

## 並列 FMO プログラム OpenFMO の高性能化

(九大情基セ<sup>1</sup>, 九州先端研<sup>2</sup>, 理研<sup>3</sup>, CREST<sup>4</sup>) 稲富雄一<sup>1</sup>, 眞木淳<sup>2</sup>, 本田宏明<sup>1,4</sup>,  
高見利也<sup>1,4</sup>, 小林泰三<sup>1</sup>, 青柳睦<sup>1</sup>, 南一生<sup>3</sup>

### Performance Improvement of Parallel FMO Program, OpenFMO

(RIIT Kyushu Univ.<sup>1</sup>, ISIT<sup>2</sup>, RIKEN<sup>3</sup>, CREST<sup>4</sup>) Yuichi Inadomi<sup>1</sup>, Jun Maki<sup>2</sup>,  
Hiroaki Honda<sup>1,4</sup>, Toshiya Takami<sup>1,4</sup>, Taizo Kobayashi<sup>1</sup>, Mutsumi Aoyagi<sup>1</sup>, and  
Kazuo Minami<sup>3</sup>

【はじめに】フラグメント分子軌道 (FMO) 法は, タンパク質や DNA, 糖鎖などの巨大生体分子に対する量子化学に基づいた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。FMO 法では, 巨大分子を小さなフラグメントに分割して, 各フラグメント (モノマー) やフラグメントペア (ダイマー) に対する小規模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, ダイマーの電子状態を独立に計算できるため, 複数の小規模電子状態計算を並列に処理できる (疎粒度並列処理)。また, 各小規模電子状態もさらに並列処理 (細粒度並列処理) できるため, FMO 法は大規模並列処理向きの計算手法と考えられており, 現在開発中の「京」コンピュータにおけるターゲットアプリケーションとしても取り上げられている。しかしながら, 1 万~数 10 万並列といった超並列処理時に効率的な FMO 計算を行うためには, 超並列実行を行うための最適化が必須である。

我々は, FMO 法がどの程度までの並列処理に耐え得るのか, という計算機科学的な観点から, 九大などで開発された FMO 計算プログラム OpenFMO の超並列処理に向けた最適化を行っている。昨年度はモノマーやダイマーの電子状態計算の負荷分散に注目した最適化について報告した。今回は, FMO 計算で最も通信コストがかかるモノマー密度行列データの保存, および, アクセス方法について検討したので, その結果を報告する。

【モノマー密度行列データの取り扱い】FMO 計算で行うモノマーやダイマーなどに対する小規模電子状態計算を行う場合には, 周辺モノマーの密度行列データが必要になる。計算を行う各プロセスがすべてのモノマー密度行列データを保存すると入力が大きくなった場合に対応できなくなるため, 大規模入力にも対応するためにはモノマー密度行列データを分散保存して, そのデータに対して効率的にアクセスする必要がある。そのために, 今回は 2 つの方法を検討した。1 つは, 計算に参加している各プロセスに分散して保存する方法である (方法 1, 図 1 参照)。この方法では, データを多くのプロセスで分散保存するため, 入力データに対する scalability が非常に高い。ただし, この方法を用いるためには MPI-2 など実装されている片側通信機構を用いる必要がある。2 つ目は, データ保存のための専用プロセスを用いる方法である (方法 2, 図 2 参照)。この方法では, モノマー密度行列データを保持してワーカプロセスからのアクセス要求に回答することを専門とするストレージプロセス (storage group に属する) と, 計算を専門に行うワーカプロセス (いずれかの worker group に属する) とを分離する。入力データに対する scalability にはストレージプロセスの増減で対応でき, また, MPI-1 でも実装されている 1 対 1 通信だけ

で記述可能である。一方で、ストレージプロセスとワーカプロセスの2種類を記述する必要があるため、コードが複雑になる。

【結果】密度行列の保存方法として前述の2つの方法を実装して、その性能を調べた。その結果を図3に示す。この図はあるモノマーの小規模電子状態計算に対する実行プロファイルで、横軸が時間経過、縦軸が並列処理で用いたプロセスのID（MPIにおけるrank番号）を表している。縦方向の白線は1対1通信を行っていることを示しており、黒色以外の部分は何らかの通信、あるいは同期待ちをしていることを表している。この結果から、MPI-2の片側通信機構を用いた方法1の実装では、大きな通信待ちが発生しており、性能が非常に悪いことが分かる。一方、データ保存専用プロセスを用いた方法2では、通信待ち時間がほとんどなく、効率よく密度行列データにアクセスできていることが分かった。

8000並列での性能評価でも、方法2によるモノマー密度行列データへのアクセス性能が非常に高いことが示された。

謝辞 本研究の一部は理化学研究所と九州大学との共同研究で行っている。また、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)、および、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」の支援を受けている。さらに、学際大規模情報基盤共同利用・共同研究拠点の公募型共同研究平成23年度採択課題「超並列フラグメント分子軌道法プログラムOpenFMOの性能評価と高性能化」による計算機利用を行っている。

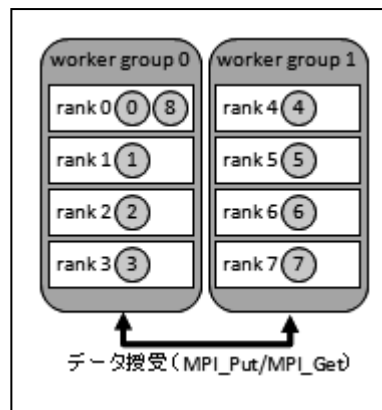


図1:方法1によるモノマー密度行列データの分散保存方法 (ワーカプロセス数=8, モノマー数=9の場合)

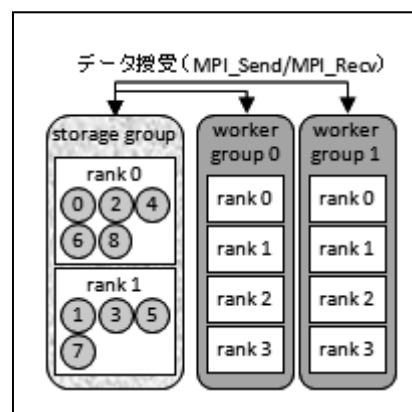


図2:方法2によるモノマー密度行列データの分散保存方法 (ワーカプロセス数=8, ストレージプロセス数=2, モノマー数=9の場合)

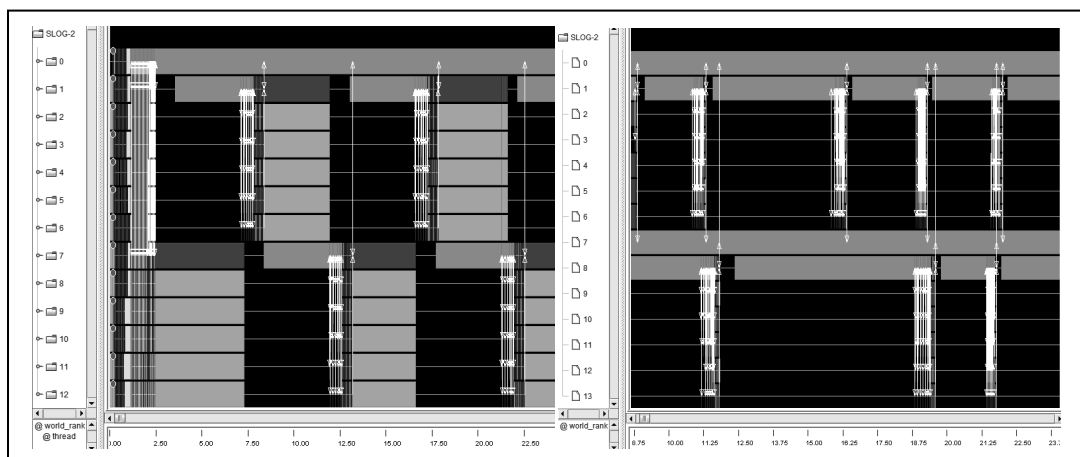


図3:モノマー電子状態計算の実行プロファイル(抜粋)  
(左)方法1(MPI-2の片側通信利用) (右)方法2(データ保存専用プロセス利用)