

フラグメント分子軌道計算プログラム ABINIT-MP の高速化

望月祐志^{1,2}, 中野達也³, 坂倉耕太⁴, 渡邊啓正⁵, 佐藤伸哉⁶, 奥脇弘次¹, 土居英男¹,大島聡史⁷, 片桐孝洋⁸立教大学 理学部化学科¹東京大学 生産技術研究所²国立医薬品食品衛生研究所 生化学部³計算科学振興財団 人材開発グループ⁴HPC システムズ株式会社 HPC 事業部⁵NEC ソリューションイノベーション株式会社 関西支社 第三グループ⁶九州大学 情報基盤研究開発センター⁷名古屋大学 情報基盤センター⁸

1. はじめに

フラグメント分子軌道(FMO)法は、並列処理を駆使してタンパク質や核酸などの生体分子を非経験的なレベルで扱える手法です。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ます。ABINIT-MP [1, 2]は、望月が取り纏め役を務める純国産のプログラムで、スパコンでの超並列実行に強みがあり、2020年度の「富岳」の優先的な試行的利用による新型コロナウイルスの特別プロジェクトに第4課題「新型コロナウイルス関連タンパク質に対するフラグメント分子軌道計算」として参画し、多数の応用計算の成果を出すことができました。ただ、その中で高速化と大規模系への対応が課題として認識されました。2021年度からは、これらに対する対策を進めてきています。今回の発表では、「富岳」や「不老」Type I等の富士通のA64FX系スパコン、それにNECのSX-Aurora TSUBASA(以下SX-AT)向けの高速化について報告します。

2. 高速化のポイント

図1に、新型コロナウイルスのメインプロテア

Speed-up of ABINIT-MP as a fragment molecular orbital program

1 Yuji Mochizuki, Rikkyo University & University of Tokkyo

2 Tatsuya Nakano, National Institute of Health Science

3 Kota Sakakura, Foundation for Computational Science

4 Hiromasa Watanabe, HPC Systems, Inc.

5 Shinya Sato, NEC Solution Innovators, Ltd.

6 Koji Okuwaki, Rikkyo University

7 Hideo Doi, Rikkyo University

8 Satoshi Ohshima, Kyushu University

9 Takahiro Katgagiri, Nagoya University

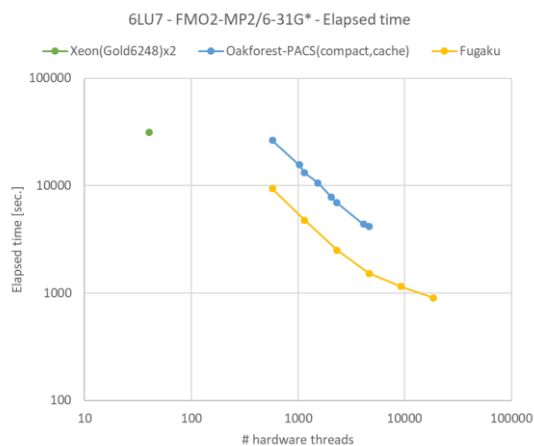


図1. ABINIT-MPの並列性能

ーゼにN3リガンドが結合した複合体(PDB-ID: 6LU7)のFMO-MP2/6-31G*(FMO計算で多用される2次摂動レベルの計算)ジョブの並列性能ですが、ABINIT-MPのスケールは及第レベルに達しています。そこで、高速化は並列化ではなく、実際の計算部を対象にすることになりました。

2020年版のOpen Ver. 1 Rev. 22を使い、FMO-MP2/6-31G*レベルでAlaGlyでコストのプロファイリングをしたところ、基底関数による2電子積分の生成に全コストの1/2がかかり、ハートリーフォック(HF)計算での添字処理とFock行列の構築が1/4コスト、その他が1/4コストとなることが分かりました。MP2計算は、生成された基底関数添字の2電子積分の4回の線形変換が本質でDGEMM処理が可能なることから目立ちません。従って、2電子積分の生成とHFの処理を高速化の対象としました。

3. A64FX 向け高速化

ABINIT-MP の 2 電子積分生成のルーチンは、垂直漸化式関係[3]に基づき、基底関数の {s, p, d} の軌道タイプの 4 つの組み合わせからコードジェネレータによって生成されており、総数としては 81 種類になります。詳細なプロファイル分析から、相対的にコストの高いルーチンに対して SIMD 化を指示詞挿入と一部変数のスカラ化などを手動で行いました[4]。こうした改良を反映させた第一弾として Ver. 2 Rev. 4 を 2021 年 9 月にリリースしました(大規模系対応も含む)。Ver. 1 Rev. 22 比の加速では対象に依存しますが、1.2 ~ 1.4 倍です[5]。

Ver. 2 Rev. 4 のリリース後、2 電子積分ルーチン群に対して SIMD 化に加えてループ分割の改良も行い、軌道タイプの組み合わせに依りますが 20%~30%の加速を得ました。また、Fock 行列の構築ルーチンでは $(1/2)^n$ ($n=1, 2, 3$) の因子を乗じて添字の同値性判断の if 分岐を排除する改良で 30%加速しました。これらを反映させた版が Ver. 2 Rev. 5 です。その後、フラグメントモニター段階で環境静電ポテンシャルを決するサイクルを低減する外挿法を導入した Rev. 5', 別途 2 電子積分のインコアバッファリングを可とした Rev. 5'' を作成し、現在はこれら 2 つをまとめた Rev. 6 が最新作業版となっています。

表 1 に、20 個のアミノ酸から成る Trp-Cage ミニタンパク質の FMO-MP2/6-31G*ジョブのタイミングを示します。実行は、名古屋大学の「不老」 Type I の 10 ノードで OpenMP で 24 スレッド、MPI で 20 プロセスの設定で行いました。改良前の Ver. 1 Rev. 22 を基準にした加速を見ますと、最新の Ver. 2 Rev. 6 では約 2 倍の加速が得られています。基底関数を cc-pVDZ とするケースも含めて他の系でもベンチマークテストを行ったところ、Ver. 1 Rev. 22 に比べての加速は 1.5~2.0 倍になっています。

表 1. Trp-Cage の FMO-MP2 ジョブのタイミング

Ver.	Rev.	Date	Sec.	Acc.
V1	R22	2020/6/3	469.6	1.00
V2	R4	2021/9/16	413.9	1.13
V2	R5	2021/12/9	344.2	1.36
V2	R5'	2022/7/13	294.2	1.60
V2	R5''	2022/9/8	282.7	1.66
V2	R6	2022/11/1	240.3	1.95

4. SX-AT 向け高速化

SX-AT のプロファイリングでもベクトル化によって改良すべきポイントは 2 電子積分の生成となりました。そこで、ループ分割等の改造を積分ルーチン群に対して行いました。その結果、前出の 6LU7 モデルの FMO-MP2/6-31G*ジョブでは、従前の Ver. 1 Rev. 22 に比べて 4.8 倍もの高速化を得ました。

5. 今後のリリース

ABINIT-MP は全国の HPCI 拠点でライブラリプログラムとして登録・公開されており、今後も改良版をリリースしていきます。次期版は 2023 年 4 月に公開を目指している Ver. 2 Rev. 8 で、A64FX 向けの高速化改良については本稿で紹介した Ver. 2 Rev. 6 がベースになります。

謝辞

本研究で A64FX 向けの高速化は、JHPCN 課題 jh210036-NAH, jh220010 の枠で行いましたが、富士通 SS 研の「A64FX システムアプリ性能検証 WG」の活動とも連動しました。また、SX-AT 向けの改造は立教大学と NEC との共同研究に依ります。なお、資金面では立教 SFR の支援を受けました。

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] Y. Mochizuki, et al., “The ABINIT-MP program”, pp. 53-67, “Recent Advances of the Fragment Molecular Orbital Method - Enhanced Performance and Applicability”, ed. Y. Mochizuki, et al., (January 2021, Springer).
- [3] S. Obara, et al., “Efficient recursive computation of molecular integrals over Cartesian Gaussian functions”, *J. Chem. Phys.*, pp. 3963-3974, (1986).
- [4] 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).
- [5] 望月祐志, 他, “FMO プログラム ABINIT-MP の整備状況 2021”, *J. Comp. Chem. Jpn.*, pp. 132-136, (2021).