Vol.2014-HPC-145 No.28
2014/7/29

Fiberミニアプリの性能評価

小村 幸浩[1,a)]　鈴木 惣一朗[1,b)]　三上 和徳[1,c)]　滝澤 真一朗[1,d)]　松田 元彦[1,e)]　丸山 直也[1,f)]

概要：本稿では次世代のスーパーコンピューティング実現のためのミニアプリ集である Fiber について紹介する．Fiber は理化学研究所および東京工業大学による将来 HPCI のあり方の調査研究の一環として開始されたプロジェクトであり，現在は理化学研究所を中心として継続して整備が進められている．本稿では Fiber ミニアプリについて概要を紹介し，一部についてはその性能の詳細について性能モデルを用いた議論を行う．

1. はじめに

アーキテクチャとアプリケーションのコデザインを推進するためにミニアプリやプロキシアプリと呼ばれる簡略化されたアプリケーションの整備が行われている [1], [2]．次世代スーパーコンピュータの実現に向けてコデザインによる設計，開発の重要性がうたわれているが，実利用に供されているアプリケーションは一般に非常に多数の機能を有した複雑な構成となっており，そのような複雑なアプリケーションの要求をアーキテクチャ設計に反映させることは必ずしも容易ではない．また，今日ではオープンソースとして開発されているアプリケーションも多く存在するが，一部には配布が制限されているアプリケーションも依然として存在し，そのような制限を有したアプリケーションを用いたコデザインを進めることは極めて困難である．ミニアプリもしくはプロキシアプリやスケルトンアプリ[*1]と呼ばれる簡略化されたアプリケーションは，見通しの良い，かつ利用制限の緩いアプリケーションを提供することにより，アプリケーションおよびアーキテクチャの相互理解の促進を目的としている．

我々は，理化学研究所および東京工業大学にて実施した「将来の HPCI システムのあり方の調査研究アプリケーション分野」（以下，アプリ FS と呼ぶ）の一環としてミニアプリ集 Fiber の整備開発を進めてきた．アプリ FS は 2018〜2020 年頃の社会的・科学的課題解決を担うアプリケーションの調査を国内の数多くの計算科学研究者による協力の下 2012 年度から 2013 年度にかけて実施したものであり，その調査結果は「計算科学ロードマップ」（以下，ロードマップと呼ぶ）としてまとめられた [3]．ロードマップには各計算科学課題に関する議論がまとめられており，またそれらを遂行するために必要なアプリケーションおよびアーキテクチャに対する必要な性能の概算が示されている．Fiber はロードマップにて提示されたアプリケーション群の一部に由来するミニアプリ集であり，現状 11 個のミニアプリから構成され，そのうち一部はすでにホームページからダウンロード可能である [4]．

本稿では現在整備がほぼ終了しているミニアプリについてその概要を紹介する．またそれらのうち CCS QCD, NGS Analyzer Mini，FFVC Mini についてその性能の詳細をモデル化を中心に説明し，ロードマップに提示された要求性能の実現可能性について論じる．

2. ミニアプリ概要

2.1 CCS QCD

ミニアプリ CCS QCD [5][6] は，高エネルギー物理学で用いられる格子量子色力学（格子 QCD）計算における，最も計算コストがかかるクォーク伝搬関数の計算部部分を抜き出したものである．CCS QCD では，Wilson 型の作用を用い，クォーク伝搬関数計算を 4 次元（空間 3 次元＋時間 1 次元）格子上の大規模疎行列の連立 1 次方程式を解く問題に帰着させている．係数行列は 4 次元格子上の 1 階差分の形になる．CCS QCD では，この連立 1 次方程式を，red/black ordering により前処理をした係数行列に対して，BiCGStab 法により解いている．プログラムは Fortran90 で記述され，並列化は，空間 3 次元に対する MPI 領域分

[1] 理化学研究所 計算科学研究機構, RIKEN AICS
[a)] yukihiro.komura@riken.jp
[b)] soichiro.suzuki@riken.jp
[c)] kazunori.mikami@riken.jp
[d)] shinichiro.takizawa@riken.jp
[e)] m-matsuda@riken.jp
[f)] nmaruyama@riken.jp

[*1] 用語として「ミニアプリ」「プロキシアプリ」「スケルトンアプリ」等が用いられるが本稿では特に断りの無い限りすべてミニアプリで統一する．

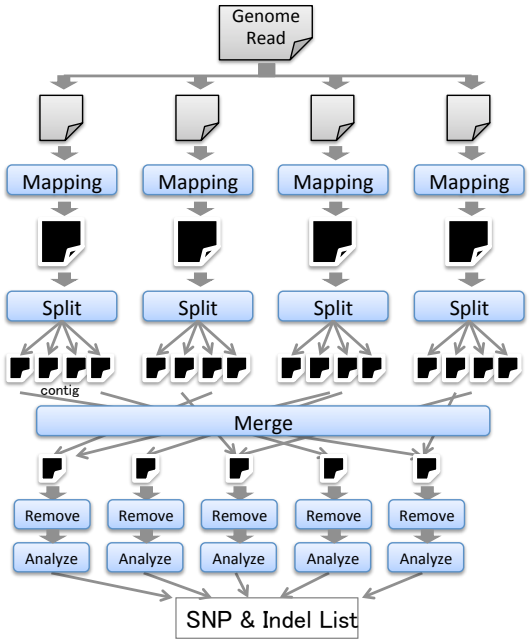

図 1 NGS Analyzer のワークフロー

割と OpenMP スレッド並列を行っている．コメント行を除いたソースコードサイズは約 1000 行である．

2.2 NGS Analyzer Mini

エクサスケールシステムでは，ゲノムデータや X 線自由電子レーザー施設 SACLA による観測データの解析等，ビッグデータ解析の需要も見込まれている．そこで Fiber ミニアプリでも，大規模な IO を行うアプリケーションの性能を評価するためのベンチマークを，ゲノム解析プログラムをベースに提供している．

Fiber では NGS Analyzer [7] をベースに，NGS Analyzer Mini を提供している．NGS Analyzer は次世代シークエンサーの出力データを高速に解析し，ヒト個人間の遺伝的差異やがんゲノムの突然変異を高い精度で同定するプログラムであり，その解析手法の詳細は日本人男性の全ゲノム解析を行った文献 [8] に記されている．

NGS Analyzer のワークフローを図 1 に示す．処理対象となる塩基配列 (Genome Read) を一定量の塩基配列で分割し，それぞれをマッピング処理し (Mapping)，参照配列中の位置を判定する．その後，コンティグ (連続する塩基配列パターン) 単位でマッピングデータを分割する (Split)．マッピングとその結果の分割は入力毎に独立した処理であり並列実行できるが，結果解析の前に，コンティグ単位で各マッピング結果をマージする必要がある (Merge)．マージは共有ファイルシステム上でのファイル IO として行われ，ノード間では通信を行わない．マッピング結果には重複が発生し得るので，マージ後マッピング結果をソートし，重複除去 (Remove) を行った後に，コンティグ単位で SNP，Indel 等の変異同定を統計的に行う (Analyze)．

NGS Analyzer Mini ではこのワークフローを実現する次の 2 種類のプログラムを提供する．

(1) 図 1 のワークフロー全体を一括実行するプログラム
(2) 図 1 のワークフロー中の IO を行う以下の 3 処理を個別に実行するプログラム
- Mapping（Split を含む）
- Remove（ソートを含む）
- Analyze

(1) は MPI 並列実行可能なシェルスクリプトとして，(2) は逐次実行するシェルスクリプトとしてそれぞれ提供している．後者はプロファイルを取得することを目的に提供している．

オリジナルの NGS Analyzer はオープンソースソフトウェア 2 種類含む，計 7 種類の C/C++プログラム，及び，それらを京コンピュータで実行するためのジョブ管理用シェルスクリプトからなる．NGS Analyzer Mini では 2 種類のオープンソースソフトウェアには手を加えず，オリジナル開発者が C/C++で実装したプログラムから不要コードを削除した．その結果，C/C++コードで 3,199 行から 2,696 行へと 16%削減した．また，京のジョブ管理用シェルスクリプトは削除し，代わりにそこで定義されていたワークフローを一括して実行するためのプログラム（上記 1）と，ワークフローにおいて，IO を行う処理を個別に実行するためのプログラム（上記 2）を実装した．これにより，748 行から 328 行へと 56%の削減となった．

2.3 FFVC Mini

ミニアプリ FFVC Mini のベースとなっているのは，熱流体解析プログラム FFV-C [9] である．FFV-C は，産業界における熱流体現象をシミュレートし，設計に必要な情報を提供することを目的としている．Navier-Stokes 方程式を直交等間隔格子上の有限体積法により離散化し，非圧縮流体として，時間ステップ毎に圧力 Poisson 方程式を反復法により解いている．

FFV-C には以下の特徴がある．反復計算部のメモリロードストアを減らすために，各格子位置での媒質情報，境界条件情報，隣接格子情報などを，ビット単位のフラグとして 32 ビット整数内にエンコードし，これらの配列として保持している．そのため，連立一次方程式の係数行列などは，実際の計算時にこれらのビット情報配列よりデコードされ使用される．また，境界形状情報を STL フォーマットのポリゴンデータとして読み込み，実行時に大規模格子を自動生成する機能を組み込んである．プログラム全体の制御は C++で記述されているが，ホットスポット部分は Fortran90 により書かれている．並列化は，空間 3 次元に対する MPI 領域分割と OpenMP スレッド並列を行っている．

ミニアプリ FFVC Mini では，オリジナルプログラム

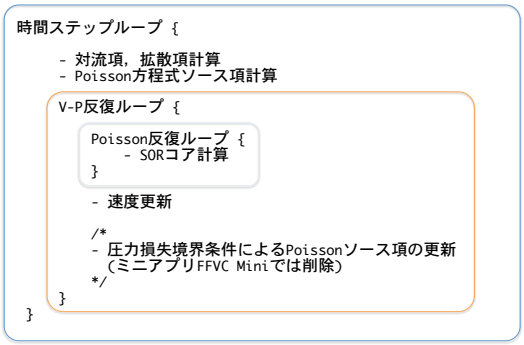

図 2 FFV-C および FFVC Mini のループ構造

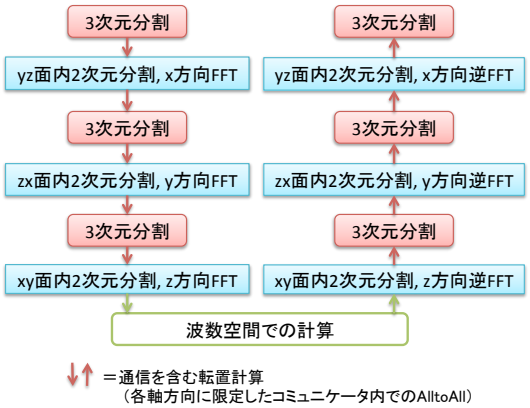

図 3 MARBLE における 3 次元 FFT 計算

FFV-C に対して，計算内容を 3 次元直方体領域内のキャビティフロー問題に特化している．また，使用アルゴリズムも，時間積分は 1 次精度 Euler 陽解法，対流項は 3 次精度 MUSCL スキーム，反復解法はストライドメモリアクセス型の 2 色 SOR 法に限定している．コメント行を除いたソースコードサイズは，FFV-C は 7 万 7 千行，FFVC Mini で 9 千行となっている．

図 2 に FFV-C および FFVC Mini のプログラムループ構造を示す．最も内側の Poisson 反復ループでは，圧力 Poisson 方程式を反復解法で解く．一方，V-P 反復ループは，速度の関数として与えられた圧力損失境界条件を Newton-Raphson 法的に扱うためのループである．そのため，圧力損失境界条件を使用しないミニアプリ FFVC Mini では，実際には V-P 反復の中身は 1 回実行すれば充分である．ただし，本ミニアプリでは，ベンチマークプログラムとしての使用を考慮して，V-P 反復，Poisson 反復とも任意の反復回数を指定して実行可能である．オリジナル FFV-C がターゲットとしている典型的な計算では，タイムステップあたり，V-P 反復は 5〜100 回，Poisson 反復は 20〜1000 回程度行われる．

2.4 MARBLE Mini

MARBLE をベースに，古典分子動力学ミニアプリ MARBLE Mini を作成した．

MARBLE [10] は，生体高分子系を計算対象とした分子動力学 (MD) シミュレーションプログラムである．クーロン相互作用計算には Particle Mesh Ewald (PME) 法を採用している．また，シンプレクティック部分剛体時間積分法により，精度の高い全エネルギーの保存を実現している．プログラムは，MPI と OpenMP によるハイブリッド並列化がなされている．MARBLE は，PME 法に 3 次元 FFT を用いているため，大規模実行時に全対全通信がスケーラビリティ上の問題となりうるが，エクサスケールでは創薬計算などにおける 10 万原子規模の系に対するアンサンブル計算での応用が期待されており，全対全通信のコストは限定的と推定できる．MARBLE は C 言語により記述されている．

生体高分子系の古典 MD では，通常，少数のイオンを含む水中でのタンパク質あるいは DNA/RNA などの運動を計算する．そのため，その構成原子数の違いから，計算時間のほとんどは，本来の計算対象である生体高分子ではなく，水分子間の相互作用に関連した計算に費やされる．このことから，ミニアプリ MARBLE Mini では，計算対象を水分子系に限定し，それ以外のコード（生体高分子内の共有結合計算など）を削除してある．また，アンサンブルもエネルギー一定のミクロカノニカル計算のみに限定している．コメント行を除いたソースコードサイズは，MARBLE は 3 万 2 千行，MARBLE Mini では 7 千行となっている．

MARBLE および MARBLE Mini の主な計算部分は，原子間相互作用の短距離成分に対する二体力計算部分と，長距離成分に対する PME 部分からなる．二体力計算は，空間を矩形領域（セル）に分割し，計算対象となる 2 原子の所属するセル対単位で行っている．この部分の並列化は，MPI による 3 次元領域分割，セル対のリストに対する OpenMP スレッド並列を行っている．MPI プロセス間の通信は，隣接ノード間で袖領域の原子に対して，二体力計算前に座標値データの通信，二体力計算後に力データの通信を行う．PME 部分での 3 次元 FFT は，現在のコードでは図 3 のように実装している．内部で使用する 1 次元 FFT 計算部分については FFTW ライブラリのルーチンを用いている．コンパイル時のオプション指定により，1 次元 FFT ルーチンを OpenMP スレッドから並列に呼び出す実装と，OpenMP 並列化された多重 1 次元 FFT ルーチンを呼び出す実装が切り替え可能である．ミニアプリ MARBLE Mini では，将来 3 次元 FFT 実装を入れ替え可能とするために，その部分のモジュール化を実施してある．

オリジナル MARBLE では，領域分割により MPI 並列がなされているが，原子に付随する一部の情報に対するデータ分割がなされていない．ミニアプリ MARBLE Mini でもこの点は変わらず，各ノードにおいて全原子数に比例するデータが複数存在したままになっている．これら未分割データは，使用メモリ量の増加のみではなく，計算性能

への影響も見られるため，今後の改善を進めていく予定である．

2.5 MODYLAS Mini

MODYLAS をベースに，古典分子動力学ミニアプリ MODYLAS Mini を作成した．

MODYLAS [11][12] は，汎用の古典分子動力学シミュレーションプログラムである．クーロン相互作用計算には Fast Multipole Method (FMM) 法を採用している．FMM 法は，多重極展開によるクーロン相互作用の計算を八分木構造を利用してシステマチックに行う方法で，数万ノードを越える大規模並列計算においても高いスケーラビリティが実現可能である．このことからエクサスケールでは，エンベロープを持ったウィルスの全原子シミュレーションなど，総原子数が 1 億を越えるような巨大系への応用が期待されている．MODYLAS は Fortran90 で記述され，MPI および OpenMP による並列化がなされている．

MODYLAS Mini では，MARBLE と同様に生体高分子系での計算性能評価を前提として，計算対象を水分子系のミクロカノニカル計算に限定している．コメント行を除いたソースコードサイズは，MODYLAS は 3 万行，MODYLAS Mini では 8 千行となっている．

MODYLAS および MODYLAS Mini の主な計算部分は，原子間相互作用の短距離成分に対する二体力計算部分と，長距離成分に対する（二体力計算部分を除いた）FMM 計算部分からなる．二体力計算は，空間を矩形領域（セル）に分割し，計算対象となる原子の所属するセル単位で行っている．メモリアクセスを連続にしキャシュ利用効率を高めるために，独自の格納形式とループ構造を採用している．この部分の並列化は，MPI による 3 次元領域分割，セルのループに対する OpenMP スレッド並列を行っている．MPI プロセス間の通信は，隣接ノード間で袖領域の原子に対して，二体力計算前に座標値データの通信を行う．MODYLAS Mini では，二体力計算に作用・反作用則を利用していないため，計算後の力データの通信は不要である[*2]．FMM 計算部分でも，二体力計算と同様な領域分割により MPI 並列可されている．そして，八分木の各レベルにおいて，多重極展開および局所展開の係数に関する計算は木構造のノード（スーパーセル）を単位にして行う．このスーパーセルに対するループを OpenMP スレッド並列化している．この部分の MPI 通信は，M2M（多重極展開の展開中心シフト）および M2L（多重極展開から局所展開への変換）計算時に参照される多重極展開係数データのプロセス間でのコピー通信である．これらの通信は，上位階層では隣接ノードを越えた近接ノード間での通信となるため，MODYLAS および MODYLAS Mini では（京/FX10 の Tofu ネットワークに最適化された）バケツリレー方式の通信ルーチンを実装している．

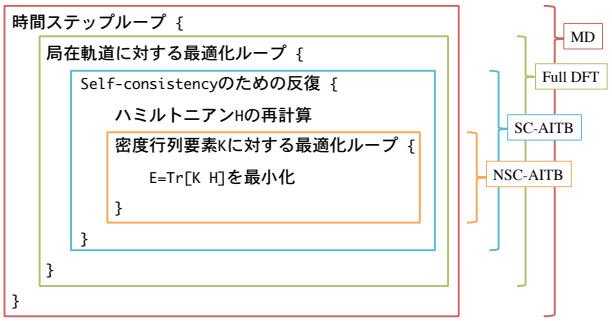

図 4 CONQUEST のループ構造

2.6 CONQUEST Mini

オーダ N 法第一原理計算プログラム CONQUEST をベースに，CONQUEST Mini を作成した．

CONQUEST [13] は，通常の第一原理計算プログラムとは異なり，Kohn–Sham 方程式の固有波動関数ではなく，密度行列を直接最適化計算で求めることにより電子状態計算を行っている．密度行列の非対角項にカットオフ長を導入することにより，計算量および使用メモリ量が電子数に比例するオーダー N 法を実現している．プログラムは Fortran90 で記述され，領域分割により MPI と OpenMP によるハイブリッド並列化がなされている．

CONQUEST では入力パラメータの指定より，図 4 に示すとおり，non-self-consistent *ab initio* tight binding (NSC-AITB) 計算，self-consistent *ab initio* tight binding (SC-AITB) 計算，full DFT (DFT) 計算，*ab initio* Molecular Dynamics (MD) 計算が可能である．ミニアプリ CONQUEST Mini では，この内，NSC-AITB 計算に特化している．また，弱スケーリング計測でのノードあたりの計算量を一定に保つために，密度行列要素に対する最適化反復回数を一定に固定するベンチマークモードをオプションとして追加している．コメント行を除いたソースコードサイズは，CONQUEST は 6 万 4 千行，CONQUEST Mini では 2 万 6 千行となっている．

CONQUEST および CONQUEST Mini のプログラム内部では，疎行列データを多用している（大雑把に説明すると，原子 i と原子 j の相対距離がカットオフ長以内の場合のみ行列要素 M_{ij} は非ゼロになる）．実行時のホットスポットなるのは，これら疎行列同士（行列毎にカットオフ長や非ゼロ要素パターンは異なる場合がある）の積計算である．

2.7 NICAM-DC

ミニアプリ NICAM-DC[14] は，全地球規模での気象現

[*2] オリジナル MODYLAS では，小〜中規模系での計算も考慮して，コンパイル時のオプション指定により作用・反作用則を用いた計算も可能．

象をシミュレーションする大気大循環モデルのアプリケーションのひとつである NICAM[15] をベースにしている．NICAM の計算内容は大きく分けると，流体力学的な計算が主となる力学過程と，放射・乱流・雲などを扱う物理過程からなる．物理過程は，要求 B/F 値の低い計算が多く，また水平格子方向に依存性がないため 2 次元領域分割により並列化した場合にプロセス間通信が発生しない．NICAM-DC は，NICAM から，より要求 B/F 値が高く，通信回数の 8 割が集中している力学過程 (Dinamical Core) のみを抜き出したミニアプリとなっている．コメント行を除いたソースコードサイズは，NICAM は 24 万 5 千行，NICAM-DC では 3 万 5 千行となっている．

NICAM-DC では，地球大気の運動を静水圧近似を行わない Navier-Stokes 方程式で記述し，それを球殻上の三次元格子を用いて有限体積法により離散化して解いている．水平方向の格子は，正二十面体を構成する正三角形要素を再帰的に分割していくことにより得られる全球で一様な三角格子を採用している．有限体積法のコントロールボリュームの水平断面は，正二十面体の頂点となる格子点では 5 角形，その他の格子点では 6 角形となる．時間積分は，水平方向は Runge-Kutta 法により陽解法で，鉛直方向は陰解法として 1 次元 Helmholtz 方程式を離散化した三重対角行列を係数に持つ連立一次方程式を解くことにより進めている．

NICAM-DC は水平方向に領域分割すること により MPI 並列化されている．領域分割は，まず正二十面体を構成する正三角形要素の隣り合った 2 要素を合わせて 10 個の菱型領域を作成する．この菱型領域を 4 等分する操作を再帰的に数回繰り返してできる菱型領域を単位として，各プロセスに計算領域を割りふっている．

NICAM-DC は Fortran90 により記述されている．

現在，京コンピュータにおいて，水平解像度 870m（3 次元格子数 700 億）でのシュミレーションが可能である．エクサスケールでは，水平解像度 220m（3 次元格子数 1 兆）での計算を目指している．

2.8 ALPS/looper

現在，量子モンテカルロシミュレーションプログラム Alps/looper[16][17] をベースとしたミニアプリの整備を進めている．

Alps/looper は，連続虚時間ループアルゴリズム量子モンテカルロ法により，量子スピン模型を空間 d 次元，温度の逆数に対応した虚数時間 1 次元の $d+1$ 次元古典系に焼き直してシミュレーションを行っている．グラフアルゴリズム (union-find) によるクラスター認識処理を多用し，リンクリスト操作や整数演算が主体であり，また条件分岐も多い．

Alps/looper は C++で記述され，並列化は，虚数時間方向に MPI 並列，空間方向に OpenMP スレッド並列がなされている．

2.9 mVMC

現在, 多変数変分シミュレーションプログラム mVMC[18] をベースとしたミニアプリの整備を進めている．

mVMC は，強相関電子系での物理量の基底状態期待値を計算するアプリケーションである．mVMC は C で記述され，MPI と OpenMP により並列化されている．

3. ミニアプリの性能およびモデル化

3.1 NGS Analyzer Mini

エクサスケールシステムが完成する 2020 年では，個人ゲノム解析は次の問題規模の計算を抱えている．

ヒト一人あたりのゲノムデータ量 現在の 100 倍，およそ 100TB.

解析対象ゲノム数 200,000 ゲノム.

解析時間 3 年，または 5 年での全解析完了を予定．1 ゲノムあたり，2520 秒程度の時間での解析を計画.

個人ゲノム解析は実行時間により問題が規定されているため，本章では実行時間のモデル，及び要求 IO 性能のモデルについて議論する．

3.1.1 大規模実行時の性能予測

2.2 章で述べたミニアプリ「(1) ワークフローの一括実行」の実行時間の予測を，ミニアプリ「(2) Mapping, Remove, Analyze の個別プログラム」を用いて行う．(2) の 3 処理を行うプログラムを京コンピュータ上で実行し，プロファイルを取得し，性能の外挿を行った．

Mapping の入力として，解析対象データは日本人初の全ゲノム解析にて用いられたゲノムデータの 1 データセット（以降 DRR000617 と表記）[*3]の一部の 25 万塩基配列 (60MB × 2)，参照配列としては NCBI にて公開されているヒトゲノムデータとした (6.3GB)[*4]．Remove，Analyze に関しては，処理対象となるコンティグ毎に入力サイズが異なり，実行時間は入力サイズが最大のコンティグに依存するため，DRR000617 全体を処理した際に生成された，最大サイズの入力を用いた．Remove は 2.5GB の入力，Analyze は 510MB と参照ゲノム 3GB の入力とした．プロファイルとして，実行時間，実行時間に占める IO の割合（ジョブ統計情報に含まれる，システム CPU 時間とした），Read/Write サイズを取得した．結果を表 1 に示す．

これをもとに実行時間のモデルを構築し，DRR000617 全体を 546 並列で実行したときの実測時間との比較を行う．546 は入力を 25 万塩基配列の組に分割したときの組数であり，1 ノード 1 組の Mapping 処理を行うよう並列数

[*3] http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=viewer&m=data&s=viewer&run=DRR000617
[*4] ftp://ftp.ncbi.nlm.nih.gov/genomes/Homo_sapiens/

表 1 京における NGS Analyzer Mini プロファイル結果

項目	Mapping	Remove	Anayze
実行時間 (sec)	218.29	1318.93	1127.43
IO の割合	0.04	0.01	0.03
Read サイズ (GB)	8.07	3.51	8.63
Write サイズ (GB)	0.33	1.67	5.31

を指定している．このとき，生成されるコンティグは 402 個であった．Remove, Analyze は 402 並列で実行されることになる．実測時間は 4,181 秒であった．各ステップ同期的に実行されるとし，ワークフロー全体の実行時間予測モデルは次の通りに定義する．

$$T_{Workflow} = T_{Mapping} + T_{Merge} + T_{Remove} + T_{Analyze} \quad (1)$$

計算を行うフェーズである，Mapping，Remove，Analyzer の実行時間のモデルを次の通りに構築する．全て同じモデルを採用し，各変数にはフェーズ毎の値を当てはめて性能を予測する．プロファイルにより測定した実行時間を T，実行時間に占める IO の割合を p，Read，Write サイズをぞれぞれ $Size_r$，$Size_w$，IO 性能を IO_Thput，実行ノード数を N とすると，各フェーズの実行時間のモデルは次のように表すことができる．

$$T_{M,R,A} = T \times (1-p) + \frac{N \times (Size_r + Size_w)}{IO\_Thput} \quad (2)$$

NGS Analyzer では計算途中にノード間通信は行わないため, 各ノード一定の時間で処理を完了できるとし, $T \times (1-p)$ でモデリングする．一方，多数のノードで実行する際には IO にて競合が起こるため，IO 性能 IO_Thput はノード数に依存するモデリングを行う．

1 ノード利用時の実測 IO 性能 IO_Thput_1 は次の通りに表される．

$$IO\_Thput_1 = \frac{Size_r + Size_w}{T \times p} \quad (3)$$

ただ，これは IO 競合を考慮していないため補正する必要がある．京コンピュータではファイルシステムとして FEFS (Fujitsu Exabyte File System) を採用している．京コンピュータの FEFS では 192 ノードグループ毎に 6 個の OSS (Object Storage Server，ファイル実体を保存するサーバ) を持ち，IOR ベンチマークによる 192 ノード並列アクセスで Read で 2.5GB/s，Write で 2.0GB/s の性能を記録している．192 ノード並列アクセス時の Read 性能を R_Thput，Write 性能を W_Thput とすると，192 ノードグループの IO 性能 IO_Thput_{192} は次の通りに表される．

$$IO\_Thput_{192} = \frac{N \times (Size_r + Size_w)}{\frac{N \times Size_r}{R\_Thput} + \frac{N \times Size_w}{W\_Thput}} \quad (4)$$

IO 競合発生時にはこの IO 性能をノード間で均等分配すると想定すると，IO 性能 IO_Thput は次の通りに表すことができる．

$$IO\_Thput = Min\ (\ IO\_Thput_1,\ \lceil \frac{N}{192} \rceil \times \frac{IO\_Thput_{192}}{N}) \times N \quad (5)$$

次に Merge の実行時間モデルについて考える．(1) ワークフロー一括実行プログラムでは，ローカルファイルシステムから共有ファイルシステムへの移動，共有ファイルシステムからローカルファイルシステムへの移動，ローカルファイルシステム上でのファイルの結合，3 回の Read・Write を行っている．コンティグ数を N_{Contig}，Mapping の出力結果のサイズを $Size_{Mapout}$ とすると，Merge の実行時間モデル T_{Merge} は次の通りとなる．

$$T_{Merge} = 3 \times (\frac{N_{Contig} \times Size_{Mapout}}{\lceil \frac{N}{192} \rceil \times R\_Thput} + \frac{N_{Contig} \times Size_{Mapout}}{\lceil \frac{N}{192} \rceil \times W\_Thput}) \quad (6)$$

ここで，N_{Contig} は 402，$Size_{Mapout}$ はプロファイルより 175MB となる．

式 1 に式 2， 5， 6 を当てはめることにより，モデルが構築できる．表 1 のパラメータを当てはめると 4,407 秒となる．実測と 5%程度の誤差があるが，2 つの理由が考えられる．1 つはモデルでは FEFS の MDS (Meta Data Server, ファイルのメタデータを管理するサーバ) への負荷を考慮していない点である．特に Merge においては，多数のノードから MDS にアクセスが集中するため，モデルでは性能を高く見積もっている可能性がある．2 点目はノードのバッファキャッシュの影響である．プロファイル測定時には Remove，Analyze はそれぞれ別々のプログラムとして実行しているため，OS のバッファキャッシュの影響が含まれていない．一方，ワークフローを一括実行する場合には 2 つの処理は立て続けに実行されるため，バッファキャッシュの影響を受けうる．そのため，モデルでは性能を低く見積もっている可能性がある．

3.1.2 NGS Analyzer のエクサシステム要求性能への外挿

目標時間内に全ゲノムの解析を完了するときに要求される IO 性能を外挿する．

外挿は 3.1.1 節同様, Mapping, Remove, Analyze プログラムを用い，プロファイルは東京大学 FX10(Oakleaf-FX) にて取得した．入力データには 3.1.1 節と同じデータを用いた．ただし，Remove，Analyze においては，ワークフローの該当フェーズ実行中の平均的な要求 IO 性能を求めるために, DRR000617 データセットを処理した際に生成された平均サイズの入力を用いた．そのため，Remove では 320MB の入力，Analyze は 50MB の解析対象と 3GB の参照ゲノムの入力とした．プロファイル結果を表 2 に示す．

表 2 東京大学 FX10 における NGS Analyzer Mini プロファイル結果

項目	Mapping	Remove	Anayze
実行時間 (sec)	242.19	110.11	228.81
IO の割合	0.04	0.03	0.10
Read サイズ (GB)	8.07	0.32	3.34
Write サイズ (GB)	0.36	0.16	0.47

このプロファイル結果を基に外挿を行う．外挿の方針として，プロファイルを取得した FX10 とノード単体性能は同規模の性能のマシンにて，2020 年の想定問題を解くときの要求 IO 性能を求める．ただし，前節では IO アーキテクチャを考慮したモデリングを行ったが，本説では IO アーキテクチャに前提を設けない．各処理毎の要求 IO 性能を求め，その最大値を全体の要求性能とする，以下のモデルにて外挿を行う．

$$IO\_Thput_{Workflow} = Max(IO\_Thput_{Mapping}\ , \quad (7)$$
$$IO\_Thput_{Remove}\ ,$$
$$IO\_Thput_{Analyze}\ )$$

各フェーズにおける IO 性能を以下の通りにモデル化する．

$$IO\_Thput_{M,R,A} = \frac{Size}{T_{2020}} \times Num\_Files \quad (8)$$

$Size$ は 1 処理あたりの入出力サイズ，T_{2020} は 2020 年の要求実行時間，Num_Files は各フェーズにおいて処理するファイル数を表す．各フェーズにおいて，Num_Files 分の並列処理を要求すると仮定している．

前述の通り，2020 年には現在の規模の 100 倍の入力を要する．Mapping では 100TB のゲノムを入力とする．プロファイル同様 25 万塩基配列毎に分割処理すると，$Size$ は 8.43GB, Num_Files は 833,334 となる．Remove, Analyze ではコンティグ単位に処理される．コンティグ毎のデータ量が 100 倍になると仮定し，Remove，Analyze の $Size$ はそれぞれ平均的に 48GB，381GB となるとする．このときの Num_Files は現在のコンティグ数同様に 7,156 とする．T_{2020} は表 2 の現在の実行時間の比率に従い，均等分割すると Mapping，Remove，Analyze でそれぞれ 1,050.27 秒，477.51 秒，994.19 秒となる．IO 時間はそれぞれ 42.01 秒, 14.33 秒, 99.42 秒となる．

以上を式 8 に代入すると，Mapping は 167.22TB/s，Remove は 23.98TB/s，Analyze は 27.42TB/s となる．式 7 より，個人ゲノム解析で要求される IO 性能は Mapping の 167.22TB/s が最大となる．一方，HPCI 技術ロードマップ白書 [19] では，2018 年には 20MW 消費電力で実現できるシステムにおいてはストレージ性能は高々 10TB/s であると述べられており，大きな乖離がある．現状の方法ではエクサスケールシステムでは問題を解くことができないため，アルゴリズム，実装方法，実行方法，問題規模の変更が必要である．

例えば，次のような変更が考えられる．Mapping においては 2.2GB の参照ゲノムを 3 回繰り返し読み込んでいる．この読み込みを 1 回に減らすだけで，1 処理あたりのデータ量を 4.03GB に減らすことができ，要求性能は 79.94TB/s となる．それでも現在の実行方法では，解析対象のゲノム 120MB に対して，参照ゲノムのサイズは 3.48GB と大きい．Mapping ではデータの依存関係が無いため，1 処理あたりのデータ量を増やし，処理回数を減らして実行することも可能である．このとき，計算量，出力サイズは入力に比例して増加する．例えば，解析対象ゲノムを 2 倍の 240MB 単位で処理する場合，実行時間，出力サイズも簡単のため 2 倍になると仮定すると，1 処理あたりのデータ量は 4.51GB，処理回数は 416,667 回となり，要求性能は 31.69TB/s となる．

本エクサシステム性能要求においては，Merge フェーズの IO 性能要求は考慮していない．現状のミニアプリにて Merge の挙動を評価する術を提供していないためではあるが，Merge で行われている IO パターンを多数のノードで行うことは，ファイルシステム，特に MDS に高負荷をかけることになり望ましくない．代替手段として文献 [20] に提案されているように，オンメモリでのデータ転送で Merge 同等の処理を置き換えることが可能なので，このような手法を用い，IO 負荷を削減することが望ましい．

3.2 CCS QCD の性能評価

3.2.1 CCS QCD のエクサシステム要求性能

CCS QCD は Clover 部と BiCGStab 部の 2 つの計算部分で構成されている．CCS QCD のエクサシステム要求性能の 1 つとして，問題サイズ 256^4 格子で以下の性能を満たすことである [3].

- BiCGStab 部の実行時間が 3.1 msec/step 以下
- Clover 部の実行性能が BiCGStab の 20%以上

ここでは，CCS QCD の BiCGStab 部の性能評価モデルを作成し，エクサシステムで要求される性能を外挿する．

CCS QCD では 4 次元空間データを，空間 3 次元で分割し，MPI プロセスにマッピングする．BiCGStab 法のデータ通信パターンを以下に示す．

- 隣接空間を担当する MPI プロセス間での境界データを交換.
- 1 要素の AllReduce の全体通信.

性能評価モデルを作成するうえで，それぞれの通信回数と通信量のデータが必要となる．境界データの通信は sendrecv の同期通信を 1 ステップ毎に 12 回行い，1 通信の通信量は $192N_i$ byte である．ここでの N_i は $i = x, y, z$ 面の要素数であり，$N_x = NY \times NZ \times (1 + NT/2), N_y = NZ \times NX \times (1 + NT/2), N_z = NX \times NY \times (1 + NT/2)$, NX, NY, NZ, NT は 1 ノードが担当する領域である．一

方，AllReduce の全体通信は 2 ステップの間に 6 回行い，16 byte の通信を 3 回 8 byte の通信を 3 回行う．

3.2.2 1 ノードの性能評価モデル

性能評価モデル作成のために，東京大学情報基盤センターに設置されている FX10 を用いて通信のない 1 ノードでの実測 F/B 値を測定する．FX10 にはシステム付属のプロファイラ機能があり [21]，その機能を用いて，BiCGStab 部の FLOP とメモリスループット量を計測し，実測 F/B 値を測定する．1 ノードの問題サイズ $(NX, NY, NZ, NT) = (8, 8, 8, 256)$ での BiCGStab 部の実測 FLOPS は 39.0 GFLOPS であり、単位時間当たりの実測メモリスループット量は 61.3 GB/s である．この値より，実測 F/B 値は 0.64 であり，実測 F/B 値を用いたルーフラインモデルを以下に示す．

$$\text{perf} = \min(\text{FLOPS}, 0.64 * \text{Bandwidth}) \quad (9)$$

ここでの FLOPS は LINPACK ベンチマークの 1 ノードの CPU 性能値であり，Bandwidth は STREAM メモリベンチマークでのメモリバンド幅の値とする．FX10 でのそれぞれの値は FLOPS = 218 GFLOPS [22]，Bandwidth = 60 GB/sec [23] である．ルーフラインモデルより，CCS QCD はメモリバウンドのアプリであり，FX10 の性能値を代入した結果は 38.4 GFLOPS である．また，FX10 に付属しているプロファイラでの実測値は 39.0 GFLOPS であり，モデルより実測が上回っている．この原因はモデルでピークとした STREAM でのメモリバンド幅より高い性能が出たためだと考えれらる．

3.2.3 複数ノードの性能評価モデル

1 ノードの実測値をもとに複数ノードの性能モデルを作成する．1 ノードから複数ノードの拡張によって，追加される計算として，ノード間のデータ通信，袖通信のための送受信データコピーがある．ノード間のデータ通信として，袖通信と全体通信の 2 つがあり，データ通信にかかる時間のモデル式をそれぞれ作成する．1 度の袖通信にかかる時間 $T_{boundary}$ と全体通信にかかる時間 $T_{AllReduce}$ のモデル式を以下に示す．

$$T_{boundary} = l + 192N_i \times g \quad (10)$$

$$T_{AllReduce} = (l_{AllReduce} + s \times g) * \log_2(Proc) \quad (11)$$

l はネットワークレイテンシ，$l_{AllReduce}$ は AllReduce の 1 ステップのレイテンシ，s はデータ長，g はネットワーク通信性能の逆数，N_i は $i = x, y, z$ 面の要素数であり，$N_x = NY \times NZ \times (1 + NT/2), N_y = NZ \times NX \times (1 + NT/2), N_z = NX \times NY \times (1 + NT/2)$ である．

1 ノードから複数ノードに拡張した際，袖通信のための送受信データコピーの計算が追加される．袖通信は 1 ステップ毎に 12 回行われ，送信のためのデータコピーと受信後のデータコピーもそれぞれ 12 回行われる．また，一度のデータコピー送 (受) 信で 2 つの倍精度複素数へのメモリアクセスがある．この送受信データコピーにかかる時間 T_{copy} のモデル式を以下に示す．

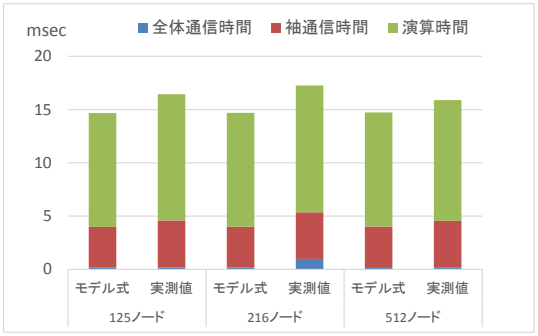

図 5 FX10 を用いた CCS QCD の性能評価モデル式と実測値の比較

$$T_{copy} = (12 \times 2 + 12 \times 2) \times 192N_i/\text{Bandwidth} \quad (12)$$

ここでは，キャッシュの効果を考慮せずにモデルを作成している．一方，1 ステップの演算時間 T_{cal} は式 9 を用いて，

$$T_{cal} = M/\text{perf} \quad (13)$$

であり，M は 1 ステップでの演算量である．以上より，複数ノードでの性能評価モデルを以下に示す．

$$T_{all} = T_{cal} + T_{copy} + 12T_{boundary} + 3T_{AllReduce} \quad (14)$$

これより，1 ノードの問題サイズを $(NX, NY, NZ, NT) = (8, 8, 8, 256)$ と固定した場合の実測値と性能評価モデル式の比較を行う．実測値の測定として，東京大学情報基盤センターに設置されている FX10 を用いる．FX10 の性能値を以下に示す．

- 1 ノードの理論性能：236.5 GFLOPS
- 1 ノードの理論メモリバンド幅：85 GB/s
- ノード間ネットワーク性能：5GB/s (双方向)

また，ネットワークレイテンシを 8 usec，AllReduce の 1 ステップのレイテンシを 8 usec とする．T_{cal} 部分は 1 ノードでの FX10 での実測値を用い，他の値に関しては FX10 の性能値を用いる．モデル式での値と実測値での計算の比較を図 5 に示す．図 5 より，性能評価モデル式と実測値に大きな差がないことがわかる．

3.2.4 CCS QCD のエクサシステム要求性能への外挿

作成した性能評価モデルを用いて，問題サイズ 256^4 格子の性能予想を行う．1 ノードの問題サイズを $(NX, NY, NZ, NT) = (8, 8, 8, 256)$ と固定した場合，問題サイズ 256^4 格子を扱うためには，32768 ノード必要である．性能評価モデルではノード数の増加は全体通信のみにしか依存せず，現性能での 32768 ノードの 1 ステップ当たりの予想計算時間は 14.8 msec である．1 ステップ当たり

の予想計算時間の内訳として，演算部分は 10.6 msec，袖通信部分は 3.81 msec また全体通信部分は 0.36 msec である．目標値の BiCGStab 部の実行時間が 3.1 msec/step であるため，ノード数を固定のまま，目標値を達成するためには特に演算部分と袖通信部分の改善が不可欠である．

現性能の評価をふまえて，エクサシステムで要求される性能について考察する．ここでは F/B 値は FX10 の実測値のままと仮定し，1 ノードの性能はルーフラインモデルと同等と仮定する．演算部分としては，CCS QCD はメモリバウンドであることから，エクサシステムではメモリバンド幅の向上が望まる．実行メモリバンド幅が 300 GB/s の場合，演算部分の計算時間は 2.19 msec　と予想される．また，通信部分はノード間ネットワーク性能 を 40 GB/s に向上させた場合，袖通信部時間は 0.47 msec と予想され，1 ステップ当たりの全計算時間が 3.02 msec となり，目標値が達成できる．

3.3 FFVC Mini の性能評価

3.3.1 FFVC Mini のエクサシステム要求性能

FFVC Mini のプログラムの制御構造は図 2 に示されている．FFVC Mini の V-P 反復，Poisson 反復回数はベンチマークプログラムとして使用することを考慮し，任意の回数を指定出来る．FFVC Mini のエクサシステム要求性能の一例として，1mm 幅の格子，1000 億セル，実時間 3 秒の計算を想定している．この系に対して，V-P 反復，Poisson 反復をディフォルトの反復回数 20，30 回の下で，100 万ステップの単精度計算を 3 時間程度の速度を目指す．つまり，エクサシステム要求性能としては格子サイズ 4096^3 の系に対して，以下の性能を満たすことである [3].

- 実行時間が 0.0108 sec/step 以下

ここでは，FFVC Mini の性能評価モデルを作成し，エクサシステムで要求される性能を外挿する．FFVC Mini ではデータ出力の設定が可能であるが，今回はデータ出力がない場合での性能評価を行う．

FFVC Mini のディフォルト反復回数の計算では，1 ステップ当たりに SOR コア計算が 600 回呼ばれ，SOR コア計算部が全計算時間の大部分を占める．ここでは SOR コア計算の性能評価モデルを作成する．

FFVC Mini では 3 次元空間データを，空間 3 次元で分割し，MPI プロセスにマッピングする．SOR 計算のデータ通信パターンを以下に示す．

- 隣接空間を担当する MPI プロセス間での境界データを交換.
- 1 要素の AllReduce の全体通信.

また境界条件は外部境界条件である．性能評価モデルを作成するうえで，それぞれの通信回数と通信量のデータが必要となる．FFVC Mini では袖通信を同期通信か非同期通信のどちらかを選択することが可能であり，今回は同期通信の場合を想定し，性能評価モデルを作成する．同期通信の場合，SOR コア計算の 1 ループ当たりに袖通信を 12 回行い，1 通信の量は $4N$ byte となる．ここでの N はそれぞれの面に対して，$(NX+2)\times(NY+2)$, $(NY+2)\times(NZ+2)$, $(NZ+2)\times(NX+2)$ であり，NX, NY, NZ　は 1 ノードが担当する領域である．一方，AllReduce の全体通信は 1 ループ当たりに 8 byte の通信を 1 度行う．

3.3.2 1 ノードの性能評価モデル

性能評価モデルを作成するために，東京大学情報基盤センターに設置されている FX10 を用いて通信のない 1 ノードでの実測 F/B 値を測定する．CCS QCD と同様に FX10 のシステムに付属しているプロファイラ機能を用いて，実測 F/B 値を測定する．1 ノードの問題サイズ $(NX, NY, NZ) = (128, 128, 128)$ の SOR コア計算の実測 FLOPS は 29.2 GFLOPS であり、単位時間当たりの実測メモリスループット量は 19.8 GB/s である．この値より，実測 F/B 値は 1.47 であり，実測 F/B 値を用いたルーフラインモデルを以下に示す.

$$\text{perf} = \min(\text{FLOPS}, 1.47 * \text{Bandwidth}) \tag{15}$$

ここでの FLOPS は LINPACK ベンチマークの 1 ノードの性能値であり，Bandwidth は STREAM メモリベンチマークでのメモリバンド幅の値とする．ルーフラインモデルより，FFVC Mini はメモリバウンドのアプリであり，FX10 の性能値を代入した結果は 88.2 GFLOPS である．一方，詳細プロファイラを用いた実測値は 29.2 GFLOPS であり，ピーク性能の約 2 割の性能しか出ていない．原因としては，本ミニアプリでは各格子あたりの情報の一部をメモリサイズ削減のためにビット配列として保持しており，整数演算であるその操作がオーバーヘッドとなっている可能性があげられる．

3.3.3 複数ノードの性能評価

1 ノードの実測値をもとに複数ノードの性能モデルを作成する．1 ノードから複数ノードの拡張によって，追加される計算として，ノード間のデータ通信，袖通信のための送受信データコピーがある．ノード間のデータ通信として，袖通信と全体通信の 2 つがあり，データ通信にかかる時間のモデル式をそれぞれ作成する．1 度の袖通信にかかる時間 $T_{boundary}$ と全体通信にかかる時間 $T_{AllReduce}$ のモデル式を以下に示す.

$$T_{boundary} = l + 4N \times g \tag{16}$$

$$T_{AllReduce} = (l_{AllReduce} + s \times g) * \log_2(Proc) \tag{17}$$

l はネットワークレイテンシ, $l_{AllReduce}$ は AllReduce の 1 ステップのレイテンシ, s はデータ長, g はネットワーク通信性能の逆数, N はそれぞれの面に対して, $(NX+2)\times(NY+2)$, $(NY+2)\times(NZ+2)$, $(NZ+2)\times(NX+2)$ である．

1 ノードから複数ノードに拡張した際，袖通信のための

送受信データコピーの計算が追加される．袖通信は SOR コア計算の 1 ループ当たり 12 回行われ，送信のためのデータコピーと受信後のデータコピーもそれぞれ 12 回行われる．また，FFVC Mini は CCS QCD の場合と比較して，コピー時の座標変換の演算回数が多い．そのため，演算量とメモリアクセス量の両方を考慮した送受信データコピーのルーフラインモデルを新たに作成する．一面のみに注目した際，データコピー送受信のメモリアクセス量は $24N$ byte (ロード 4，ストア 2)，演算量は $88N$ FLOP となる．メモリアクセス量と演算量から送受信データコピー部の F/B 値を 3.6 として，送受信データコピー部の性能評価モデルを作成する．現在，公開されている FFVC Mini では送受信データコピー部では OpenMP を使用した並列処理を行っていない．そのため，CPU1 コアとメモリ 1 チャンネルの性能を用いたルーフラインモデルから送受信データコピー部の性能評価モデルを作成する．送受信データコピー部のルーフラインモデルを以下に示す．

$$\mathrm{perf}_{copy} = \min(\mathrm{FLOPS}_1, 3.6 * \mathrm{Bandwidth}_1) \qquad (18)$$

ここでの FLOPS_1 は LINPACK ベンチマークの 1 コアの性能値であり，$\mathrm{Bandwidth}_1$ は 1 コア計算での STREAM メモリベンチマークでのメモリバンド幅の値とする．送受信データコピー部に関しては CPU バウンドになる．このルーフラインモデルを用いて，送受信データコピー部にかかる時間 T_{copy} のモデル式を以下に示す．

$$T_{copy} = 12 \times 88N/\mathrm{perf}_{copy} \qquad (19)$$

SOR コア計算の 1 ループの演算時間 T_{cal} は式 15 を用いて，

$$T_{cal} = M/\mathrm{perf} \qquad (20)$$

であり，M は SOR コア計算の 1 ループでの演算量である．以上より，複数ノードでの性能評価モデルを以下に示す．

$$T_{all} = T_{cal} + T_{copy} + 12T_{boundary} + T_{allreduce} \qquad (21)$$

これより，1 ノードの問題サイズを $(NX, NY, NZ) = (128, 128, 128)$ と固定した場合の実測値と性能評価モデル式の比較を行う．比較として，SOR コア計算部の 1 ループ当たりの計算時間を用いる．実測値の測定として，東京大学情報基盤センターに設置されている FX10 を用いる．FX10 の性能値は CCS QCD の性能評価のところに示されている．モデル式での値と実測値での計算の比較を図 6 に示す．図 6 より，隣接通信の実測値とモデル値の差が大きいことがわかる．今回袖通信では X(Y,Z) の正負方向に対して， Isend，Irecv，Wait_all 用いた同時通信を行っている．図 6 の近接通信の実測値はこの 3 つの時間の和を用いている．FFVC Mini では外部境界条件を用いているため，隅に配置されたノードでは袖通信する方向が減少し，減少した方向での送受信データコピー部の計算もなくなる．そのため，通信待ち時間が過大に評価されているため差が出たと予想される．また，性能評価モデル式全体と実測値の比較では大きな差がないとことがわかる．

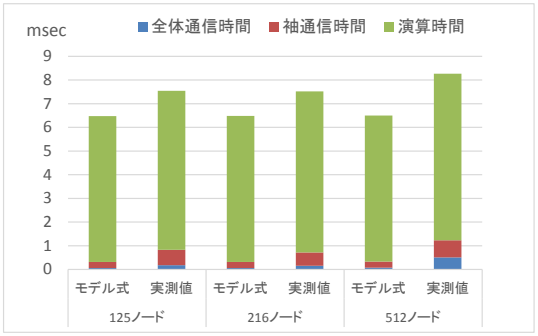

図 6 FX10 を用いた FFVC Mini の性能評価モデル値と実測値の比較

3.3.4 FFVC Mini のエクサシステム要求性能への外挿

作成した性能評価モデルを用いて，格子サイズ 4096^3 の性能予想を行う．1 ノードの問題サイズを $(NX, NY, NZ) = (128, 128, 128)$ と固定した場合，問題サイズ 4096^3 格子を扱うためには，32768 ノード必要である．性能評価モデルではノード数の増加は全体通信のみにしか依存せず，現性能での 32768 ノードでの SOR コア計算部分の 1 ループ当たりの予想計算時間は 6.5 msec である．1 ステップではこのループが 600 回あるため，現性能での 1 ステップ当たりの予想計算時間は 3.9 sec となる．

これより，以下の仮定を用いて，エクサシステムの要求を満たす条件を考察する．

- 1 ノードの性能はルーフラインモデルと同等とし，F/B 値も変わらないとする．
- 要求ノード数は 32768 と固定
- 送受信バッファへのデータコピーで，OpenMP 並列が行われる．

この仮定での現性能の 1 ループ当たりの予想計算時間は 1.6 msec となり，1 ステップではループが 600 回あるため，1 ステップ当たりの予想時間は 0.960 sec となる．エクサシステムの目標値は実行時間が 0.0108 sec/step 以下であるため，少なくても現性能の 100 倍以上の性能が要供される．具体的な値としてはメモリバンド幅が 15 TB/s CPU 性能 が 22 TFLOPS の場合，1 ステップの実行予想時間が 0.0056 sec となる．また，この性能での送受信データコピーの計算時間は 0.0005 sec となる．次にノード間通信部分についての要求性能について考察する．現通信性能での 1 ステップ当たりの予想通信時間は 0.154 sec であるが，通信量自体が少ないため，この部分の改善のためにはネットワーク速度の強化に加え，レイテンシの減少が必要となる．具体的な値としてはノード間ネットワーク性能を

200 GB/s に向上させ，レイテンシを 0.1 usec にした場合，1 ステップ当たりの予想全通信時間は 0.0031 sec と，1 ステップ当たりの全計算時間が 0.0092 sec となり，目標値が達成できる．また，FFVC Mini では非同期通信の計算も選択可能であり，非同期通信にした場合，送受信データコピーの計算と袖通信がオーバーラップする．しかし，送受信データコピー部の OpenMP 並列がきちんと行われた場合の現性能での SOR コア計算部の予想時間は 4.9×10^{-2} msec であり，これは袖通信にかかる時間 2.5×10^{-1} msec より小さい．そのため，ネットワーク性能の要求性能を下げるためには，送受信データコピー部だけではなく，全体の計算と通信のオーバーラップが必要となる．

4. まとめ

Fiber ミニアプリ集は理化学研究所および東京工業大学による将来 HPCI のあり方の調査研究の一環として開始されたプロジェクトであり，現在は理化学研究所を中心として継続して整備が進められている．本稿では開発がほぼ完了している 9 個のミニアプリについて概要を紹介し，そのうち 3 つのミニアプリについては性能モデルの構築を行い，将来の要求性能の実現可能性を検討した．引き続き性能モデルの改善および他のミニアプリの性能モデルの構築等を進める予定である．また 2014 年 6 月現在では CCS QCD と FFVC Mini のみ公開しているが，他のミニアプリについても近日中に公開する予定である．

謝辞 ミニアプリの開発・整備にあたり，以下の方々（順不同）からソースコードの提供，入力データの提供，ミニアプリ化への助言など，多大な協力をいただきましたことを感謝いたします．石川健一様（広大），玉田嘉紀様（東大），藤本明洋様（理研），小野謙二様（理研），池口満徳様（横市大），岡崎進様（名大），安藤嘉倫様（名大），宮崎剛様（物材研），八代尚様（理研），藤堂眞治様（東大），今田正俊様（東大），森田悟史様（東大）．本研究の一部は「将来 HPCI システムのあり方の調査研究アプリケーション分野」（代表：富田浩文）および文部科学省「特定先端大型研究施設運営費等補助金（次世代超高速電子計算機システムの開発・整備等 ）」で実施された内容に基づくものです．本論文の結果の一部は，理化学研究所のスーパーコンピュータ「京」を利用するとともに，「京」以外の HPCI システム利用研究課題を遂行して得られたものです（課題番号:hp120261）．本論文の結果の一部は東京大学情報基盤センターの Oakleaf-FX を用いて得られたものです．

参考文献

[1] Heroux, M. A., Doerfler, D. W., Crozier, P. S., Willenbring, J. M., Edwards, H. C., Williams, A., Rajan, M., Keiter, E. R., Thornquist, H. K. and Numrich, R. W.: Improving Performance via Mini-applications, Technical Report SAND2009-5574, Sandia National Laboratories (2009).

[2] Karlin, I., Bhatele, A., Keasler, J., Chamberlain, B. L., Cohen, J., DeVito, Z., Haque, R., Laney, D., Luke, E., Wang, F., Richards, D., Schulz, M. and Still, C.: Exploring Traditional and Emerging Parallel Programming Models using a Proxy Application, *27th IEEE International Parallel & Distributed Processing Symposium (IEEE IPDPS 2013)*, Boston, USA (2013).

[3] 計算科学ロードマップ（平成 26 年 3 月）：`http://hpci-aplfs.aics.riken.jp/`.

[4] Fiber Miniapp Suite: `http://fiber-miniapp.github.io/`.

[5] Boku, T., Ishikawa, K.-I., Kuramashi, Y., Minami, K., Nakamura, Y., Shoji, F., Takahashi, D., Terai, M., Ukawa, A. and Yoshie, T.: Multi-block/multi-core SSOR preconditioner for the QCD quark solver for K computer, *arXiv preprint arXiv:1210.7398* (2012).

[6] 寺井優晃，石川健一，杉崎由典， 南一生，庄司文由，中村宜文，藏増嘉伸，横川三津夫：スーパーコンピュータ「京」における格子 QCD の単体性能チューニング，情報処理学会論文誌. コンピューティングシステム， Vol. 6, No. 3, pp. 43–57 (2013).

[7] NGS Analyzer: `http://www.csrp.riken.jp/application_d_e.html#D2`.

[8] Fujimoto, A., Nakagawa, H., Hosono, N., Nakano, K., Abe, T., Boroevich, K. A., Nagasaki, M., Yamaguchi, R., Shibuya, T., Kubo, M., Miyano, S., Nakamura, Y. and Tsunoda, T.: Whole-genome sequencing and comprehensive variant analysis of a Japanese individual using massively parallel sequencing., *Nature Genetics*, Vol. 42, No. 11, pp. 931–936 (2010).

[9] Ono, K., Kawashima, Y. and Kawanabe, T.: Data Centric Framework for Large-scale High-performance Parallel Computation, *Procedia Computer Science*, Vol. 29, No. 0, pp. 2336 – 2350 (2014).

[10] Ikeguchi, M.: Partial rigid-body dynamics in NPT, NPAT and NPγT ensembles for proteins and membranes, *Journal of computational chemistry*, Vol. 25, No. 4, pp. 529–541 (2004).

[11] MODYLAS: `http://www.modylas.org`.

[12] Andoh, Y., Yoshii, N., Fujimoto, K., Mizutani, K., Kojima, H., Yamada, A., Okazaki, S., Kawaguchi, K., Nagao, H., Iwahashi, K. et al.: MODYLAS: A Highly Parallelized General-Purpose Molecular Dynamics Simulation Program for Large-Scale Systems with Long-Range Forces Calculated by Fast Multipole Method (FMM) and Highly Scalable Fine-Grained New Parallel Processing Algorithms, *Journal of Chemical Theory and Computation*, Vol. 9, No. 7, pp. 3201–3209 (2013).

[13] Bowler, D., Choudhury, R., Gillan, M. and Miyazaki, T.: Recent progress with large-scale ab initio calculations: the CONQUEST code, *physica status solidi (b)*, Vol. 243, No. 5, pp. 989–1000 (2006).

[14] NICAM-DC: `http://scale.aics.riken.jp/nicamdc/`.

[15] Satoh, M., Matsuno, T., Tomita, H., Miura, H., Nasuno, T. and Iga, S.-i.: Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations, *Journal of Computational Physics*, Vol. 227, No. 7, pp. 3486–3514 (2008).

[16] ALPS/looper: `http://wistaria.comp-phys.org/alps-looper`.

[17] Todo, S. and Kato, K.: Cluster algorithms for general-S quantum spin systems, *Physical review letters*, Vol. 87, No. 4, p. 047203 (2001).

[18] Tahara, D. and Imada, M.: Variational Monte Carlo Method Combined with Quantum-Number Projection and Multi-Variable Optimization, *Journal of the Physical Society of Japan*, Vol. 77, No. 11, p. 114701 (2008).
[19] HPCI 技術ロードマップ白書（2012 年 3 月）：http://www.open-supercomputer.org/.
[20] 滝澤真一朗，松田元彦，丸山直也：MapReduce による計算科学アプリケーションのワークフロー実行支援，ハイパフォーマンスコンピューティングと計算科学シンポジウム（HPCS2014) (2014).
[21] 富士通株式会社：FX10 スーパーコンピューターシステム Oakleaf-FX / Oakbridge-FX 利用手引書 (2014).
[22] 東京大学情報基盤センター：Oakleaf-FX の使い方 Fujitsu PRIMEHPC FX10, http://nkl.cc.u-tokyo.ac.jp/pFEM/FX10-introduction.pdf.
[23] 大島聡史：富士通 PRIMEHPC FX10 チューニング連載講座 1, http://www.cc.u-tokyo.ac.jp/support/press/news/VOL14/No2/201203tuning-fx10-hard.pdf (2012).