12-NA07

並列分子軌道計算プログラム OpenFMO の高性能化

南一生（理化学研究所）

概要
京コンピュータをはじめとしたクラスタ型超並列計算機での効率的な超並列実行を目指して，並列フラグメント分子軌道（FMO）プログラム OpenFMO の高性能化を行うことが本研究の目的である．前年度までの最適化の加えて，今年度は，数万～数 10 万並列実行時のさらなる性能向上を目指して最適化を行う予定である．中間報告までに FMO 計算で多数回実行されるモノマーやダイマーに対する小規模電子状態計算に含まれる繰り返し計算（SCF 計算）の並列性能の低さが全体の並列性能の低下を招いていることが分かった．そこで，SCF 計算部分で性能低下を招いている原因のひとつと思われる reduction 処理部分に MPI_Allreduce ではなく，MPI_Reduce 関数を用いるように変更した．その結果，モノマー電子状態計算において，256 並列時に並列化効率が約 0.77 ポイント改善することが確認された．

1. 研究の目的と意義

【研究の目的】

フラグメント分子軌道（Fragment Molecular Orbital Method, FMO）法[1-3]はたんぱく質や DNA，糖鎖などの生体分子に対する第一原理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 法では計算対象となる巨大な分子を 20～40 原子程度の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）に対する小規模な電子状態計算(フラグメント電子状態計算)を行うことで，分子全体の電子状態を近似する．複数のモノマー，あるいはダイマーに対するフラグメント電子状態計算を並列に実行（粗粒度並列化）できること，ならびに，各モノマー，ダイマーの電子状態計算自身も更に並列処理（細粒度並列化）が可能であることから，FMO 法は超並列処理向きの計算手法であると考えられている．GAMESS[4]や ABINIT-MP[3,5]といった並列 FMO 計算プログラムの既存実装があり，1,000 並列程度であれば効率よく並列処理できることが確認されている[6]．しかし，10 万並列を超えるような超並列実行時に高い並列化効率を保つためには，負荷分散を正確に行う，通信負荷を削減する，あるいは，使用する計算機のアーキテクチャに適したプログラミングを行う，などの最適化を行って，並列化効率を落とす要因を可能な限り取り除く作業が必須である．そのような精緻な最適化作業を行う場合には，対象とするプログラムが単純な構造を持っている方が好ましい．九州大学で開発されている並列 FMO 計算プログラム OpenFMO[7,8]は，非経験的第一原理電子状態計算手法の中で最も単純な Hartree-Fock（HF）法を基にした FMO 計算に特化したプログラムである．そのため，ソースコードが比較的短く（全体で約 54,000 行），標準的な並列処理ライブラリである MPI を用いた並列化が行われているため，実行プロファイル取得，ならびに，最適化作業が行いやすい．一昨年度，および，昨年度の成果で，OpenFMO プログラムを用いて FMO 計算の効率的な超並列処理において重要となる，各フラグメント電子状態計算の細粒度並列処理部分についての性能評価を行い，効率を低下させている場所を特定して，動的負荷分散を用いることで負荷均等化を図り並列性能を向上させたり，FMO 計算で扱うモノマー密度行列と呼ばれる中間データへのアクセスに伴う通信に注目してアクセス性能を向上させ，データアクセスに伴う通信による並列性能低下を抑えることが出来るようになった．今年度は，数 1000～数万並列での効率的な FMO 計算全体を効率的に行うための最適化を，理論化学の研究者と協力して検討することにした．

【研究の意義】

FMO 法は，これまでにもたんぱく質や DNA などの

生体分子と薬剤との相互作用解析に応用されてきており，特に，創薬分野における有用な道具として期待されているシミュレーション手法の 1 つである．創薬分野では，候補化合物の絞り込み（スクリーニング）に計算機シミュレーションを用いることが考えられる．FMO 法をスクリーニングに用いるためには，現在よりも短時間に，多数回計算することが必要になる．そのためには，中規模，あるいは，大規模分子に対する FMO 計算を，今よりも格段に高速に行う必要がある．

FMO 法は粗粒度，細粒度の 2 段階並列化が可能な計算手法であるため，超並列処理向きの計算手法であるが，既存実装は最近の大型計算機の主流となっている小型 SMP 計算機（計算ノード）を超高速相互結合網で接続した SMP クラスタ型計算機の特徴を考慮したプログラム設計になっていないため，数万～数 10 万並列（超並列）実行時には，効率的な処理が困難だと考えられる．現在稼働中，あるいは，開発中のスーパーコンピュータは数万プロセッサ（コア）を搭載した超並列計算機であり，今後そのような計算機を利用する機会が増加することを考えると，近い将来には，比較的容易に数万並列のプログラムを実行できるようになると思われる．そのような環境下で，大規模 FMO 計算を高速に実行するためには，超並列実行時に効率的に並列処理できるプログラムの開発が必須である．超並列計算機アーキテクチャを考慮して高性能な超並列 FMO プログラムが作成できれば，創薬分野におけるスクリーニングを計算機シミュレーションでサポートすることが可能となるため，薬剤の開発コストの削減や開発期間の短縮に寄与できると考えている．

2. 当拠点公募型共同研究として実施した意義

(1) 共同研究を実施した大学名と研究体制

大学名：九州大学

研究体制：

南一生（理化学研究所）

分散データ保存，アクセス方法に対する計算機科学的な観点からの支援

高見利也（九州大学）

OpenFMO の性能評価

稲富雄一（九州大学）

(2) 共同研究分野

超大規数値計算系応用分野

(3) 当公募型共同研究ならではという事項など

超並列化に向けたプログラムの最適化には，実機を用いた性能評価を行うことが不可欠である．今回，スーパーコンピュータを実際に利用して，性能評価を行うことができ，並列性能を向上させるための方針を決めるための情報を得ることができた．

3. 研究成果の詳細と当初計画の達成状況

(1) 研究成果の詳細について

【FMO 法概要】

FMO 法は，計算対象となる大規模分子を，20～40 原子の小さなフラグメントに分割して，分割した各フラグメント（モノマー），および，フラグメントペア（ダイマー）に対する小規模電子状態（分子のエネルギー，電子分布の様子など）計算を行うことで，分子全体の電子状態を近似する計算手法である．FMO 法で最終的に求めたい量は，分子全体のエネルギー $E_{\mathrm{total}}^{\mathrm{FMO}}$，および，密度行列 $\mathbf{D}_{\mathrm{total}}^{\mathrm{FMO}}$ であるが，FMO 法では以下のように近似する．

$$E_{\mathrm{total}}^{\mathrm{FMO}} = \sum_{I>J}^{N_{\mathrm{frag}}} E_{IJ} - \left(N_{\mathrm{frag}} - 2\right) \sum_{I}^{N_{\mathrm{frag}}} E_I \qquad (1)$$

$$\mathbf{D}_{\mathrm{total}}^{\mathrm{FMO}} = \sum_{I>J}^{N_{\mathrm{frag}}} \mathbf{D}_{IJ} - \left(N_{\mathrm{frag}} - 2\right) \sum_{I}^{N_{\mathrm{frag}}} \mathbf{D}_I$$

ここで，N_{frag} はフラグメント（モノマー）数，$\{E_I\}$（$\{E_{IJ}\}$）は，モノマー（ダイマー）のエネルギー，また，$\{\mathbf{D}_I\}$（$\{\mathbf{D}_{IJ}\}$）は，モノマー（ダイマー）の密度行列を，それぞれ表わす．モノマーの電子状態計算を行うためには，計算しているモノマー自身の密度行列のほかに，その近傍にあるモノマーの密度行列も必要となる．

$$E_I^{n+1}\left(\mathbf{D}_I^{n+1}\right) \leftarrow f\left(\mathbf{D}_I^n, \left\{\mathbf{D}_K^n\right\}_{K \in \text{neighborhood of monomer } I}\right) \qquad (2)$$

式(2)は，モノマーのエネルギー，密度行列が該当モノマーとその近傍にあるモノマーの密度行列の関数であることを表わしている．一般に，式(2)

の引数として与えているモノマー I の密度行列 $\mathbf{D}_I^n$ と，出力として得られる $\mathbf{D}_I^{n+1}$ は異なる．FMO 法では，この関数の入力 $\{\mathbf{D}_K^n\}$ と出力 $\{\mathbf{D}_K^{n+1}\}$ の差が十分に小さくなる（モノマーの密度行列が収束する）まで，各モノマーの電子状態計算を繰り返す．この処理を，Self- Consistent Charge (SCC)処理，と呼ぶ．さらに，FMO 計算では，ダイマーの電子状態計算を，SCC 処理で収束したモノマー密度行列を用いて行う．

$$E_{IJ}(\mathbf{D}_{IJ}) \leftarrow f\left(\mathbf{D}_I, \mathbf{D}_J, \{\mathbf{D}_K\}_{K\in\text{neighborhood of monomer } I \text{ and } J}\right) \tag{3}$$

ダイマーの電子状態計算も，モノマーの場合と同様に，ダイマーのエネルギー，密度行列は，ダイマーを構成する 2 つのモノマー I ， J と，その近傍にある密度行列の関数である．FMO 法では，式(2)，(3)で得られたモノマー，ダイマーのエネルギー，密度行列，および，式(1)を用いて，分子全体の電子状態を求める．

FMO 法では計算精度を犠牲にすることなく計算時間を削減するために，ダイマーの電子状態計算に対する近似を行う．この近似では，ダイマーを構成する 2 つのモノマー間の距離が離れている場合に，繰り返し計算が必要で計算負荷の大きな Self-Consistent Field (SCF)計算を行わずに，2 つのモノマーのエネルギーと，モノマー間の静電相互作用で，ダイマーの電子状態を近似する．

$$\begin{aligned} E_{IJ} &\leftarrow f\left(E_I, E_J, \mathbf{D}_I, \mathbf{D}_J\right) \\ \mathbf{D}_{IJ} &\simeq \mathbf{D}_I \oplus \mathbf{D}_J \end{aligned} \tag{4}$$

この近似のことを**ダイマーES 近似**，この近似を行うダイマーのことを**ES ダイマー**と呼ぶ．一方，負荷の重い SCF 計算を行うダイマーを **SCF ダイマー**と呼ぶ．ダイマーES 近似を適用することで，ES ダイマーを構成する 2 つのモノマー I ， J の密度行列のみを用いてダイマー電子状態計算を行うことができる．**図 1** に FMO 計算の概略を示す．計算のはじめに，モノマー電子状態計算で必要となるモノマー密度行列の初期値を計算する．その

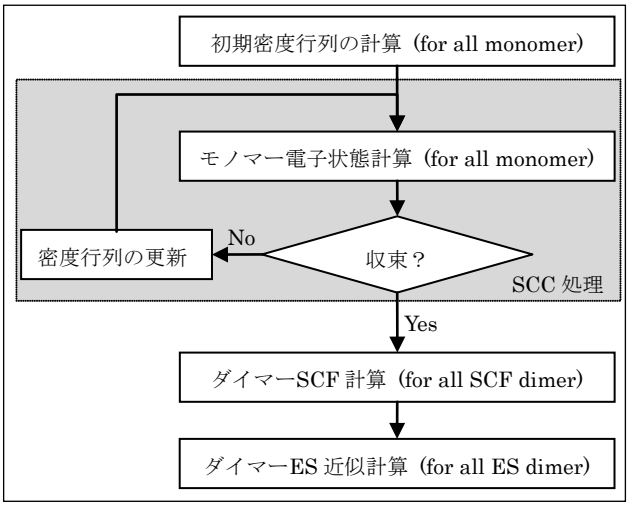

図 1 ：FMO 計算の流れ

密度行列を用いて，各モノマーの電子状態計算を行い，エネルギーや新たな密度行列を求める．この操作を，得られた密度行列と与えた密度行列とが一定の収束条件を満たすまで繰り返す．モノマーの密度行列が収束したら，その結果を用いて，ダイマーの電子状態計算（ダイマーSCF 計算，ダイマーES 近似計算）を行い，(1)式を用いて分子全体の電子状態（エネルギー，密度行列）を計算する．

【並列 FMO プログラムの基本構造】

前述のように，FMO 法は複数のモノマー（ダイマー）のフラグメント電子状態計算を並列に処理する粗粒度並列化と，各フラグメント電子状態計算自身を並列処理する細粒度並列化を行うことが容易であるため，2 段階並列処理向きの計

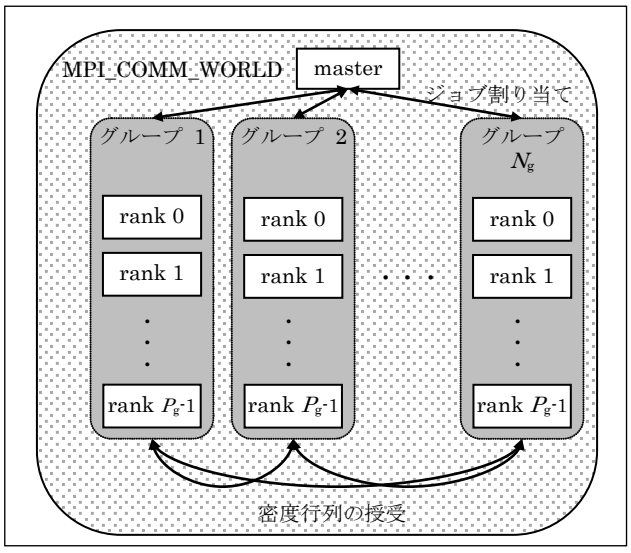

図 2 ：並列 FMO プログラムの基本構造

算手法である．並列 FMO プログラムの基本的な構造を，MPI での並列化した場合を例に**図２**に示す．MPI プロセスは，１つが FMO 計算の制御を行うマスタープロセスとなり，残りが各モノマー，ダイマーに対するフラグメント電子状態計算を行うワーカープロセスになる．ワーカープロセスは複数（図では N_g 個）のグループに分けられる．このグループ単位で，モノマー，ダイマーのフラグメント電子状態計算を行う（粗粒度並列処理）．各グループには複数（図では P_g 個）のプロセスが含まれており，マスタープロセスに割り当てられたフラグメント電子状態計算を P_g プロセスで並列処理する（細粒度並列化）．このように，並列 FMO プログラムは 2 段階並列化されており，かつ，マスター - ワーカー型の実行スタイルになる．ただ，注意しなければならないのは，各グループ間でモノマー密度行列データの授受が必要となるため，純粋なマスター - ワーカー型プログラムではない点である．

これまでの最適化により，FMO 計算で行うフラグメント電子状態計算部分の細粒度並列化における負荷均等化，および，グループ間で効率的にモノマー密度行列データにアクセスするための最適化を行い，各部分については高い性能が得られることを既に報告した．また，中間報告では，これらの最適化を行ったプログラムを実際に数万並列で実行して，大規模並列時には，各フラグメント（モノマーやダイマー）の電子状態計算の並列化性能の低下が問題にあることを示した．今回は，その点の改善に主眼を置いた OpenFMO プログラムの最適化を行った．

【フラグメント電子状態計算部分の改良】

FMO 計算における細粒度並列部分である各フラグメント（モノマー，ダイマー）に対する小規模電子状態計算の並列化効率の低下が FMO 計算全体の並列性能に大きく影響していること中間報告で示した．図３がその結果であるが，計算量の 99%以上を占める分子積分計算部分は 256 並列時で 94%という高い並列化効率を達成しているが，SCF 計算部分の並列性能が非常に低いため，電子状態計算全体の並列化効率が 256 並列時で 80%弱まで低下することが分かった．そこで，今回，SCF 計算部分の並列性能の改善を図ることにした．

図４に小規模電子状態計算で行う SCF 計算の手順を示す．SCF 計算では，まず，1 電子ハミルトン行列，および，重なり行列（ともに，1 電子積分計算），および，密度行列の初期値を計算する．次に，与えられた密度行列と 2 電子積分を基にして，2 電子ハミルトン行列（Fock 行列の密度行列依存部分．以降，G 行列）を計算する．この 2 電子積分は計算量が $O(N^4)$ と大きく，得られる値の数もそれに比例する．したがって，計算機の規模が小さい場合には主記憶に保存できないため，SCF 計算の

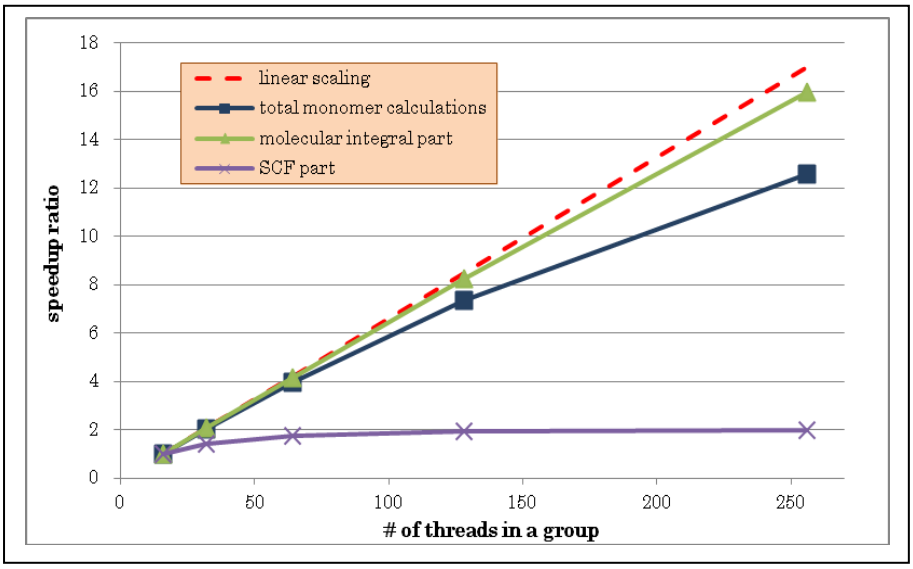

図３：モノマー電子状態計算部分の速度向上比

繰り返し計算のたびに計算する（Direct SCF 法）か，一部を保存して再利用して残りは毎回計算する方法（Partially Direct SCF 法），あるいは，２次記憶（HDD など）を利用する方法などが利用されている．しかし，今回考慮している大規模並列処理の場合には，SCF 計算利用できる計算機資源，特に，主記憶の総容量が大きくなるため，膨大な２電子積分を１度計算して主記憶に保存し，繰り返し計算で再利用することが可能である．そのため，大規模並列処理時には，２電子積分の繰り返し計算が不必要となり，G 行列計算の処理時間は大幅に短縮される．

G 行列計算が終わると，１電子ハミルトン行列，および,G 行列を用いた Fock 行列の合成,および，エネルギー計算を行い，その後，Fock 行列と重なり行列を用いた一般化固有値問題を解く．さらには，この固有値問題で得られた固有値（MO 係数行列）を使って，新たに，密度行列を作成する．得られた密度行列と与えた密度行列（あるいは，エネルギー値）がある閾値未満の差で一致（収束）したら，SCF 計算は終了する．収束していない場合には,密度行列を新たに得られたもに更新して，再び，G 行列計算以下を行う．

この SCF 計算では，G 行列計算部分をプロセス間並列処理しており，計算量と通信コストの兼ね合いから，他の部分は，スレッド並列のみを適用している．G 行列計算のプロセス並列部分では，まず，各プロセスが，自身が保存している２電子積分と密度行列を用いて G 行列の一部を計算して，全プロセスで計算した結果を Reduction 処理することで完全な G 行列を求めている．OpenFMO プログラムでは，この Reduction 処理において，これまで，MPI_Allreduce 関数を用いていた．その理由は，G 行列計算以外の処理（一般化固有値問題求解を含む）をプロセス並列していないため，全プロセスで同じ計算を行っても並列性能に影響しないと考えられること，ならびに，次の繰り返し計算における G 行列計算で必要となる密度行列をすべてのプロセスが計算するため，G 行列計算の際に密度行列の broadcast 処理が不要になること，

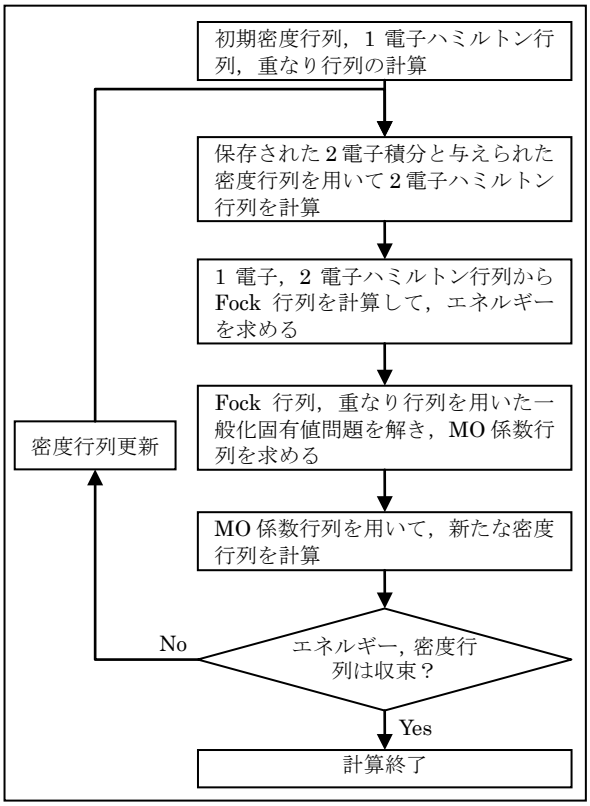

図４：SCF 計算のフローチャート

による．しかしながら，MPI_Allreduce は MPI_Reduce に比べて，プロセス間の処理時間のばらつきの影響を受けやすいと考えられる．また，プロセス並列を行っていない処理の部分は，全く同じ処理を行っているとはいえ，OS ジッタなどの影響による計算終了時刻のばらつき発生も否定できない．そこで，G 行列計算における Reduction 処理に MPI_Reduce 関数を用いて,そのルートプロセスで新たな密度行列計算までのプロセス並列非適用部分の処理を行い，得られた密度行列データを broadcast する，という方法に変更することにした．

この変更の適用前，および，適用後のコードを用いて，モノマー密度行列計算部分の並列性能の評価を行った．評価には，九州大学情報基盤研究開発センターの高性能演算サーバー FUJITSU PRIMERGY CX400 （8core Xeon(model E5-2680, 2.7GHz) × 2/node, 1476 nodes, Interconnect=Infiniband FDR)を用い，コンパイラは富士通 C コンパイラ(バージョン 1.2.0),MPI は富士通 MPI（バージョン 1.2.0）,また，数値演算ライブラリとして SSL-II を利用した.入力デー

タとしては，アデノウィルス KNOB ドメイン（PDB ID=1nob，16764 原子）で，2 アミノ酸残基を１フラグメント（全体で 576 フラグメント）として計算を行い，分子軌道の展開に用いる基底関数には，量子化学計算で一般に用いられている 6-31G(d,p)（142,974 関数）を使用した．８スレッド並列に固定して，各モノマー電子状態計算を行うプロセス数を４～６４まで変化させた（スレッド数は３２～５１２）．

その結果を表１に示す．この表には，変更適用前（Allreduce）と適用後（Reduce+Bcast）の 32 プロセス，64 プロセス並列時の並列化効率（4 プロセス並列時を１としている）．また，この表には，SCF 計算だけでなく，FMO 計算で行うモノマー電子状態計算に現れる各種分子積分計算部分を含んだ，全体の並列性能である．この結果からは，MPI_Reduce と MPI_Bcast を組み合わせて用いたほうが MPI_Allreduce を用いた場合に比べて，わずかではあるが，並列性能が向上していることが分かった．もともと，モノマー電子状態計算では分子積分計算が 99%以上の計算量を占めているため，SCF 計算部分の一部を変更しただけでは，ほとんど性能に影響しない可能性もあると考えていたが，わずか（64 プロセス並列時で約 0.77 ポイント）とはいえ，並列性能向上が見られたことは，プロセス間処理のばらつきに対する MPI_Reduce 関数の感受性の低さ（すなわち，性能悪化のしにくさ）の表れであると考えている．

表１：モノマー電子状態計算部分の並列化効率

	32 プロセス	64 プロセス
Allreduce	91.89	81.60
Reduce+Bcast	92.25	82.37

(2) 当初計画の達成状況について

一昨年度，昨年度は，FMO 計算の特定部分（細粒度並列部分，および，グループ間データ共有手法）に注目して最適化を行って来たが，今年度は，FMO 計算全体での並列性能低下要因の解析，および，更なる性能向上を目指した最適化を行うことを目標とした．中間報告までに，OpenFMO プログラムの超並列実行時にどのような問題が生じるのかを確かめるため，OpenFMO の大規模並列実行を行い，その性能を評価し，その結果，一昨年度行った細粒度並列処理部分の負荷均等化により，モノマー電子状態計算における分子積分計算部分の並列化効率が 256 並列時に 94%という非常に高い並列性能を達成することができていることを確認した．その一方で，小規模並列時には膨大な分子積分計算に隠れていた対角化などの逐次計算部分の計算時間が大規模並列時に目立つようになり，フラグメント電子状態計算全体の並列性能に大きく影響することが分かり，現状のプログラムでは，細粒度並列処理（フラグメント電子状態計算）部分が 100 並列を超えると，並列性能が低下することが分かった．また，中規模タンパク分子を用いた大規模（2 万）並列時の FMO 計算全体の性能評価を行い，細粒度並列部分の並列化効率がグループあたりスレッド数 128 の場合に約 87%であったことと，2 万並列時の粗粒度並列性能（並列化効率 93%）の結果から，全体で約 80％の高い並列化効率で FMO 計算が実行できていることが分かった．また，2 万並列時に 576 フラグメントの分子に対する HF/FMO 計算が 30 分程度で計算できることも確認した．

中間報告以降には，並列性能を低下させている SCF 計算部分の並列性能向上を目的として，内部で用いている通信部分の処理を変更し，わずかではあるが，並列性能が向上することを確認した．

昨年度までに多くの最適化を行っていたこともあり，今年度適用できた最適化は，SCF 計算の通信部分の変更だけであった．しかしながら，これまでの最適化により，数万並列での効率的な FMO 計算を行うことが可能になった．したがって，当初計画は，ほぼ達成したと考えている．

（参考文献）

1) K. Kitaura et al., Chem. Phys. Lett., 312, 319-324 (1999)
2) K. Kitaura et al., Chem. Phys. Lett., 313, 701-706 (1999)

3) T. Nakano et al., Chem. Phys. Lett., 318, 614-618 (2000)
4) M.W.Schmidt et al., J. Comput. Chem., 14, 1347-1363 (1993)
5)戦略的革新シミュレーションスフとウェアの研究開発「タンパク質 - 化学物質相互作用マルチスケールシミュレーション」
http://www.ciss.iis.u-tokyo.ac.jp/rss21/theme/life/synergy/index.html
6) T. Ikegami et al., SC|05 Seattle, WA, technical paper, 2005
7) OpenFMO homepage
http://www.openfmo.org/OpenFMO/index.html
8) J. Maki et al., Proc. HPCAsia07, 137-142(2007)

4. 今後の展望

我々は，超並列計算機を用いた大規模生体分子に対するハイスループット量子化学計算を実現するために，並列 FMO 計算プログラム OpenFMO に対して様々な最適化を行ってきた．その結果，数万並列での効率的な FMO 計算を実現し，2 万並列時には，1 万 6 千原子の大規模分子に対する FMO 計算を 30 分未満で行えることを示した．

今回利用しているプログラムは，近似の粗い（計算精度の低い）Hartree-Fock 法に基づいた FMO 計算に特化したコードであるが，より計算精度が高いと考えられている Korn-Sham 方程式に基づいた密度汎関数（DFT）法も Hartree-Fock 法と同様の並列化が適用できるため，DFT 法に基づいた FMO 計算も同様の並列性能が期待される．したがって，DFT 法ベースの FMO 計算を用いることで，大規模生体分子に対する高精度な量子化学計算を，超並列計算機を用いることにより，短時間で行える可能性が見えてきた．

並列化できない部分の処理を，他の処理と重ねて実行することにより更なる処理効率の向上も期待できることから，スパコン利用による生体分子に対する超高速第一原理電子状態計算が身近に実行できるのも，そう遠い将来ではない．それが可能になれば，スパコンを用いたコンピュータシミュレーション支援が，効率的な創薬に貢献できると考えている．

5. 研究成果リスト

(1) 学術論文（投稿中のものは「投稿中」と明記）

1. Yuichi INADOMI, Jun MAKI, Hiroaki HONDA, Toshiya TAKAMI, Taizo KOBAYASHI, Mutsumi AOYAGI and Kazuo MINAMI, "Performance Tuning of Parallel Fragment Molecular Orbital Program (OpenFMO) for Effective Execution on K-computer", Journal of Computer Chemstry, Japan, accepted.

2. 稲富雄一，眞木淳，本田宏明，高見利也，小林泰三，南里豪志，青柳睦，南一生，並列 FMO プログラム OpenFMO におけるモノマー密度行列データ保存方法，Journal of Computer Chemstry, Japan, 投稿中

(2) 国際会議プロシーディングス

なし

(3) 国際会議発表

1. Yuichi Inadomi, Jun Maki, Hiroaki Honda, Toshiya Takami, Tsuyoshi Nanri, Mutsumi Aoyagi and Kazuo Minami, "Performance Tuning of FMO Code (OpenFMO) for Effective Massively Parallel Execution", Theory and Applications of Computational Chemistry(TACC-2012), 2-7, Sep. 2012, Pavia, Italy. ポスター

2. Yuichi Inadomi, "Performance improvement of FMO program for effective massively parallel execution on K-computer", International Workshop on Massively Parallel Programming Now in Molecular Science, 28, Jan. 2013, 招待講演

(4) 国内会議発表

1. 稲富雄一，眞木淳，本田宏明，高見利也，小林泰三，青柳睦，南一生，「並列 FMO プログラム OpenFMO の高性能化」，ポスター 2P108，第 6 回分子科学討論会，2012.9.18-21，東京大学

2. 稲富雄一，眞木淳，本田宏明，高見利也，小林泰三，南里豪志，青柳睦，南一生，「並列 FMO プログラム OpenFMO の高性能化」，ポスター1P15，日本コンピュータ化学会秋季年会 2012，

2012.10.13-14, 山形大学

(5) その他（特許，プレス発表，著書等）

なし