

# FMOプログラムABINIT-MPのGPU化への対応

Customization of GPU Performance Tuning for ABINIT-MP

坂倉耕太<sup>1)</sup>, 望月祐志<sup>2)3)</sup>, 中野達也<sup>4)</sup>, 成瀬彰<sup>5)</sup>,

大島聰史<sup>6)</sup>, 星野哲也<sup>7)</sup>, 片桐考洋<sup>7)</sup>

Kota Sakakura, Yuji Mochizuki, Tatsuya Nakano, Akira Naruse,

Satoshi Oshima, Tetsuya Hoshino, Takahiro Katagiri

1) 計算科学振興財団 (〒650-0047 神戸市中央区港島南町7-1-28)

2)立教大学理学部 (〒171-8501 東京都豊島区西池袋3-34-1)

3)東京大学生産技術研究所 (〒153-8505 東京都目黒区駒場4-6-1)

4) 高度情報科学技術研究機構神戸センター (〒650-0047 神戸市中央区港島南町1-5-2)

5)エヌビディア合同会社 (〒107-0052 東京都港区赤坂2-11-7ATT新館13F)

6)九州大学情報基盤研究開発センター (〒819-0395 福岡市西区元岡744)

7)名古屋大学情報基盤センター (〒464-8601 名古屋市千種区不老町)

We have been developing the ABINIT-MP program for fragment molecular orbital(FMO) calculations. In this paper, we report on the tuning method of the ABINIT-MP on GPU. As a result, the computational performance of the code with two electron repulsion integrals GPU tuning improved by up to 65.7 times higher than the original one.

**Key Words :** Fragment molecular orbital, FMO, ABINIT-MP, GPU, Two electron repulsion integrals

## 1. はじめに

フラグメント分子軌道(FMO)法[1][2]は、タンパク質や核酸などの生体分子をフラグメントに分割し、各フラグメントに対する小規模電子状態計算することで、巨大分子全体の電子状態を近似的に高速に解く手法である。計算結果は、対象系の詳細な相互作用解析に好適であり、理論創薬や生物物理学の分野で広く利用されている。ABINIT-MPは、「富岳」や「不老」といった富士通のA64FXスパコンや、ベクトル機であるNEC SX-Aurora TSUBASA(SX-AT)環境など様々なアキテクチャに対応している。

現在、ネクスト「富岳」のフィジビリティスタディが進められており、次世代機にはGPUの搭載が想定されている[3]。ABINIT-MPは、ネクスト「富岳」を頂点とする次世代のHPCI資源を活用したFMO計算のための準備をこれからも進めていく必要があり、OpenACCを用いたGPU化に着手した。

## 2. GPU化にむけた設計、ミニアプリによる試行

ABINIT-MPの計算時間で最もコストが高いのは2電子積分計算部分であり、計算機アキテクチャによって異なるが、そのコストはおよそ50%以上になる(図1)。ABINIT-MPの2電子積分計算は、Obaraのアルゴリズム[4]に従い、{s,p,d}関数の組み合わせ $3^4 = 81$ 種類、具体的には、

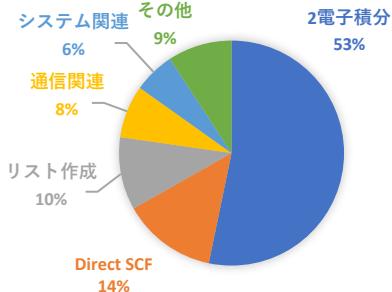


図1 ABINIT-MP のコスト分布

(ss|ss)～(dd|dd)の計81種類のサブルーチンによって構成されている。また、各2電子積分ルーチンは展開項数に起因する4重ループ構造になっており、最内ループ長は、例えば6-31G基底であれば、最大で6、展開項ペアループとして考慮しても高々36程度である。GPUで高効率に処理するためには、数千並列以上のループ長が望ましく、そのためには、現状の処理ループ構造を大きく変更する必要がある。

GPU化の第一歩として、2022年度下期の東京大学情報基盤センターの支援プログラムにおいて、ABINIT-MPのOpenACCによるGPU化を実施した。作業の効率化を考慮し、ABINIT-MPのHF計算のコア部分を切り出して作成したミニアプリケーション(以後、ミニアプリと略記)をその対象とした。まず、冗長であった81種類の2電子積分ルー

チングを対象性や類似処理をまとめる形で、26種類のルーチンに再編成し、GPUの豊富な演算コアをバッチ的に満たせるように改変した。その結果、東京大学情報基盤センターのWisteria Aquariusサブシステムを使って8プロセス(8GPU)実行で、FMOダイマー計算を想定したテストデータにおいて、6-31G\*基底で6.4倍、s関数の縮約の長いcc-pVDZ基底では8.2倍の加速が得られており[5]、GPU化による有用性を確認した。

### 3. ABINIT-MP本体へのGPU化の実装

#### (1) ループ構造の改変

GPU評価ミニアプリ版で試行した最適化施策を参考に、ABINIT-MPの最新の公開版本体(Ver. 2 Rev. 8)[6]のGPU化を行った。現状のループ構造は、図2で示すように自己無撞着電荷(SCC)反復ループ内で各フラグメントに相当するモノマー単位でそれぞれSCF反復計算が行われ、軌道シェルペアIJ、KLループ内で、軌道角運動量タイプ別に処理を割り振り、それぞれの各計算ルーチンにて2電子積分が計算されている。

```
[MONOMER PART]
do SCC-loop
  do monomer-loop
    <parallel MPI>
    do scf-loop
      do IJ
        <parallel OpenMP>
        do KL
          select case(ang_type)
            case (ss|ss)
            case (ps|ss)
            case (ps|ps)
            case (pp|ss)
            case (pp|ps)
            case (pp|pp)
            do rs
              do tu
                calc ERI
          enddo
        enddo
      enddo
    enddo
  enddo
```

図2 ループ構造イメージ（修正前）

並列処理に関しては、モノマーループ、IJ軌道シェルペアループにてそれぞれ、MPI、OpenMP化されている。

GPU化に際し、図3のように、SCC、モノマーループ内で、各軌道角運動量タイプ別の2電子積分計算をまとめ、それぞれの処理内でIJ、KL軌道シェルペアループを1重ループ化し、GPU対象ループとして整備した。また、2電子積分ルーチン群についてはミニアプリで81種類から26種類に削減したが、さらに冗長性を排除し、21種類に再編した。まずは、モノマー部分、s,p軌道に関する2電子積分タイプの整備を優先して開発、実装した。

#### (2) OpenACC指示子によるGPU化

各軌道角運動量タイプ別の2電子積分の軌道シェルペアIJ、KLの4重ループを1重化したループをGPU対象ループとし、OpenACC指示子にてGPU化した。また、CPU/GPU

```
[MONOMER PART]
do SCC-loop
  do monomer-loop
    set index table for GPU
    sort angular momentum type integrals
  do scf-loop
    do n_ss_ss !on GPU
    do n_ps_ss !on GPU
    do n_ps_ps !on GPU
    do n_pp_ss !on GPU
    do n_pp_ps !on GPU
    do n_pp_pp !on GPU
    <parallel MPI>
      ij = list()
      kl = list()
      do rs
        do tu
          calc ERI
      enddo
    enddo
  enddo
enddo
```

図3 ループ構造イメージ（修正後）

間のデータ転送を最小にするため、2電子積分計算に使用される原子座標、電子密度、補助積分データをはじめ、予め計算しておいた重なり積分値等の各変数や変換インデックステーブルの配列をOpenACCのDataコンストラクトを使用して、各モノマーのSCF収束計算前のタイミングでデータ転送した(図4)。

```
!$acc data copyin(x,y,z,dc,pps, nsh2exp) copy(fc)
do ia = 0,num_angtype-1
```

```
!$acc parallel num_workers(2) vector_length(16)
!$acc loop gang worker
do n4 = 1, nsize
  ish = nslist(1,n4)
  jsh = nslist(2,n4)
  ksh = nslist(3,n4)
  lsh = nslist(4,n4)
```

図4 OpenACC指示子によるGPU最適化イメージ  
上) データ転送 下) 並列処理

### 4. 性能測定

計算の実行環境には、ミニアプリの評価と同様に、Wisteria Aquariusを用いた。使用したGPUはNVIDIA社のA100である。テストデータとしてChignolin（残基数20）HF/6-31Gを採用した。GPU有無での1プロセス実行時のモノマー部分の2電子積分タイプ別に測定した。その結果、表1に示すように、24.3倍～65.7倍の加速が得られており、

表1 GPU有無における計算時間と加速

(Chignolin/HF/6-31G) Wisteria Aquarius

積分タイプ	CPU(s)	GPU+(s)	Acc.
(ss ss)	59.1	0.9	65.7
(ps ss)	148.0	3.3	44.9
(ps ps)	112.7	4.6	25.0
(pp ss)	58.3	2.4	24.3
(pp ps)	120.3	3.3	36.5
(pp pp)	42.9	1.2	36.0

大幅な性能加速を確認できた。特に、s関数はcc-pVDZなどの基底では縮約長が伸びるために、より高い加速が期待できる。

本稿では1プロセス1GPU実行での報告になるが、複数GPUを活用するための開発整備を計画中である。特にダイマーの計算では、GPU対象ループ長はモノマー部分に対して4乗に比例するため、複数GPUを使用した性能加速が期待できる。

## 5. まとめと今後の予定

本稿では、ABINIT-MPのGPU化への取り組みについて報告した。最計算コスト部分である2電子積分計算部分のGPUを用いた加速において良好な結果を得ることができた。今後は、**1) ダイマー部分の2電子積分計算のGPU化、2) MPI並列化と複数GPU利用整備、3) MPI/OpenMP処理部分との非同期実行制御、4) 1電子積分計算等の高コスト処理部分のGPU化検討**、の順で、GPUを用いた高速化の整備を進めていく予定である。ダイマーの方がHF計算の規模が大きくなるので、GPU化のメリットが顕在化する期待があることは書き添えたい。

## 謝辞

HFミニアプリのGPU対応は、2022年度下期の東京大学情報基盤センターのGPU移行推進プログラムの中で、エヌビディア合同会社の古家真之介氏のコーディネートで成されたものです。また、同センターの中島研吾先生、下川辺隆史先生、芝隼人先生(当時)のサポートにも感謝します。最後に、本研究開発は立教SFRの資金援助を受けていることを記します。

## 参考文献

- [1] Fedorov, K, Kitaura ed., "The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems", 2009, CRC.
- [2] Y. Mochizuki, S. Tanaka, K. Fukuzawa, ed., "Recent Advances of the Fragment Molecular Orbital Method", 2021, Springer.
- [3] <<https://cs-forum.github.io/roadmap-2023/>>.
- [4] S. Obara, A. Saika, *J. Chem. Phys.*, 84, 3963 (1986)  
DOI:10.1063/1.450106
- [5] Y. Mochizuki, T. Nakano, K. Sakakura, K. Okuwaki, et al., *J. Comput. Chem. Jpn.*, 23, 4(2024)  
DOI: 10.2477/jccj.2024-0001
- [6] <[http://www.cenav.org/abinit-mp-open\\_ver-2-rev-8/](http://www.cenav.org/abinit-mp-open_ver-2-rev-8/)>.