Vol.2014-HPC-144 No.3
2014/5/26

通信削減アルゴリズム CAQR の RSDFT の直交化処理への適用と評価

片桐 孝洋†, 高山 恒一††, 米村 崇††, 熊洞 宏樹††, 猪貝 光祥†3,
北上 純一†3, 江口 義之†4, 深谷 猛†5, 山本 有作†6, 岩田 潤一†7
内田 和之†7, 大島 聡史† 中島 研吾†

本報告では，量子力学的第一原理シミュレーションのソフトウェア RSDFT における直交化処理に，通信削減アルゴリズムを用いた QR 分解アルゴリズムである CAQR を組み込んだ性能について報告する．東京大学情報基盤センターの FX10 を用いた 1,024 ノード実行（4,096MPI，MPI 当たり 4 OMP 実行のハイブリッド MPI-OpenMP 実行）におけるバンド分割が 64 の時の実行では，従来の Gram-Schmidt 法による直交化に比べ CAQR を利用すると，最大で 11 倍の高速化が得られる事例があった．

1. はじめに

本報告は，レイテンシコアの高度化・高効率化による将来の HPCI システムに関する調査研究（以降，単に調査研究とよぶ)におけるターゲットアプリケーションにおいて，エクサスケールの計算機環境に向く，従来アルゴリズムの変更に関する報告である．

本報告の構成は以下のとおりである．2 章で調査研究の目的を簡単に述べる．3 章は，本報告でのターゲットアプリケーションである RSDFT と，エクサスケールの計算機に向くアルゴリズムの 1 つと考えられている通信削減アルゴリズムにおいて，直交化処理を行う CAQR アルゴリズムの説明する．4 章は，富士通 PRIMEHPC FX10 を用いた性能評価を報告する．最後に，本報告の知見を述べる．

2. 調査研究の目的

2.1 将来の HPCI システムのあり方の調査研究

本調査研究は，第 4 期科学技術基本計画（平成 23 年 8 月 19 日閣議決定)で掲げられた国家存立の基盤としての世界最高水準のハイパフォーマンス・コンピューティング技術の強化，及び科学技術基盤の充実強化に向けた重要な取り組みの一つとして，HPC 技術等の HPCI システムの高度化に必要な技術的知見を獲得することを目的とし，平成 24 年度，平成 25 年度に調査研究を実施するものである[1].

2.2 レイテンシコアの高度化・高効率化による将来の HPCI システムに関する調査研究

本調査研究は，東京大学を中心とし，九州大学，富士通，日立製作所，日本電気による調査研究である[1]．2018 年頃設置可能な並列システムを，汎用型プロセッサからのアプローチでフィージビリティ・スタディ（FS）を行う．アプリケーション，システムソフトウェア，アーキテクチャの co-design を行う．システムソフトウェアスタック共通化 (From PC cluster to high-end machines)を行う．

2.3 本調査研究における進め方

「今後の HPCI 技術開発に関する報告書」[2]，および，「計算科学ロードマップ白書」[3]を尊重し，京および FX10 におけるアプリケーション並列性能および I/O 性能，耐故障性および運用・保守の観点で課題を精査し，概念設計に反映する．進め方の概要を図 1 に示す．

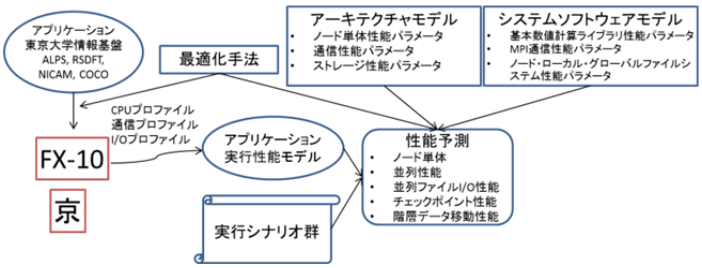

図 1　本調査研究における調査研究の進め方

2.4 利用シナリオ

利用シナリオ（図 1 における「実行シナリオ群」）とは，各アプリケーションの実行において，特徴的なジョブの実行形態のことである．主に以下のアンサンブル型を想定し，入出力ファイルなどの I/O 性能を含め，システム全体の設計に反映させる．

- **アンサンブル型**：全系の 1/10～1/100 の資源を利用する 1 ジョブに対し，複数ジョブを同時実行して全資源を使い切る形態．この形態では，複数同時のファイル入力，および複数同時のファイル出力が起こる．

3. RSDFT と CAQR

3.1 RSDFT の概要

前節で説明した FS のターゲットアプリケーションの 1 つに，RSDFT (Real-Space Density-Functional Theory)が

† 東京大学 情報基盤センター スーパーコンピューティング研究部門
†† 日立製作所 情報・通信システム社
†3 日立超 LSI システムズ
†4 日立ソリューションズ東日本
†5 理化学研究所 計算科学研究機構
†6 電気通信大学 情報理工学部
†7 東京大学 大学院工学系研究科

ある．RSDFT は Si ナノワイヤ等，次世代デバイスの根幹材料の量子力学的第一原理シミュレーションである．実空間差分法を利用している．

3.2 RSDFT の処理

図 2 に，RSDFT の処理の流れ図（参照：論文[4]）を示す．

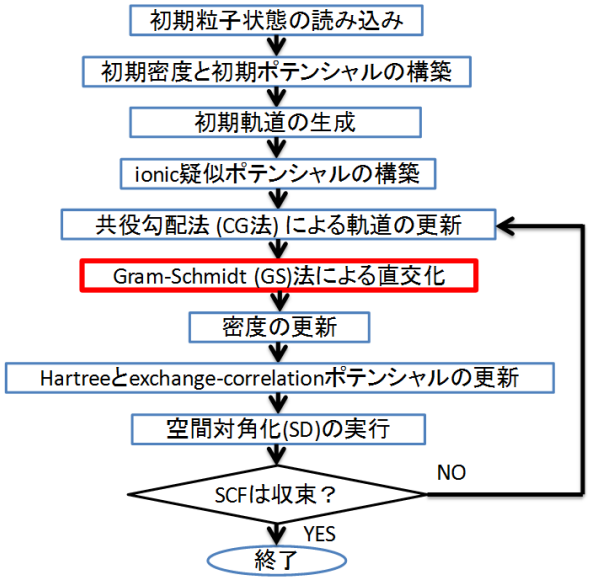

図 2　RSDFT の処理の流れ図（参照：論文[4]）

図 2 において，実行時間の分布は，計算機，入力データ，および，並列数に依存する．論文[4]によると，京コンピュータの実行では，約 47%が SD の時間，約 26%が CG の時間，そして，約 26%が GS 法による直交化の時間である．計算量的には N を原子数とすると，SD と GS は $O(N^3)$，CG は $O(N^2)$ のため，大規模計算では SD と GS の部分が増えていくことが予想される．そこで本稿では，GS 法による直交化の並列実行による高速化に焦点を当てる．

3.3 RSDFT における GS 法の実装

RSDFT における現在の GS 法による直交化の実装は，行列－行列積である BLAS3 演算を利用した，ブロック化 GS 法を実装している[4]．そのため，ノード内にデータが多数ある実行では，ブロック化 GS 法の性能は BLAS3 性能と等価になる．したがって，高効率での実行が可能である．ところが，超並列実行では，メモリ制約や実行時間制約により，必ずしもノード内のデータ量が大きくならない状況がある．その場合は，通信時間が無視できなくなる．また，図 2 にしめすように RSDFT 処理は直交化だけではないため，その他の処理を含む全体時間で問題サイズが定まる．そのため，プロダクトランの行列サイズは小さくなる傾向がある．

RSDFT においては，固有ベクトルの要素を分割する「格子分割」（もしくは「空間分割」）と，固有ベクトル全体を分割する「バンド分割」がある．バンド分割では，通信時間を最小化するため，直交化処理に入る前に 2 次元プロセッサーグリッドにおける行方向のプロセス群で，重複した固有ベクトルを所有するように MPI_AllgatherV を発行する．この説明を図 3 にのせる．

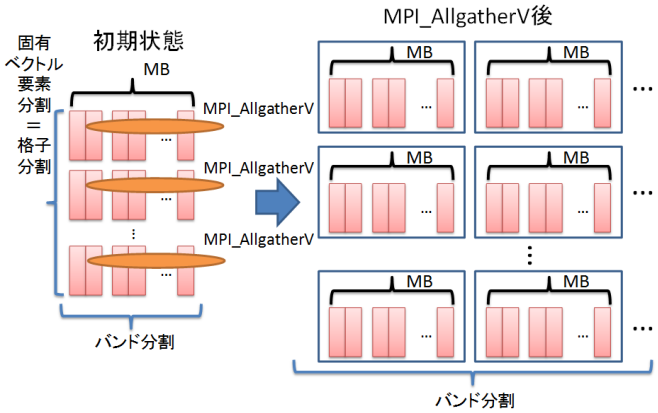

図 3　従来 GS におけるバンド分割時の MPI_AllgatherV

図 3 から，バンド分割時においては，MPI_AllgatherV が必要となる．また，重複してデータを所有するため，メモリ空間を余分に必要とする．したがって，メモリ量が取れない状況では，実行不可能になる可能性がある．これらは，現在の RSDFT の実装における問題点であり，MPI_AllgatherV を用いない実装も原理的には可能である．ただし MPI_AllgatherV を使わない実装の場合，原理的に，現在の GS 法による直交化よりも通信時間が増加する．

バンド分割に限らず空間分割においても，GS 法による直交化時の通信時間は，並列数にしたがって増えていく．特に，1 回の通信当たりのデータ転送の時間よりも，メッセージ立ち上げ時間（通信レイテンシ）が無視できなくなる．通信レイテンシが顕著になる問題は，GS 法に限らず，エクサスケールコンピューティングに向けた超並列処理で生じる一般的な問題である．

3.4 CAQR の概要

そこで現在，通信レイテンシの時間を最小化する並列アルゴリズムである，通信削減（Communication Reducing, CR），もしくは，通信回避（Communication Avoiding, CA）アルゴリズムの研究が盛んである．直交化分解である QR 分解に CA を適用したアルゴリズムを，CAQR[5][6]とよぶ．

いま，直交化対象となる固有ベクトルの集合を行列 A とする．このとき，$A=QR, Q^TQ=I$ となる行列分解（QR 分解）を行うと，行列 Q が直交化された行列となる．

このとき，従来の GS 法（修正 GS 法の分散並列化）では，対象となる行列 A を行方向（つまり，RSDFT の空間分割）にデータ分散すると，必要な内積演算をするためにプロセス間で逐次に発行されるリダクション演算（MPI_Allreduce）が必要となる．このため，高並列処理でリダクション演算の通信レイテンシがボトルネックになる．

一方 CAQR では，アルゴリズムを従来から変更する．具体的には，A を m x n 行列とすると，長細の行列（$m >> n$）

用の QR 分解である Tall-Skinny QR（TSQR）を採用し，通信回数を削減する．さらに TSQR での QR 分解の結果を利用し，任意のサイズの QR 分解を，通信回数を削減した上で行う方法が CAQR である[5][6]．

図 4 に，従来 GS 法による直交化（QR 分解）と，CAQR において TSQR 相当の処理にける通信パターンの例を示す．

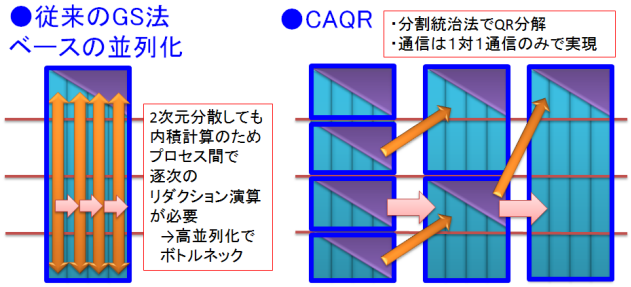

図 4　従来 GS 法と CAQR との違い．(4MPI プロセスの例)．紫の上三角は QR 分解された行列 R を示す．

CAQR では，従来 GS 法と通信の総量は変わらないが，通信回数が削減できる．図 4 に示すように，CAQR ではローカルな QR 分解による並列性の向上と，行列 Rに関する通信（二分木通信）の通信処理を，1 対 1 通信で実現できる．また，通信について，$O(\log P)$，ここで Pはプロセス数，の並列性が抽出できるメリットがある．

一方，従来 GS 法では，行方向分割した場合，プロセス全体にわたるリダクション演算は避けることができないa．かつ図 4 が示すように，このリダクション演算は O(列数)の逐次性がある．そのため，通信レイテンシが無視できなくなる．ただし従来 GS 法でも，演算処理がブロック化できるため，BLAS3 演算が主体になる．そのため，演算効率を高くすることができる．

3.5 CAQR の実装法の概略

CAQR の実装は多数考えられる．ここでは，本稿で採用した CAQR コードの実装について，その概略を述べる．

まず，図 5 のように，行列 A を，ブロック幅 g_bl で，列方向に 1 次元ブロック-サイクリック分割する．

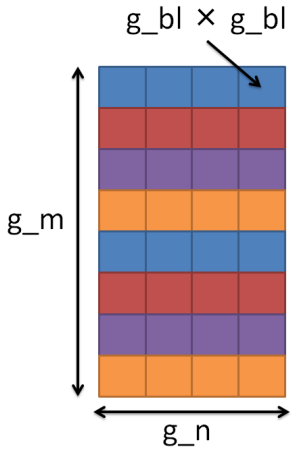

図 5　本稿で利用する CAQR のデータ分散（行列 A）．g_bl は CAQR のブロック幅（パネルサイズ）である．

CAQR は，以下の 3 つの手順で QR 分解を行う．

1．第 1 列から順番に，TSQR アルゴリズムにより，列ブロックの QR 分解（パネルの QR 分解）を行う．

2．残りの部分のパネル更新を行う．

3．QR 分解が終了したら，逆順に直交変換を作用させて Qを陽的に生成する．

QR 分解の際の Qは compact-WY 表現の形で保持する．TSQR は，二分木にしたがって三角行列を上下に二つならべた行列の QR 分解をする．Qの生成時も同様である．

4. 性能評価

4.1 計算機環境

本稿で扱う計算機の詳細を以下にまとめる．

- FX10 スーパーコンピュータシステム（FX10）
  - OS：専用 OS（XTCOS）
  - 計算ノード数：4,800
  - 1 ノード理論性能：236.5GFLOPS
  - 総理論演算性能：1.13PFLOPS
  - 1 ノード記憶容量：32GB，総主記憶容量：150TB
  - インターコネクト：6 次元メッシュ/トーラス
  - ノード間ネットワーク性能：5GB/s×双方向
  - コンパイラ：富士通 Fortran コンパイラ．Version 1.2.1 P-id: T01641-02.

最大で 1,024 ノード（16,384 コア）を用いて性能評価を行う．

4.2 CAQR の実装

RSDFT に実装されている従来 GS 法のコードは，ブロック化 GS 法の並列版であり，京コンピュータ向きに十分にチューニングされているコード[4]を利用した．

本稿で利用する CAQR コードは，JST CREST 「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」領域，「ポストペタスケールに対応した階層モデルによる超並列固有値解析エンジンの開発」プロジェクト（代

a 従来 GS 法で列方向分割した場合（RSDFT のバンド分割のみの場合）は，ベクトル全体を転送する逐次の通信が必要になる．

表：櫻井鉄也教授）[7]で開発中のコードを利用した．

RSDFT においては，図 2 の SD 中での固有値ソルバーにおいて，ScaLAPACK の利用に対応するため，データ分散が 2 次元ブロック-サイクリック分散になっている．ところが上記の CAQR のコードは現在，1 次元ブロック-サイクリック分散のみ対応している．そのため，CAQR を呼び出す前後でデータの再分散をしないと CAQR のプログラムが利用できない．このデータ再分散は，RSDFT におけるデータ分散を再構築すると不要にできる．そのため本稿の評価では，このデータ再分散の時間は含まず，直交化の時間のみを測定する．

4.3 従来 GS の性能

原子数 4,096，メモリバンド数 8,192 について，RSDFT における GS 直交化の実行時間を測定する．

図 6 は，ノード数を可変にした場合の直交化部分の時間である．ここで，ノード数と MPI プロセス数は同じであり，MPI プロセスから派生する OMP スレッド数を 16 に固定した，ハイブリッド MPI-OpenMP 実行である．RSDFT における空間分割数は MPI プロセス数と同じである．バンド分割はしていない．また，プロセッサーグリッドの構成は，4x8 (32 ノード)，8x8 (64 ノード)，および，32x32 (1,024 ノード)である．1,024 ノード実行時の利用コア数は，16,384 コアである．

従来 GS において，演算時間として行列 - 行列積の時間(DGEMM)，行列 - ベクトル積の時間(DGEMV)を計測した．また，通信時間には，MPI_Allreduce の時間，および，MPI_AllgatherV の時間が含まれている．

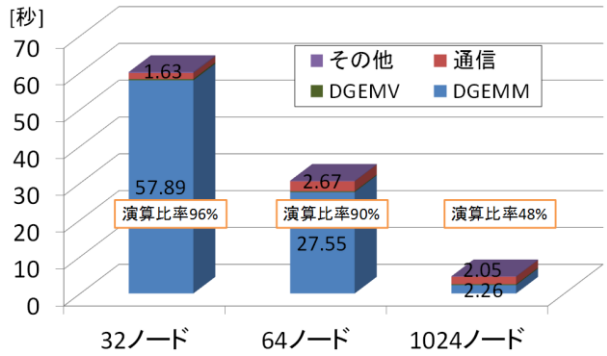

図 6　従来 GS におけるノード数可変のデータ．
ノード数と MPI プロセス数は同じ．
OMP スレッド数は 16 に固定．

図 6 から，利用ノード数が増えるにしたがい，実行時間が減少する．かつ，全体の時間に対する，演算時間(DGEMM と DGEMV の時間)の占める割合(演算比率)が減少していく．

本稿で用いる CAQR は通信時間を減少させたアルゴリズムのため，従来 GS で通信時間が無視できなくなる状況で評価する．したがって図 6 から，演算時間が 48%となる 1,024 ノードの実行条件を採用することにする．

次に 1,024 ノード実行に固定し，MPI プロセスから派生する OMP スレッド数を変更させ，コアすべてを使い切るハイブリッド MPI-OpenMP 実行の性能を評価する．その結果を，図 7 に載せる．

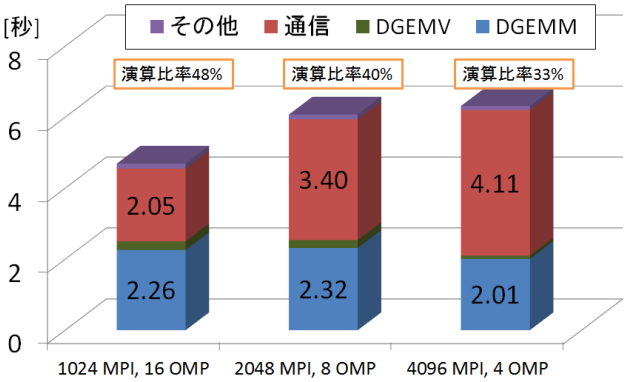

図 7　従来 GS における OMP スレッド数可変のデータ．
ノード数は 1024 ノードで固定．
空間分割数と MPI プロセス数は同じ．バンド分割なし．

図 7 から，OMP スレッド数が減るにしたがって (MPI プロセス数が増えるにしたがって)，実行時間が増加する．言い換えると，空間分割数が増えるにしたがって，実行時間が増加するといえる．

次に 1,024 ノード実行において，空間分割とバンド分割の影響を見るため，OMP スレッド実行数を 4 に固定し，空間分割数（S）とバンド分割数（B）を変化させた場合の実行時間を，図 8 に示す．

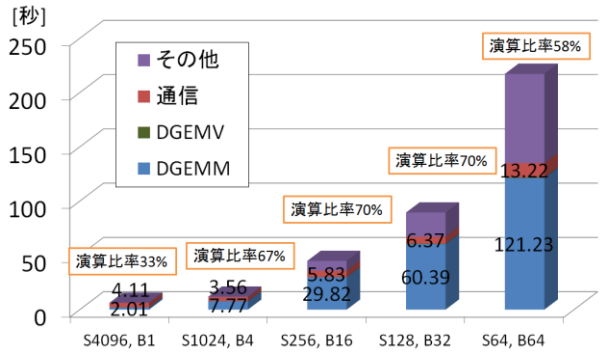

図 8　従来 GS におけるバンド分割（B）可変の性能．
ノード数は 1,024 ノードで固定．OMP 数は 4 で固定．
空間分割数(S)×バンド分割数(B)と
MPI プロセス数は同じ．

図 8 では，バンド分割数(B)を増やしていく（すなわち，空間分割数(S)を減らしていく）と，実行時間が増加する．特に DGEMM の時間が，バンド分割数に比例して増加している．ここでは，8,192 次元の行列を 1,024 ノード(プロセッサーグリッド：64x64)で分割している都合から，1MPI プロセス当たりの行列サイズは 128x128 となる．いま，DGEMM

における演算ブロックは 128 を指定しており，この状況では，データ分散のためのブロック幅と，演算のためのブロック幅が同じになる．

この状況では，MPI プロセス当たり小さな行列を扱っている．そのためバンド分割を進めていくと，何らかの都合で，DGEMM 性能が劣化していることが，図 8 におけるバンド分割を増やしたときの性能劣化の要因であると考えられる．たとえば，演算ブロック数を 128 より小さくすると性能が改善される可能性があるが，詳細な分析は今後の課題である．また，その他の時間も，バンド分割を増やしたときに増加している．この原因調査も必要である．

4.4 CAQR の性能

図 9 に，CAQR の性能をのせる．空間分割数(S)と OMP スレッド数は可変である．ノード数は 1,024 ノードで固定している．空間分割数(S)と MPI プロセス数は同じである．また，CAQR の通信は 1 対 1 通信のみである．

S4096 実行の時，ジョブの計算ノードへの割り当てを 1 次元指定（1 次元）と，2 次元指定（2 次元，32x32）したものをのせる．なお CAQR はベクトルを 1 次元ブロック-サイクリック分散するため，従来 GS の空間分割がされた状態（ただし、データ分散は 1 次元のブロックサイクリック分散に限定で，このデータ分散は従来 GS の実装ではできない．）と等価である．CAQR の演算ブロック g_bl は 128 である．

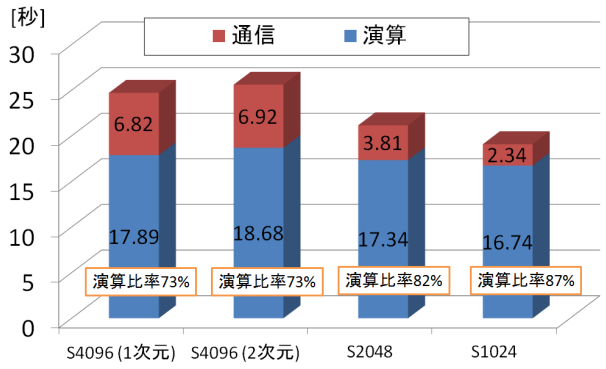

図 9　CAQR の性能．空間分割数(S)と OMP スレッド数は可変．ノード数は 1024 ノードで固定．
空間分割数(S)と MPI プロセス数は同じ．

図 9 から，CAQR では空間分割数（S）が減ると，全体時間，および，通信時間ともに高速になる．図 9 の結果から，S1024 の実行時間を CAQR の実行時間にとる．

4.5 従来 GS と CAQR の性能比較

図 10 に，CAQR においては S1024 の時，従来 GS においては空間分割 S1024 以上，かつ，バンド分割が 4 以上である（S1024, B4）（S256, B16）（S128，B32）（S64, B64）の時の，演算時間と通信時間を抽出してのせる．

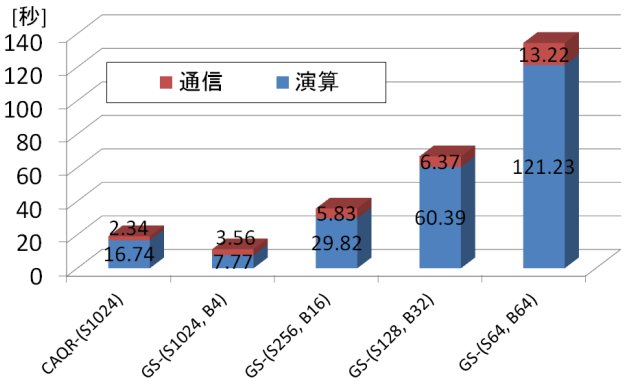

図 10　CAQR と従来 GS との比較．空間分割数(S)と OMP スレッド数は可変．ノード数は 1024 ノードで固定．
空間分割数(S)と MPI プロセス数は同じ．

図 10 では，空間分割 S が 1,024 の時，CAQR の演算時間が従来 GS の演算時間の約 2.2 倍かかる．これは，アルゴリズム的に CAQR は従来 GS よりも 2 倍の演算量があることが確認されていることから妥当な結果である．一方，通信時間については，CAQR は 2.34 秒に対し，従来 GS は 3.56 秒であり，通信削減の効果が示されている．ただし，演算時間と通信時間の総合時間は，CAQR が 19.08 秒に対し，従来 GS が 11.33 秒であり，従来 GS の方が高速である．

一方，従来 GS において，バンド分割を 16，32 と増やしていくと，従来 GS は遅くなっていく．通信時間の増加もあるが，すでに指摘したように，DGEMM 時間を含む演算時間の増加が著しい．結果として，従来 GS で(S64, B64)の時，演算時間と通信時間を総合すると約 134 秒になる．この時，CAQR による速度向上は約 7.03 倍となる．また，演算以外のその他の時間を入れると，CAQR は 19.08 秒に対し，同様の条件の従来 GS では 217 秒である．このことから，CAQR を導入することで約 11.3 倍の高速化になる．

すでに説明したように，現在の RSDFT の従来 GS の実装では，通信時間を減らすためにバンド分割時に MPI_AllgatherV を発行する実装をしている．そのため，従来 GS の図 10 の状況は，メモリ量が多くなる代わりに，十分に通信時間が削減される実装での評価といえる．もしメモリ量制約の観点から，MPI_AllgatherV を用いない実行をしなくてはならない場合は，図 10 の従来 GS の通信時間よりも増加することが予想される．

したがって以上から，バンド分割を多くしないといけない状況においては，CAQR の導入により，従来 GS より高速化できる可能性があるといえる．

5. おわりに

本報告では，通信削減アルゴリズムを適用した CAQR を，RSDFT の直交化部分に組み込んだ性能について報告した．

性能評価の結果，現在の RSDFT の実装においては，バ

ンド分割を増やすと従来 GS による直交化時間が増加する場合がある．その場合においては，CAQR が高速になることがある．ただし CAQR の利点を享受するには，データ分散を現在の CAQR の実装である 1 次元ブロックーサイクリック分散にする必要がある．そのため，RSDFT のコード自体の修正が必要になる．

今後の課題として，より柔軟な 2 次元ブロックーサイクリック分散による CAQR を実現することがあげられる．また，CAQR の演算部分(DGEMV の部分)の実行効率を改良し，従来 GS の演算時間に対して高速化する必要がある．さらに，RSDFT において，バンド分割を増やしたときの性能劣化の原因についても解析する必要がある．

謝辞

本研究は，文部科学省「将来の HPCI システムのあり方の調査研究」（平成 24 年度～平成 25 年度），および，JST CREST 「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」領域，「ポストペタスケールに対応した階層モデルによる超並列固有値解析エンジンの開発」プロジェクト（代表：櫻井鉄也教授）の支援による．

また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:hp120128）．

参考文献

1) 文部科学省「将来の HPCI システムのあり方の調査研究」に係る実施機関の選定について」，平成 24 年 6 月 15 日．
http://www.mext.go.jp/b_menu/houdou/24/06/1322138.htm
2) HPCI 技術ロードマップ白書，2012 年 3 月．
http://open-supercomputer.org/wp-content/uploads/2012/03/hpci-roadmap.pdf
3) 計算科学ロードマップ白書，2012 年 3 月．
http://open-supercomputer.org/wp-content/uploads/2012/03/science-roadmap.pdf
4) Hasegawa, Y., Iwata, J., Tsuji, M., Takahashi, D., Oshiyama, A., Minami, K., Boku, T., Shoji, F., Uno, A., Kurokawa, M., Inoue, H., Miyoshi, I., Yokokawa, M.: First-principles Calculations of Electron States of a Silicon Nanowire with 100,000 Atoms on the K computer, Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (2011)
5) Demmel, J., Grigori. L., Hoemmen, M. and Langou, J.: Communication-optimal parallel and sequential QR and LU factorizations, Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report, No. UCB/EECS-2008-89 (2008)
6) Anderson, M., Ballard, G., Demmel, J., and Keutzer, K.: Communication-Avoiding QR Decomposition for GPUs, Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report, No. UCB/EECS-2010-131 (2010)
7) 「ポストペタスケールに対応した階層モデルによる超並列固有値解析エンジンの開発」プロジェクト HP
http://h4es.cs.tsukuba.ac.jp/