

# フラグメント分子軌道法プログラム ABINIT-MPのGPUによる高速化 Acceleration of fragment molecular orbital method program ABINIT-MP using GPU

高度情報科学技術研究機構  
中野 達也

大阪大学D3センター  
坂倉 耕太

エヌビディア合同会社  
成瀬 彰

名古屋大学情報基盤センター  
片桐 孝洋、星野 哲也

九州大学情報基盤研究開発センター  
大島 聡史

東京大学情報基盤センター  
中島 研吾

鹿児島大学  
石川 岳志

立教大学理学部  
東京大学生産技術研究所  
望月 祐志

キーワード：GPU、Miyabi、フラグメント分子軌道法、FMO、ABINIT-MP

フラグメント分子軌道 (FMO) 法プログラム ABINIT-MP は、小規模サーバから「富岳」のようなスパコンまでに対応した大規模電子状態計算プログラムである。ポスト「富岳」(「富岳」NEXT) のような次世代スパコンでは、CPUの他にGPUのような演算加速器を搭載した計算機が主流になると考えられている。このため ABINIT-MP を GPU に対応させることが急務となっている。高度情報科学技術研究機構 (RIST) の利用支援部においても利用支援の観点から、国プロアプリのGPU対応に取り組んでいる。本解説ではこれまでに行った ABINIT-MP から Hartree-Fock 計算部分を切り出したミニアプリケーションを作成し、計算時間の大半を占め

る二電子積分計算とFock行列生成のGPU化を行い、東京大学情報基盤センターのWisteria-A (Aquarius) を用いてベンチマーク計算を行った結果について報告する。今回はそこから得られた知見を元に、OpenACCを用いてFMO法全体のGPU対応を行い、最先端共同HPC基盤施設(JCAHPC)のMiyabi-Gを用いてベンチマーク計算を行ったので、その結果についても報告する。

## 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電子状態を計算することで、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。FMO法の大きな特徴として、対象系の詳細な相互作用解析が可能である点にあり、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 A64FX スパコンや、ベクトル機であるNEC SX-Aurora TSUBASA (SX-AT) 環境など様々なアーキテクチャに対応している。

現在、「富岳」NEXT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のような演算加速器の搭載が想定されている。ABINIT-MPは整備・改良を進め、「富岳」NEXT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進めていく必要がある。そこで、本解説ではOpenACC [3] を用いたABINIT-MPのGPU対応の現状について報告したい。

## 2. ミニアプリケーションによる Hartree-Fock (SCF) 計算のGPU対応

ABINIT-MPの計算時間で最もコストが高いのは2電子積分計算及びFock行列生成部分であり、計算機アーキテクチャによって異なるが、そのコストはおおよそ60%以上になる(図1) [4]。

ABINIT-MPの2電子積分計算は、Obaraのアルゴリズム [5] に従い、{s,p,d} 関数の組み合わせ $3^4=81$ 種類、具体的には、(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

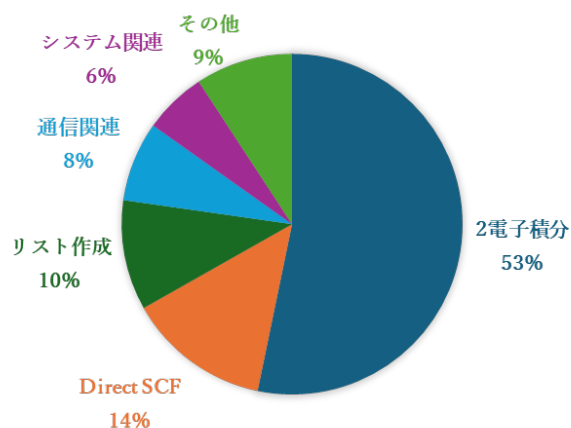


図1 ABINIT-MPのコスト分布 (Ala9Gly 分子。計算レベルはMP2/6-31G\*)

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 (Self-Consistent-Field; SCF) 計算 [6] 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電子積分ルーチンを対称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[4, 7-15]、GPU化による有用性が確認された。

### 3. ABINIT-MPのHF (SCF) 計算のGPU対応

ミニアプリの成果を踏まえ、ABINIT-MPのHF計算部分（2電子積分の計算及びFock行列の生成）のGPU対応を試みた。2電子積分についてはミニアプリのs, p積分のコードを活用した。

OpenACCでは、`!$acc parallel` 指示行と`!$acc end parallel` 指示行の間がGPU化の対象となる。`loop` 指示行で、`collapse(n)` とすると、コンパイラがn重ループを1重ループに変換してくれるので、ループ長を長くすることが出来る。`private` 節ではループ内で使用するローカル変数の指定を行う。`gang` 節、`worker` 節は階層化されたGPU内の実行単位を指定するもので、最外側の`loop` 指示行で指定すればよい。ここでは設定自体はコンパイラに任せている。また、総和計算部分は`atomic` 指示行で同期をとっている。

（後で述べるベンチマークにおいては、`gang`、`worker` 節の後に、さらに`seq` 節を追加して実施している。`seq` 節の有無で、コンパイル時のOpenACCのメッセージに変化はなく、アダマンタンの例にて確認したところ、計算時間にも差異はなかった。）図3、4の`vector` 節は更に細かい粒度で並列化を指示するものである。

現在の非経験的 (ab initio) 分子軌道法プログラムでは、6-31Gのようなcontracted basis setを用いる場合は、primitive Gaussianの線形結合 (contracted shell) からs軌道、p軌道のような原子軌道 (Atomic Orbital; AO) を構成する。GPU化を行う際には、ループ長を長くするために、AOではなくcontracted shellの4重ループを最も外側のループに採用することで、ループ長の確保を行った。ここで、`ixnccs` が分子を構成す

るcontracted shellの個数である。また`loop` 指示行に`collapse` 節を指定することで、4重ループを1重ループにまとめ、更にループ長を伸ばしている (図2)。Fock行列の生成に関しても、ミニアプリの成果を利用し、従来用いていたVeillardによる多重if文を用いたアルゴリズム [16] (二電子積分の添え字をif文で分類しFock行列を生成) を、[17] を参考にループでFock行列を計算するアルゴリズムに変更した。新しく作成した、2電子積分及びFock行列生成ルーチンの擬似コードを図2～図4に示した。

ベンチマーク計算には、最先端共同HPC基盤施設 (JCAHPC) [18] のMiyabi-Gを用い、アダマンタン分子 (図5) について、HF/STO-3G及びHF/6-31G計算を行った。Miyabi-Gは、米国NVIDIA社による超高速CPU-GPU専用リンクNVLink-C2Cで接続したGH200 Grace-Hopper Superchipを搭載した計算ノード1,120ノードから構成される国内初の汎用大規模システムである。(通常のジョブでは256ノードまで利用可能。) 共有メモリのコンパイルオプションには、`-gpu=mem:unified:nomanagedalloc` を指定し、CPUとGPUでメモリを共有する設定を有効にした。HF/6-31-Gを用いた場合、Miyabi-GでコンパイルしたABINIT-MP Ver.2 Rev.8を用いて、1CPU (1core) で計算した場合に対して、GPU対応を行ったABINIT-MPでは1CPU (1core)-1GPUを用い、26.4倍の計算時間の加速が得られた (表1)。また、NVIDIA Nsight Systems [19] によるプロファイリングを行ったところ (図6)、GPUの使用率に関して妥当な結果 (95%以上) が得られた。切れ込み状にGPUの稼働率が下がっている点については、GPU化されていない対角化ルーチンをコールしているためと思われる。また一部GPUではなくCPUの稼働率が他の部分よりも長く高くなっている点については、現在解析中である。

```

!$acc parallel
!$acc loop collapse(4) private(a,b,c,d,sint) gang worker
  do ics=1,ixnccs
    do jcs=1,ixnccs
      do kcs=1,ixnccs
        do lcs=1,ixnccs
select case
  case((ss|ss))
    ① (ss|ss)の計算
    ② Fock行列の計算
  case((ss|sp),(ss|ps),(sp|ss),(ps|ss))
...
end select
      end do
    end do
  end do
end do
!$acc end parallel

```

図2 2電子積分及びFock行列生成の擬似コード (ixnccsは分子を構成するcontracted shellの個数)

```

!$acc loop collapse(4) private(pq,...) vector
  do ipc = 1,ngi
    do iqj = 1,ngj
      do irc = 1,ngk
        do isc = 1,ngl

        初期積分 (Boysの関数) の計算
        (ss|ss)の計算
      end do
    end do
  end do
end do

```

図3 ①(ss|ss) 計算の擬似コード (ngi, ngj, ngk, nglはそれぞれ、i 番目、j 番目、k 番目、l 番目のcontracted shellを構成するprimitive Gaussianの個数。)

#### 4. FMO法のGPU対応

これらの結果を踏まえ、FMO法全体のGPU対応を行った。FMO法は、2体まで考慮したFMO2法 [20]、3体まで考慮したFMO3法 [21-23]、4体まで考慮したFMO4法 [24] までは実用化されている。ここでは、FMO2法について、GPU対応を行った。

FMO2法の基本式を式 (1) に示した。ここで  $E_{total}$  は系の全エネルギー、 $E_{IJ}$  はダイマーのエネルギー、 $E_I$  はモノマーのエネルギー、 $N_f$  はフラグメントの個数である。

$$E_{total} \approx \sum_{I>J} E_{IJ} - (N_f - 2) \sum_I E_I \quad (1)$$

```

!$acc loop collapse(4) vector
do ip = ips, ipe
do iq = iqs, iqe
do ir = irs, ire
do is = iss, ise
Fock行列の計算
...
!$acc atomic update
fock(iq, ip) = fock(iq, ip) + dc(is, ir) * val * 2. d0
!$acc end atomic
...
end do
end do
end do
end do

```

Atomic Orbitalのループ

図4 ② Fock生成ルーチンの擬似コード (ip, iq, ir, isはAOのインデックス。fockがFock行列。dcは電子密度行列。valは係数を掛けた2電子積分の値である。)

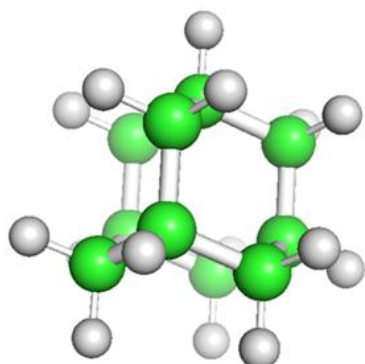


図5 アダマンタン分子の分子構造

表1 Miyabi-G (1CPU(1core)-1GPU) による計算時間の短縮

| 計算方法      | 1CPU (1core)*<br>(秒) | 1CPU (1core)-1GPU<br>(秒) | 倍率          |
|-----------|----------------------|--------------------------|-------------|
| HF/STO-3G | 0.6                  | 0.1                      | 6.0         |
| HF/6-31G  | 18.5                 | 0.7                      | <b>26.4</b> |

\*Miyabi-GでコンパイルしたABINIT-MP Ver.2 Rev.8による値

式 (1) を変形すると式 (2) が得られる。 $E_I'$  は周囲のフラグメントからの環境静電ポテンシャルエネルギーを除いたフラグメント  $I$  のエネルギー、

$$E_{total} \approx \sum E_I' + \sum \Delta \tilde{E}_{IJ} \quad (2)$$

式 (2) の第二項が、フラグメント間の相互作用エネルギー (Inter Fragment Interaction Energy; IFIE / Pair Interaction Energy; PIE) に相当する [25、26]。IFIE/PIEをエネルギー分割すると、Pair Interaction Decomposition Analysis (PIEDA) 法が得られる (式 (3))。

$$\Delta \tilde{E}_{IJ} = \Delta \tilde{E}_{IJ}^{ES} + \Delta \tilde{E}_{IJ}^{EX} + \Delta \tilde{E}_{IJ}^{CT+mix} \quad (3)$$

分割された3つのエネルギー成分は、それ

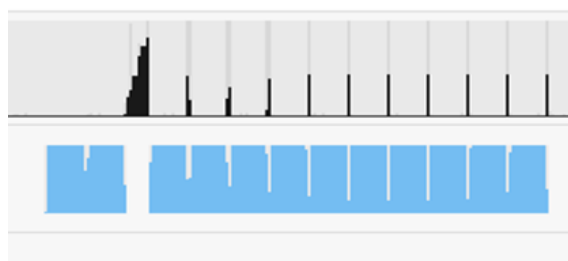


図6 NVIDIA Nsight Systemsによるアダマンタン HF/6-31G 計算の Miyabi-G における CPU 及び GPU の稼働率。上段の黒が CPU の稼働率、下段の水色のバーが GPU の稼働率を示す。

ぞれ静電項 (ES)、交換反発項 (EX)、電荷移動項CT+mix に相当している [26]。現在 FMO 計算結果の解析には PIEDA 法が広く用いられている。

FMO 法では  $N_f$  を系全体のフラグメント数とすると、電子状態を計算するモノマー SCF やダイマー SCF の演算量は  $O(N_f)$  に、モノマー及びダイマーに対する環境静電ポテンシャル並びに離れたモノマー間の相互作用を静電相互作用で近似する dimer-ES 近似の演算量は  $O(N_f^2)$  となる (図 7)。

モノマー SCF 及びダイマー SCF 計算のコードは、アダマンタンを計算した時のコードと同じになる。モノマー及びダイマーに対する環境静電ポテンシャルは、SCF 計算の 2 電子積分計算とほぼ同様であるが、使用できる積分の対称性は SCF 計算の場合よりも少なくなる。また、静電ポテンシャルを計算する部分は、2 電子積分にモノマーの電子密度行列を掛けるだけなので、Fock 行列生成ルーチンより簡単になる。Dimer-ES 近似

は、モノマー J からのモノマー環境静電ポテンシャルにモノマー I の電子密度行列を掛けただけなので、2 電子積分の計算ルーチンは、モノマー環境静電ポテンシャルの 2 電子積分計算ルーチンと共有化している。

次に GPU に対応した ABINIT-MP による FMO 計算のベンチマークを行った。ベンチマーク計算に使用した分子は、Gly5、Ala9Gly、Chignolin、Trp-Cage の 4 種類であり (図 8)、計算レベルは HF/6-31G とした。また、ベンチマーク計算には環境静電ポテンシャルに対する近似は使用しなかった。計算時間には、PIEDA 及び IFIE の計算時間も含まれている。表 2、表 3 に示したように、GPU に対応することで、FMO 計算についても 15 倍の以上の計算時間の加速が得られた。また、Trp-Cage について、FMO 計算全体の NVIDIA Nsight Systems によるプロファイリングを行った (図 9)。プロファイルの前半部はモノマー計算部分であり、後半がダイマー計算部分となる。GPU の稼働率は概

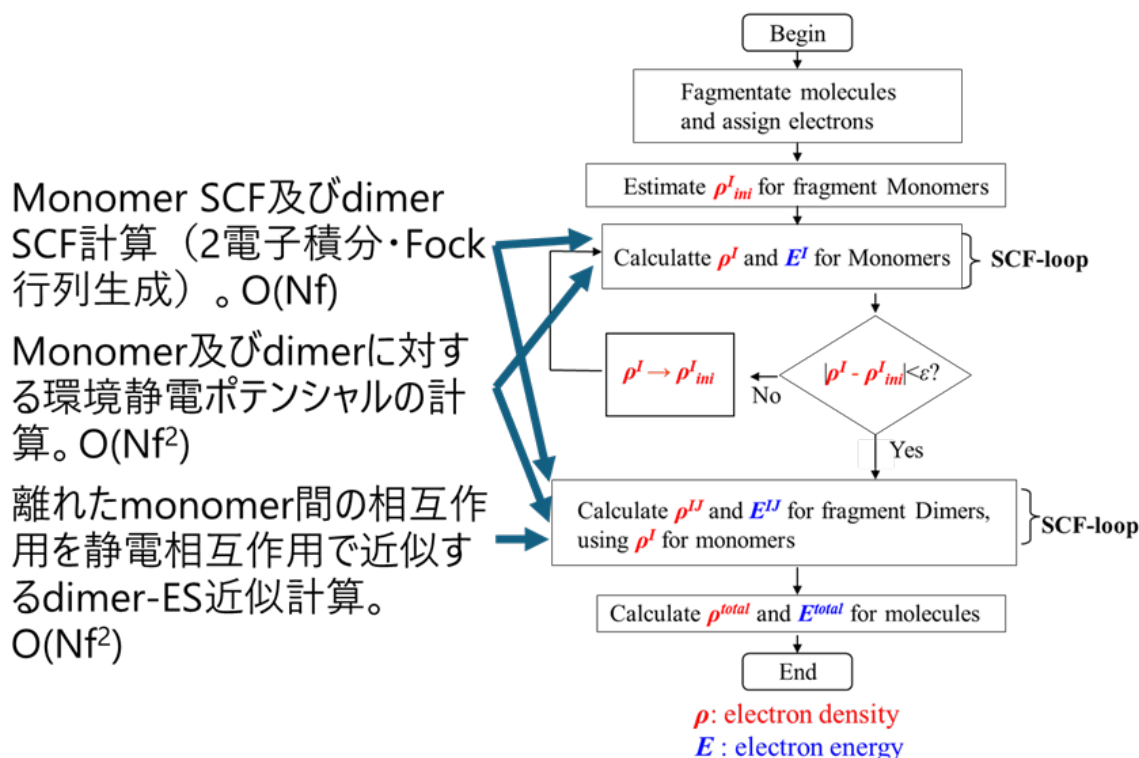


図7 GPU 対応した FMO 計算の主要ルーチン ( $N_f$  はフラグメントの個数)

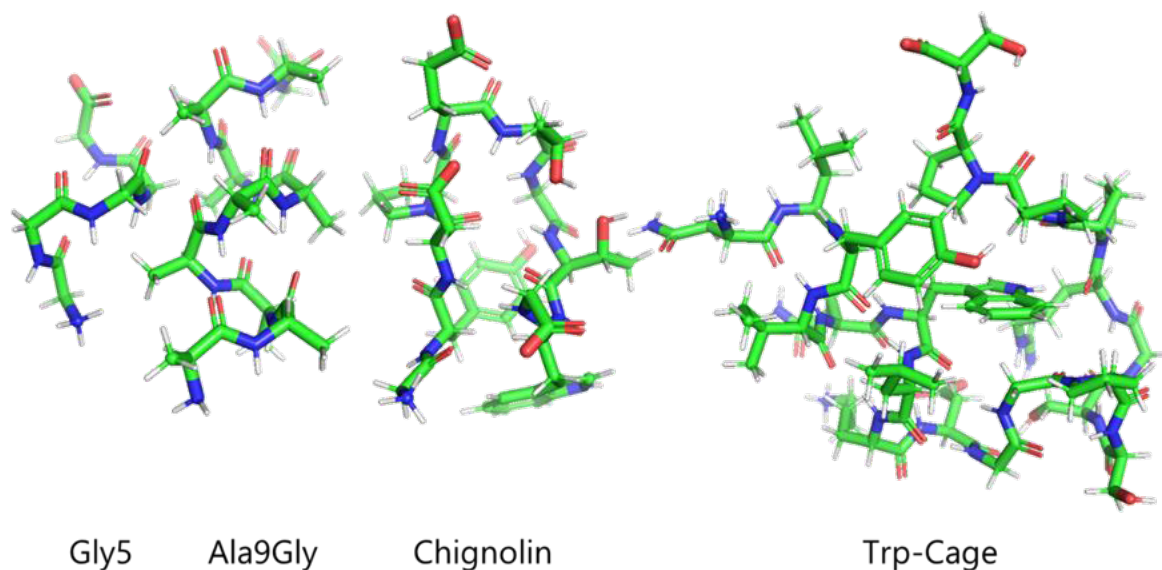


図8 FMO計算のテストに用いたGly5、Ala9Gly、Chignolin、Trp-Cageの分子構造

表2 Miyabi-Gによるペプチドの計算時間の短縮 (1CPU (1core)-1GPU)、FMO2-HF/6-31G)

| 計算分子    | 主要ルーチン      | 1CPU (1core)*<br>(秒) | 1CPU (1core)-1GPU<br>(秒) | 倍率   |
|---------|-------------|----------------------|--------------------------|------|
| Gly5    | Monomer SCC | 71.4                 | 2.7                      | 26.4 |
|         | Dimer-ES    | 0.0                  | 0.0                      | 1.0  |
|         | Dimer SCF   | 77.4                 | 3.6                      | 21.5 |
|         | Total       | 148.7                | 6.3                      | 23.6 |
| Ala9Gly | Monomer SCC | 412.9                | 15.7                     | 26.3 |
|         | Dimer-ES    | 3.0                  | 0.2                      | 15.0 |
|         | Dimer SCF   | 459.8                | 24.3                     | 18.9 |
|         | Total       | 856.0                | 40.0                     | 21.4 |

\*Miyabi-GでコンパイルしたABINIT-MP Ver.2 Rev.8による値

表3 Miyabi-Gによるタンパク質の計算時間の短縮 (1CPU (1core)-1GPU)、FMO2-HF/6-31G)

| 計算分子      | 主要ルーチン      | 1CPU (1core)*<br>(秒) | 1CPU (1core)-1GPU<br>(秒) | 倍率   |
|-----------|-------------|----------------------|--------------------------|------|
| Chignolin | Monomer SCC | 1762.0               | 81.0                     | 21.8 |
|           | Dimer-ES    | 4.5                  | 0.3                      | 15.0 |
|           | Dimer SCF   | 1845.5               | 127.2                    | 14.5 |
|           | Total       | 3607.5               | 208.1                    | 17.3 |
| Trp-Cage  | Monomer SCC | 8773.1               | 337.0                    | 26.0 |
|           | Dimer-ES    | 72.1                 | 4.7                      | 15.3 |
|           | Dimer SCF   | 6828.7               | 486.4                    | 14.0 |
|           | Total       | 15601.7              | 823.4                    | 18.9 |

\*Miyabi-GでコンパイルしたABINIT-MP Ver.2 Rev.8による値

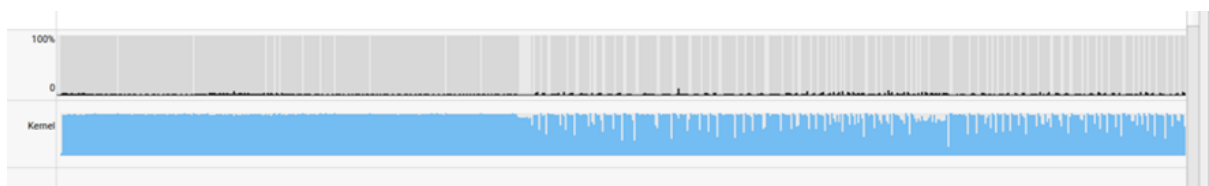


図9 NVIDIA Nsight SystemsによるTrp-Cage FMO2-HF/6-31G計算のMiyabi-GにおけるCPU及びGPUの稼働率。上段の黒がCPUの稼働率、下段の水色のバーがGPUの稼働率を示す。

ね95%以上であり妥当な結果と言える。ダイマー計算の方がGPUの稼働率が下がる場所が多いのは、ダイマーはモノマーの約2倍の大きさがあり、GPU化されていない固有値計算部分に時間がかかっているためと考えられる。

## 5. FMO計算のマルチノード・マルチGPUへの対応

次にFMO計算のマルチノード・マルチGPUへの対応を行った。実行スクリプトの主要部分を図10に示した。

Gly5、Ala9Gly、Chignolin、Trp-Cageの4分子について、ノード数（GPU数）=1、2、4、8、16における経過時間と、ABINIT-MP Ver.2 Rev.8 使用コア数（1、2、4、8、16）を変えた場合の経過時間の比較を行った。Ver.2 Rev.8はMPI/OpenMP並列化に対応しているため、OpenMPのみで並列化すると計算時間的には不利になるが、1コアと1GPUでの計算速度の比較を行うために、この条件で測定を行った。マルチノード・マルチGPUと1CPU（マルチコア）の計算時間を比較したグラフを、図11、12に示した。1CPU（1コア）と比較すると1ノード（1GPU）は非常に高速であり、Trp-Cageに

ついて、16GPUを用いると16コアで計算した場合の約130倍の高速化が得られた。また、図11を見ると8GPU辺りで、高速化の鈍化がみられるが、これはTrp-Cageが20残基と小さなテスト系であるため、実用計算で行われる数百から数千残基のタンパク質であれば、Miyabi-Gの256ノードを十分に使い切ることが出来ると考えている。

## 6. 今後の展開

今回行った小規模なベンチマーク計算からも、GPUのような演算加速器を搭載した計算機の優位性が示されたと考えている。今後は、Miyabi-Gの性能をフルに活用するために、マルチノード・マルチコア・マルチGPUへの対応が極めて重要と考えられる。このためにMiyabi-Gには、Multi-Process Service（MPS; 複数プロセスが同時に1つのGPUを効率的に使えるようにする仕組み）及びMulti-Instance GPU（MIG; 1つの物理GPUを、複数の独立した「仮想GPU」に分割できる技術）が用意されているが、MIGは1ノード内でしか利用できないため、MPSの利用が選択肢となる。MPSについては、東京大学情報基盤センターの塙教授よりご提供頂いた、MPSラッパースクリプトを

```
#PBS -l select=ノード数:mpiprocs=1
mpirun -n ノード数 ABINIT-MP MPI 版実行バイナリ
```

図10 マルチノード・マルチGPU実行スクリプトの主要部分

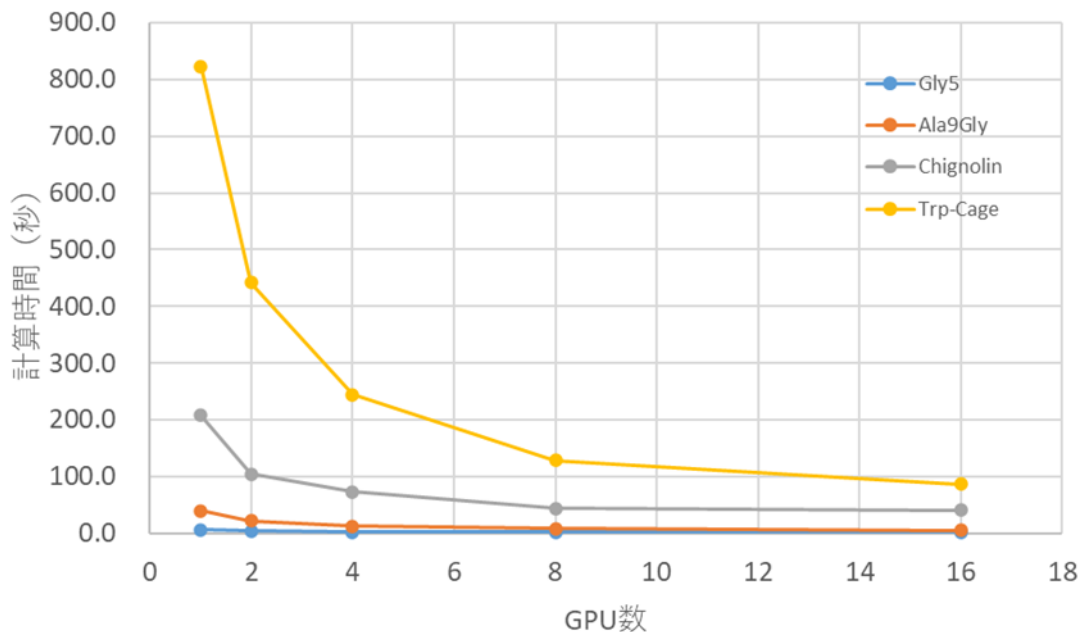


図11 マルチノードーマルチGPUの計算時間

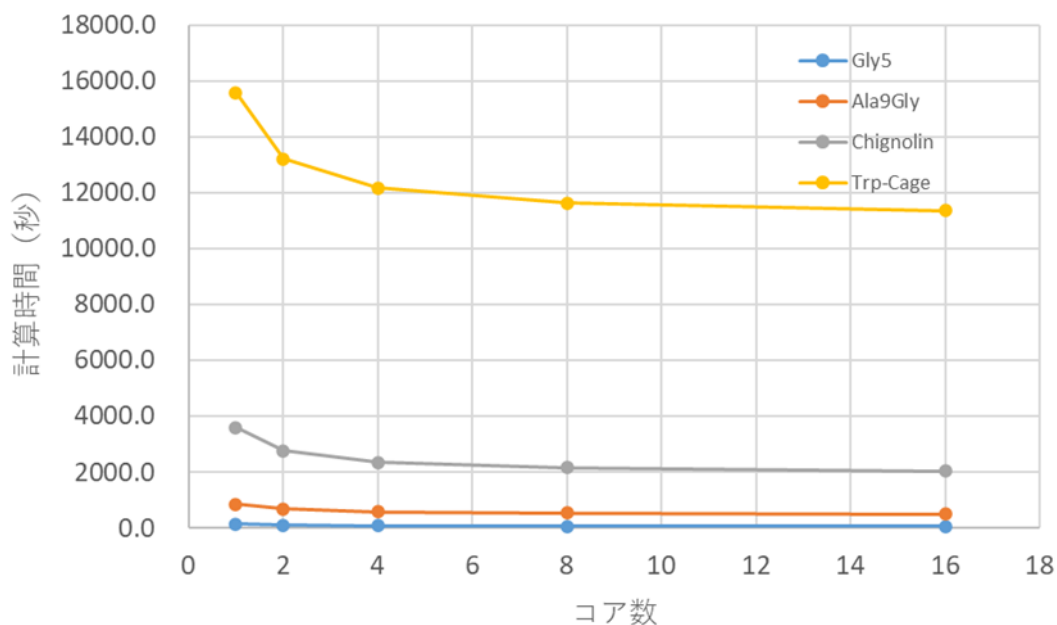


図12 1CPU (マルチコア) の計算時間

用いて検証計算を行う予定である。また、今回のベンチマーク計算では、6-31G基底関数を使用した。が、今後はcc-pVDZやcc-pVTZといったより高精度の基底関数が標準的に用いられると考えられる。そのためには、2電子積分計算の更なる高速化と高量子数 (d, f, g, h, ...) 積分への対応が必須であり、GPU

に対応したResolution of Identity (RI) 近似による2電子積分計算の導入を検討している [27-30]。最後に、HF計算では電子相関が含まれないためMP2計算のRI及びGPU対応を検討している [31-34]。

## 謝辞

HF ミニアプリのGPU対応は、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。本研究開発は立教SFR「FMOプログラムABINIT-MPの研究開発と整備」の資金援助を受けています。最後にMiyabi-Gの利用について、gz06「アプリケーションのGPUポータリング」、jh240001/jh250002「FMOプログラムABINIT-MPの次世代化」による支援を受けました。ここに記して感謝致します。

## 参考文献

- [1] D. 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://www.openacc.org/>>
- [4] 坂倉耕太、望月祐志、中野達也、成瀬彰、大島聡史、星野哲也、片桐孝洋、FMOプログラムABINIT-MPのGPU化への対応. 計算工学講演会論文集 Vol.29 (2024年6月)
- [5] S. Obara, A. Saika, Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. J. Chem. Phys. 84, 3963-3974 (1986)
- [6] A. Szabo, N. S. Ostlund 著、大野公男、阪井健男、望月祐志 訳、新しい量子化学 上、東京大学出版会、1987
- [7] 望月祐志、坂倉耕太、成瀬彰、古家真之介、下川辺隆史、芝隼人、大島聡史、星野哲也、片桐孝洋、HF計算に特化したABINIT-MPミニアプリのGPU化の試み. 理論化学討論会、横浜、2023/5/19.
- [8] 望月祐志、中野達也、坂倉耕太、奥脇弘次、土居英男、加藤季広、滝沢寛之、成瀬彰、大島聡史、星野哲也、片桐孝洋、MOプログラムABINIT-MPの整備状況2023. J. Comp. Chem. Jpn. 4-8, 23 (2024)
- [9] 坂倉耕太、中野達也、望月祐志、FMOプログラムABINIT-MPの高速化と大規模系対応. 第5回スーパーコンピュータ「不老」ユーザ会、名古屋、2024/10/2
- [10] 望月祐志、坂倉耕太、土居英男、奥脇弘次、加藤季広、滝沢寛之、成瀬彰、大島聡史、星野哲也、片桐孝洋、FMOプログラムABINIT-MPの現状と今後. 分子科学討論会2024、京都、2024/9/19
- [11] 坂倉耕太、望月祐志、中野達也、成瀬彰、大島聡史、星野哲也、片桐孝洋、FMOプログラムABINIT-MPのGPU化への対応. 第29回計算工学講演会、神戸、2024/6/12
- [12] 望月祐志、フラグメント分子軌道(FMO) 計算プログラムABINIT-MPの現状と今後. スーパーコンピューティング・ジャパン2024「最先端シミュレーションとAI、これからのスーパーコンピューティング」、2024/3/12
- [13] 望月祐志、中野達也、坂倉耕太、奥脇弘次、土居英男、加藤季広、滝沢寛之、大島聡史、星野哲也、片桐孝洋、FMOプログラムABINIT-MPの整備状況2023. 日本コンピュータ化学会2023年秋季年会、高松、2023/11/24
- [14] 望月祐志、中野達也、坂倉耕太、土

- 居英男、奥脇弘次、加藤季広、滝沢寛之、大島聡史、星野哲也、片桐孝洋、ABINIT-MPプログラムの現状と今後. *J. Comp. Chem. Jpn.* 23, 85-97 (2024)
- [15] 坂倉 耕太、望月 祐志、中野 達也、大島聡史、星野 哲也、片桐 孝洋、FMOプログラムABINIT-MPのGPU化と性能評価. 第30回計算工学講演会、大宮2025/6/6
- [16] A. Veillard, The logic of SCF procedure, in *Computational Techniques in Quantum Chemistry and Molecular Physics*, 201-250 (Reider, Dordrecht, 1975)
- [17] Y. Mochizuki, M. Matsumura, T. Yokura, Y. Hirahara, T. Imamura, Vectorization of Direct Fock Matrix Construction in DIRAC-DHF Calculations. *J. Nuclear Science Tech.* 39, 195-199 (2002)
- [18] < <https://www.jcahpc.jp/> >
- [19] < <https://developer.nvidia.com/nsight-systems> >
- [20] K. Kitaura, E. Ikeo, T. Asada, T. Nakano, M. Uebayasi, Fragment molecular orbital method: an approximate computational method for large molecules. *Chem. Phys. Lett.* 313, 701 (1999)
- [21] D. G. Fedorov, K. Kitaura, The importance of three-body terms in the fragment molecular orbital method, *J. Chem. Phys.* 120, 6832-6840 (2004)
- [22] D. G. Fedorov, K. Kitaura, The three-body fragment molecular orbital method for accurate calculations of large systems. *Chem. Phys. Lett.* 182-187, 433 (2006)
- [23] Y. Komeiji, Y. Mochizuki, T. Nakano, Three-body expansion and generalized dynamic fragmentation improve the fragment molecular orbital-based molecular dynamics (FMO-MD). *Chem. Phys. Lett.* 484, 380-386 (2010)
- [24] T. Nakano, Y. Mochizuki, K. Yamashita, C. Watanabe, K. Fukuzawa, K. Segawa, Y. Okiyama, T. Tsukamoto, S. Tanaka, Development of the four-body corrected fragment molecular orbital (FMO4) method. *Chem. Phys. Lett.* 523, 128-133 (2012)
- [25] S. Amari, M. Aizawa, J. Zhang, K. Fukuzawa, Y. Mochizuki, Y. Iwasawa, K. Nakata, H. Chuman, T. Nakano, VISCANA: Visualized Cluster Analysis of Protein-Ligand Interaction Based on the ab Initio Fragment Molecular Orbital Method for Virtual Ligand Screening. *J. Chem. Inf. Comput. Sci.* 46 (2006) 221-230
- [26] D. G. Fedorov, K. Kitaura, Pair interaction energy decomposition analysis. *J. Comput. Chem.* 28, 222-237 (2006)
- [27] Reinhart Ahlrichs, Efficient evaluation of three-center two-electron integrals over Gaussian functions, *Phys. Chem Chem. Phys.* 6, 5119-5121 (2004)
- [28] A. Asadchev, E. F. Valeev, Memory-Efficient Recursive Evaluation of 3-Center Gaussian Integrals. *J. Chem. Theory Comput.* 19, 1698-1710 (2023)
- [29] J. Kalinowski, F. Wennmohs,

- F. Neese, Arbitrary Angular Momentum Electron Repulsion Integrals with Graphical Processing Units: Application to the Resolution of Identity Hartree – Fock Method. *J. Chem. Theory Comput.* 13, 3160-3170 (2017)
- [30] T. B. Pedersen, F. Aquilante, R. Lindh, Density fitting with auxiliary basis sets from Cholesky decompositions. *Theor. Chem. Acc.* 124, 1-10 (2009)
- [31] T. Ishikawa, K. Kuwata, RI-MP2 Gradient Calculation of Large Molecules Using the Fragment Molecular Orbital Method, *J. Phys. Chem. Lett.* 3, 375-379 (2012)
- [32] T. Ishikawa, K. Sakakura, Y. Mochizuki, RI-MP3 Calculations of Biomolecules Based on the Fragment Molecular Orbital Method. *J. Comput. Chem.* 39, 1970-1978 (2018)
- [33] 望月祐志、中野達也、坂倉耕太、杉崎研司、田中成典、フラグメント分子軌道計算による開殻・多参照の量子系の扱い. *計算工学* 29, 4813-4817 (2024)
- [34] 望月祐志、FMOプログラムABINIT-MPの次世代化. *JHPCN第17回シンポジウム*、2025/7/11