

## A64FX を用いたフラグメント分子軌道計算プログラムの性能評価

満田 晴紀<sup>1</sup> 片桐 孝洋<sup>2</sup> 坂倉 耕太<sup>3</sup> 中野 達也<sup>4</sup> 望月 祐志<sup>5</sup> 大島 聡史<sup>2</sup> 永井 亨<sup>2</sup>名古屋大学 情報学部 コンピュータ科学科<sup>1</sup>名古屋大学 情報基盤センター<sup>2</sup>計算科学振興財団 人材開発グループ<sup>3</sup>国立医薬品食品衛生研究所 生化学部<sup>4</sup>立教大学 理学部化学科<sup>5</sup>

## 1. はじめに

近年、タンパク質などの生体高分子における分子軌道計算の計算機シミュレーションは、高分子ダイナミクス理解やそれを踏まえた創薬などの面から重要視されている。直近では、新型コロナウイルス表面のスパイクタンパク質や、メインプロテアーゼにおける分子同士の相互作用エネルギーを量子力学的に計算することで、タンパク質同士およびタンパク質と薬剤との結合の強さを定量化し、有効な薬剤の開発に役立つ取り組みがなされている。その取り組みで用いられるソフトウェアの一つに、ABINIT-MPがある[1]。

本研究では、フラグメント分子軌道(FMO)計算を行う ABINIT-MP について、A64FX を用いた性能評価を行う。

## 2. ABINIT-MP について

ABINIT-MP は、フラグメント分子軌道(FMO)計算を高速に行うソフトウェアである。

ABINIT-MP が行う FMO 計算で用いるフラグメント分子軌道法 (FMO 法) は、生体高分子のような巨大分子の非経験的分子軌道計算を行うことができる方法である。非経験的分子軌道法とは、分子間のハミルトニアンから電子部分を取り出し、電子に関するシュレディンガー方程式を立て、その方程式を Hartree-Fock (HF) 近似の下で解いて系の電子エネルギーを求める方法である。それを応用した FMO 法では、まず計算対象の分子系をいくつかのフラグメントに分割し、すべてのフラグメントに対し初期電子密度を作

る。その後、フラグメントの HF 計算を行い、電子密度とフラグメントのエネルギーを求める。

この HF 計算において、それぞれのフラグメントにおける HF 計算は相互に独立して行うことができるため並列計算に適している。特に、HF 計算内の 2 電子積分生成のプロセスに対して効率よい計算ができるよう改善していくことが課題になる[2]。

## 3. 性能評価

## 3.1 評価方法

本実験では ABINIT-MP Ver.2 Rev.4 を用いて評価を行った。評価には名古屋大学情報基盤センターが提供するスーパーコンピュータ「不老」Type I サブシステムを使用した。評価環境は表 1 のとおりである。

表 1 性能評価環境

CPU	A64FX (Armv8.2-A + SVE), 48 コア+2 アシスタントコア(I/O 兼計算ノードは 48 コア+4 アシスタントコア), 2.2GHz
メモリ	HBM2, 32GiB
ノード数	2
OpenMP スレッド数	12 / プロセス
コンパイラ	富士通 Fortran コンパイラ 4.6.1 tcsds-1.2.33
コンパイラオプション	-Kfast -Knosimd -Kocl

入力ファイルは cc-pVDZ 基底系の Ala9Gly-ccpvdz-mp2.smp.new3. ajf を用いた。

初めにプログラム全体に富士通社の提供する基本プロファイルを行い、特徴の見られた部分から詳細プロファイルを行った。

## 3.2 結果

初めに Ala9Gly-ccpvdz-mp2.smp.new3.sh ファイルを用いて、プログラム全体の基本プロフ

Performance Evaluation of a Fragment Molecular Orbital Calculation Program on A64FX

1 Haruki Mitsuda, Information system, Computer science, School of Informatics, Nagoya University

2 Takahiro Katagiri, Satoshi Ohshima, Toru Nagai, Information Technology Center, Nagoya University

3 Kota Sakakura, Foundation for Computational Science

4 Tatsuya Nakano, National Institute of Health Science

5 Yuji Mochizuki, Rikkyo University

ファイルを行った。結果を図2に示す。

図2の結果から、F積分の最深部に位置するsub\_系の関数にコストがかかっていることが分かった。

Cost	% Operation (s)	Barrier	% Start	End	
161607	100.0000	16168.7451	15725	9.7384	Application
23916	14.7989	2392.7847	0	0.0000	1 551 get_tel_pd_fix2
20887	12.9245	2089.7407	0	0.0000	1 551 darpj
16180	10.0119	1618.8115	15518	9.6023	-- -- unknown
6885	4.2603	688.8424	0	0.0000	1 257 get_tel_rs_fix
5859	3.6255	586.1902	0	0.0000	16 87 sub_ssss
3487	2.1391	345.3747	0	0.0000	15 175 sub_spps
3333	2.0624	333.4664	52	0.0322	463 648 dm2_ene_sj_OMP_i
3072	1.9009	307.3524	0	0.0000	16 174 sub_ssss
3048	1.8861	304.9513	0	0.0000	16 108 sub_spps
2707	1.6751	270.8342	0	0.0000	16 188 sub_spps

図2 基本プロファイル結果

続いて、sub\_xxxx 関数に着目し、詳細プロファイルを行った。sub\_xxxx.F90 ファイルのループ区間に計測区間を設けて詳細プロファイルを行った。結果は図3、図4とおりでである。

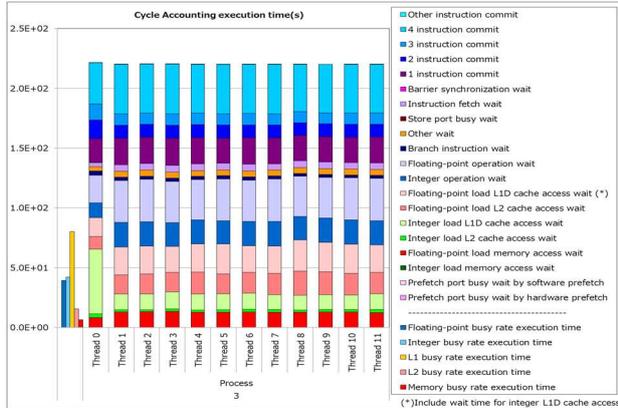


図3 詳細プロファイル結果 (実行時間の内訳)

Process	Thread	Execution time (s)	Gflops	Floating-point operation peak ratio (%)	Memory throughput (GB/s)	Memory peak ratio (%)	Effective instruction	Floating-point operation	SIMD instruction rate (%) (/Effective instruction)	SVE operation rate (%)	Floating-point pipeline Active element rate (%)	IPC
3	0	4.42E+02	0.66	1.00%	0.34		7.45E+11	2.92E+11	4.96%	74.87%	70.13%	0.81
3	1	4.41E+02	0.66	1.00%	0.32		4.39E+11	2.91E+11	8.51%	74.66%	70.09%	0.77
3	2	4.41E+02	0.68	1.03%	0.32		4.36E+11	3.00E+11	9.05%	76.07%	70.24%	0.78
3	3	4.41E+02	0.63	0.95%	0.31		4.32E+11	2.79E+11	8.40%	76.18%	70.08%	0.77
3	4	4.41E+02	0.68	1.02%	0.34		4.73E+11	2.99E+11	8.11%	75.20%	70.29%	0.81
3	5	4.41E+02	0.66	1.00%	0.31		4.20E+11	2.91E+11	9.05%	76.01%	70.29%	0.75
3	6	4.41E+02	0.64	0.97%	0.30	15.1%	4.21E+11	2.84E+11	8.71%	75.57%	70.11%	0.75
3	7	4.41E+02	0.67	1.01%	0.31		4.22E+11	2.94E+11	8.78%	76.28%	69.97%	0.75
3	8	4.41E+02	0.74	1.11%	0.36		4.74E+11	3.24E+11	9.16%	76.88%	70.68%	0.78
3	9	4.41E+02	0.69	1.05%	0.33		4.37E+11	3.06E+11	9.15%	75.90%	70.15%	0.77
3	10	4.41E+02	0.70	1.06%	0.33		4.46E+11	3.10E+11	8.97%	75.25%	70.09%	0.77
3	11	4.41E+02	0.68	1.02%	0.30		4.12E+11	2.86E+11	8.50%	76.24%	70.15%	0.75
3	12	4.41E+02	0.69	1.02%	0.37	15.1%	3.56E+11	3.56E+11	8.33%	75.60%	70.19%	0.77
3	OMP 3 total	4.41E+02	8.09	1.02%	0.37	15.1%	3.56E+11	3.56E+11	8.33%	75.60%	70.19%	0.77

図4 詳細プロファイル結果 (統計情報)

図3から、整数・浮動小数点演算待ちや、プリフェッチ待ち時間が大きい。図4から、SIMD命令の割合は平均8.33%と低く、SIMD命令があまり生成されていない。

そこで、当該ループに対して、SIMD化を促進するディレクティブを挿入した。その後、詳細プロファイルを行った。結果を図5、図6に示す。

Process	Thread	Execution time (s)	Gflops	Floating-point operation peak ratio (%)	Memory throughput (GB/s)	Memory peak ratio (%)	Effective instruction	Floating-point operation	SIMD instruction rate (%) (/Effective instruction)	SVE operation rate (%)	Floating-point pipeline Active element rate (%)	IPC	gfps
3	0	2.22E+02	1.20	1.90%	0.66		4.52E+11	2.66E+11	8.04%	74.50%	69.93%	1.03	2.04
3	1	2.20E+02	1.22	1.92%	0.62		3.16E+11	2.68E+11	11.66%	74.60%	69.98%	1.08	1.43
3	2	2.20E+02	1.23	1.96%	0.61		3.15E+11	2.76E+11	12.37%	76.1%	70.26%	1.07	1.63
3	3	2.20E+02	1.13	1.81%	0.61		3.06E+11	2.52E+11	11.79%	76.26%	69.91%	1.09	1.38
3	4	2.20E+02	1.24	1.97%	0.64		3.28E+11	2.74E+11	11.49%	75.16%	70.17%	1.07	1.49
3	5	2.20E+02	1.20	1.90%	0.59		3.09E+11	2.69E+11	12.12%	76.07%	70.15%	1.08	1.40
3	6	2.20E+02	1.17	1.83%	0.58	2.94%	3.05E+11	2.97E+11	11.86%	75.99%	69.93%	1.08	1.38
3	7	2.20E+02	1.21	1.92%	0.59		3.12E+11	2.67E+11	11.69%	74.14%	69.82%	1.08	1.41
3	8	2.20E+02	1.19	1.87%	0.70		3.41E+11	3.01E+11	12.40%	77.01%	70.29%	1.04	1.65
3	9	2.20E+02	1.27	2.01%	0.63		3.25E+11	2.89E+11	12.32%	75.93%	69.99%	1.04	1.45
3	10	2.20E+02	1.29	2.04%	0.63		3.22E+11	2.84E+11	12.24%	75.24%	69.89%	1.04	1.40
3	11	2.20E+02	1.24	1.93%	0.59		3.07E+11	2.72E+11	12.54%	76.22%	69.97%	1.08	1.40
3	12	2.20E+02	14.81	1.92%	7.32	2.94%	3.95E+11	3.26E+11	11.92%	75.62%	69.91%	1.07	1.28
3	OMP 3 total	2.20E+02	14.81	1.92%	7.32	2.94%	3.95E+11	3.26E+11	11.92%	75.62%	69.91%	1.07	1.28

図5 SIMD化後の詳細プロファイル結果 (実行時間の内訳)

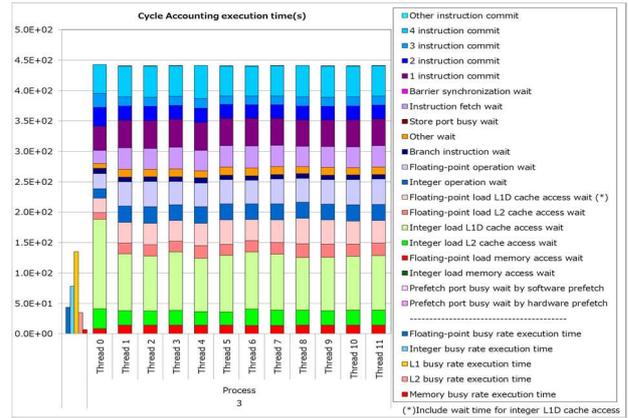


図6 SIMD化後の詳細プロファイル結果 (統計情報)

図5、図6からSIMD化促進により、実行時間が約半分となったことがわかる。またそれに伴い、FLOPS値も2倍近く向上し、対ピーク性能も1.95%になった。また図6から、演算待ち時間の占める割合が減少し、逆に、L1Dキャッシュの整数ロードアクセス待ち時間の占める割合が増加した。当日の発表では、その他の最適化と性能評価の詳細を報告する予定である。

#### 4. おわりに

本報告では、フラグメント分子軌道(FMO)計算ソフトウェアABINIT-MPのA64FX向けコード最適化とその性能評価を行った。性能プロファイルの結果、SIMD化促進のディレクティブ挿入により、約2倍の速度向上を達成できる例を確認した。

今後の課題として、ループ分割や融合のループ変換を施す場合の性能評価など、さらなるコード最適化と、性能解析を行う必要がある。

謝辞 本研究はJSPS 科研費 JP19H05662の助成を受けたものです。

#### 参考文献

[1] S.Tanaka,et.al. “Electron-correlated fragment-molecular-orbital calculations for biomolecular and nano systems”, Phys. Chem. Chem. Phys.,pp.10310-10344, (2014).  
 [2] 望月祐志, 他, “FMO プログラム「ABINIT-MP」のA64FX スーパーコンピュータ向け高速化と大規模化”, 計算工学ナビ, 21, pp.6, (2021).