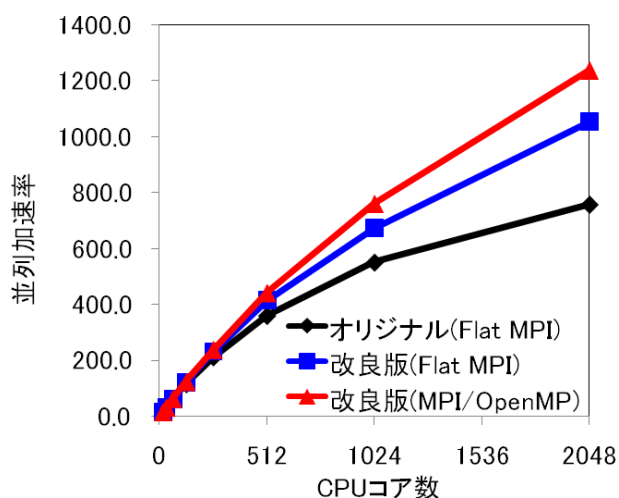


図 3 TiO<sub>2</sub> クラスタ全計算の並列加速率

表 1 TiO<sub>2</sub> クラスタ全計算時間(秒)及び並列加速率(カッコ内)

CPU コア数	16	256	1024	2048
オリジナル	18176.4	1368.6	527.6	383.5
Flat MPI	(16.0)	(212.5)	(551.2)	(758.3)
改良版	18045.6	1241.2	428.7	273.7
Flat MPI	(16.0)	(232.6)	(673.5)	(1054.9)
改良版	18121.6	1214.6	381.1	234.2
MPI/OpenMP	(16.0)	(238.7)	(760.8)	(1238.0)

MPI/OpenMP ハイブリッド並列化により、Fock 行列計算の並列加速率はオリジナルに比べ向上しており、CPU コア数が多くなるほど、その差は大きくなっている(表 2)。

オリジナル版の初期軌道計算の割合は、16CPU コアでは 0.9%であるのに対し、2048CPU コアでは 37.5%になっている(表 3)。これが、図 3 におけるオリジナル(Flat MPI)と改良版(Flat MPI)の差

の主な原因である。改良版では、2048CPU コアでも 5.9 – 7.0%と割合は小さく、計算全体の並列加速率への影響は小さい。シリアル計算ではわずかな割合(1%以下)の計算でも、1000CPU コア規模の計算では無視することはできない。これからの大規模計算では、主要な計算部分のみの並列化だけではなく、計算全体の高速化・並列化が重要である。

表 2 TiO<sub>2</sub> クラスタの Fock 行列計算時間(秒)及び並列加速率(カッコ内)

CPU コア数	16	256	1024	2048
オリジナル	17881.8	1175.2	334.0	188.6
Flat MPI	(16.0)	(243.5)	(856.6)	(1517.0)
改良版	17953.5	1175.2	360.0	203.1
Flat MPI	(16.0)	(244.4)	(797.9)	(1414.4)
改良版	17777.6	1150.4	316.4	174.8
MPI/OpenMP	(16.0)	(247.3)	(899.0)	(1627.2)

表 3 TiO<sub>2</sub> クラスタの初期軌道計算時間(秒)及び全計算時間に占める割合(カッコ内)

CPU コア数	16	256	1024	2048
オリジナル	166.2	143.6	143.6	143.8
Flat MPI	(0.9%)	(10.5%)	(27.2%)	(37.5%)
改良版	20.2	18.6	18.9	19.2
Flat MPI	(0.1%)	(1.5%)	(4.4%)	(7.0%)
改良版	18.6	13.2	13.6	13.8
MPI/OpenMP	(0.1%)	(1.1%)	(3.6%)	(5.9%)

## 4 まとめ

Hartree-Fock 計算全体の MPI/OpenMP ハイブリッド並列化アルゴリズムを開発し、2048CPU コアまでのベンチマーク計算を行った。初期軌道計算改良及びハイブリッド並列化により、計算時間、並列加速率ともにオリジナルの GAMESS よりも大幅に向上した。本研究を基にして、高精度計算への展開が可能になった。

## 参考文献

- [1] D. Cremer, J. Gauss, *J. Comput. Chem.* **7**, 274 (1986).
- [2] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, J. A. Montgomery, *J. Comput. Chem.* **14**, 1347 (1993).