

1E01

超並列フラグメント分子軌道法プログラム OpenFMO の開発

(1 九大, 2 九州先端研, 3 筑波大, 4 東工大, 5 産総研) ○稲富雄一¹, 眞木淳²,
本田宏明¹, 石元孝佳¹, 梅田宏明³, 渡邊寿雄⁴, 池上努⁵, 青柳睦¹

Development of massively parallel program of fragment molecular orbital method, OpenFMO

(1 Kyushu Univ., 2 ISIT, 3 Univ. of Tsukuba, 4 Titech, 5 AIST)

○Yuichi Inadomi¹, Jun Maki², Hiroaki Honda¹, Takayoshi Ishimoto¹,
Hiroaki Umeda³, Toshio Watanabe⁴, Tsutomu Ikegami⁵, Mutsumi Aoyagi¹

【はじめに】フラグメント分子軌道(FMO)法はタンパク質や DNA などの大規模生体分子の電子状態計算を高速に行うために開発された計算手法である。巨大分子を小さなフラグメントに分割して、フラグメント (モノマー) および、フラグメントペア (ダイマー) に対する小規模な電子状態計算を行うことで、分子全体の電子状態を近似する。各モノマー (ダイマー) に対する電子状態計算を独立に計算できること、ならびに、各小規模電子状態計算自身もさらに並列処理可能であることなどから、大規模並列処理向きの計算手法であると考えられており、数 1000 並列での FMO 計算を効率よく行えることも示されている。しかしながら、数万～数 10 万といった超並列 FMO 計算を効率よく行うためには、更なる並列性能向上が必要となる。また、FMO 計算をより高速に実行できる環境を整えるためには、今後開発が予定されているエクサスケールスパコンで想定されるアーキテクチャ向けのプログラム開発も必須となる。我々は、効率的な超並列 FMO 計算を目指して並列 FMO プログラム OpenFMO の開発を行っており、高速化のための様々な最適化を行ってきた。また、今後出現が予想される計算機アーキテクチャを踏まえたコード開発を計算機科学の研究者と共同で行っているところである。本発表では、これまでの成果、および、現在行っている研究内容について報告する。

【OpenFMO とは】OpenFMO は九大と九州先端研でスクラッチから開発され C 言語で記述された並列 FMO プログラムである。Hartree-Fock 法に基づいた FMO 計算に特化しているため、ソースコードが 5 万行強と非常にコンパクトで、最適化などを比較的容易に適用することができる。また、並列化において MPI や OpenMP という事実上の標準となっている並列化手法を用いていること、ならびに、特定の計算機ベンダーが提供する特殊な機能を使っていないことなどから、ほとんどの並列計算機環境で利用可能である。

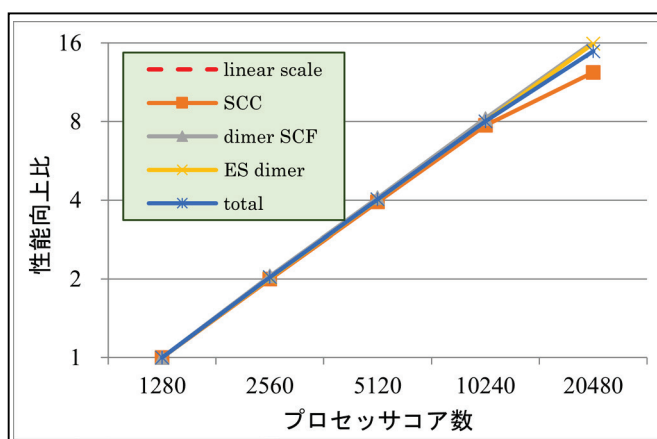
【最適化 1 OpenMP/MPI ハイブリッド並列化】京 (コンピュータ) をはじめとする最近の大型計算機は、数～数 10 プロセッサ (コア) を搭載した多数の小型計算機 (ノード) を Infiniband などの高速ネットワークで相互結合したクラスタ型並列計算機アーキテクチャを持つ。そのような計算機で効率よく動作するプログラムの開発においては、同一ノード上にあるコア間の通信 (ノード内通信) とノードを跨ぐ通信 (ノード間通信) の通信性能差を考慮した並列化が必要となる。そのために、我々は、ノード内並列化には OpenMP によるスレッド並列化を、また、ノード間並列化には MPI によるプロセス並列化をそれぞれ用いた OpenMP/MPI ハイブリッド並列化を分子積分コードに適用した。この並列化手法は、京における並列化手法としても推奨されている。

【最適化 2 OpenMP/MPI によるグローバルカウンタの実装】FMO 計算だけでなく一般の量子化学計算におけるカーネルコード (計算量の大きい部分) の 1 つである分子積分を多数のプロセスを用いて効率的に並列処理するためには、グローバルカウンタ(GC)を用いた動的負荷分散の適用が有効である。GC は並列処理を行っている全プロセスが参照可能なカウンタであり、GC にアクセスしてカウ

ンタ値を取得し、その値に基づいて積分計算を行い、それが終了したら、次のカウンタ値を取得して、...を繰り返すことで積分計算を各プロセスに分配して並列処理を行う。GAMESSやNWChemといった汎用量子化学計算プログラムで用いられているGCの実装にはARMCIライブラリが用いられているが、我々のターゲットマシンの1つである京で動作するARMCIライブラリは京の各ノード上の8コアのうち1コアをライブラリが占有する(計算に使えない)実装となっているため、その利用は効率的な並列処理の妨げとなる。そこで我々は、ARMCIの代わりにOpenMPとMPIを用いて計算機資源の無駄の少ないGCを実装した。

【最適化3 中間データの分散保存・アクセス方法の改良】FMO計算では、並列計算を行っているすべてのプロセスからモノマー密度行列データの参照、および、更新が可能である必要がある。このモノマー密度行列データは入力データの規模に比例するため、巨大な入力に対する効率的なFMO計算を可能にするためには、データを複数のノードに分散保存して、分散保存されたデータに効率よくアクセスする必要がある。この目的を達成するために、今回、(1)MPI-2規格のMPIライブラリで実装されている片側通信機能を使う手法、および、(2)データ保存を行う専用プロセス(データサーバ)を用いる手法、の2つを実装して、その性能評価を行った。その結果、片側通信を用いる方法ではMPI-2での片側通信機能の実装の影響で、密度行列データへのアクセス性能が非常に悪くなることが分かった。一方で、データサーバを用いる方法では、データアクセスによる遅延もほとんどなく、非常に効率よく密度行列データにアクセスできることが分かった。

【性能評価】最適化を適用したコードを用いて、並列性能の評価を行った。用いた計算機は九州大学情報基盤研究開発センターの富士通PRIMERGY CX400(8-core Xeon×2/node, 1476 nodes, interconnect=Infiniband FDR)である。計算に用いるコア数を1280~20480まで変化させてアデノウィルスKNOBドメイン(PDB ID=1nob, モノマー数576)に対するFMO計算を行い、その計算時間を測定した(右図参照)。その結果、FMO計算全体では2万並列でも非常に効率よく並列処理



図：大規模並列時の性能向上比(1280並列時を1とした場合)

できることが分かった。特に、dimer SCF計算、および、近似ダイマー(ES dimer)計算部分は、2万並列でも、ほぼ理想的な速度向上を示した。一方で、モノマーSCF(SCC)計算部分は、2万並列時に並列化効率が低下していることが分かる。これは、各プロセスに割り当てられる計算量が小さくなったため、プロセッサ間の負荷バランスがとりにくくなったことが原因であると考えられる。更なる効率の向上のためには、SCC部分の並列化効率を上げるための工夫が必要となる。

【検討中の課題】現在、高性能なGC、および、データサーバの実装を容易にするためのミドルウェアの開発を九大と富士通の共同研究、および、産総研で行っている。また、次世代スパコンへの搭載が確実視されている演算加速器を用いた分子積分計算プログラムについても、九大、筑波大、東工大の共同で開発している。また、計算中に計算機の一部が故障しても、計算を続行できる(耐障害性を持つ)FMOプログラムの開発を、九大と筑波大、および、理研との共同研究で行っている。

謝辞本研究は、科学研究費補助金基盤研究(C)「超並列フラグメント分子軌道法プログラムライブラリの開発」(課題番号22550015)、JST,CRESTの研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」の研究課題「省メモリ技術と動的最適化技術によるスケーラブル通信ライブラリの開発」、ならびに、研究課題「ポストペタスケールシステムのための電力マネージメントフレームワークの開発」の支援を受けている。