jh160009-NAJ

タイルアルゴリズムの大規模適用時の通信最適化

鈴木智博（山梨大学・大学院総合研究部）

概要　数値線形代数計算の重要なアルゴリズムに Cholesky、LU、QR などの行列分解がある。行列分解の計算量は比較的多いため大規模な問題に対して高速な実装が求められている。行列分解のタイルアルゴリズムは、行列を小行列（タイル）に分割し、個々のタイルに対して処理を行うことで、細粒度のタスクを大量に生成することが可能である。高並列な計算環境において、タスク並列プログラミングモデルに基づき、非同期にタスクを実行することで、並列計算資源を効果的に利用することが期待できる。我々のこれまでの研究によって、共有メモリ環境においては十分に高速な実装が得られているものの、分散メモリ環境においては十分な性能を発揮する実装が得られていない。本研究の目的は、大規模問題に適用する行列分解のタイルアルゴリズムに対して、タスクスケジューリング手法の改良、通信削減アルゴリズムの適用などによりノード間通信の最適化を行うことである。

1.　共同研究に関する情報

(1) 共同研究を実施した拠点名

東京大学情報基盤センター

(2) 共同研究分野

- ■　超大規模数値計算系応用分野
- □　超大規模データ処理系応用分野
- □　超大容量ネットワーク技術分野
- □　超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

- 鈴木智博（山梨大学・大学院総合研究部）アルゴリズム開発、研究総括
- 大島聡史（元東京大学・情報基盤センター）プログラム並列化、最適化

2.　研究の目的と意義

数値線形代数計算の中で、多くの用途に利用されるという意味で重要なアルゴリズムに行列分解がある。具体的には Cholesky、LU、QR などのアルゴリズムである。科学技術計算の大規模化に伴ない、大規模問題に適用可能な高速な実装が求められている。行列分解のタイルアルゴリズムは、行列を小行列（タイル）に分割し、個々のタイルに対して処理を行うことで、細粒度のタスクを大量に生成することが可能である。高並列な計算環境において、タスク並列プログラミングモデルに基づき、非同期にタスクを実行することで、並列計算資源を効果的に利用することが期待できる。

我々のこれまでの研究により共有メモリ環境においては十分に高速な実装が得られているものの、より大規模な行列を扱うことができる分散メモリ環境においては十分な性能を発揮する実装が得られていない。タイルアルゴリズムのタスクには依存関係があるので、非同期にタスクを実行するためにはタスクの実行状況をノード間で共有する必要がある。そのため比較的小さいデータ量のノード間通信が多く必要となる。

本研究の目的は、大規模問題に適用する行列分解のタイルアルゴリズムに対して、ノード間通信の最適化を行うことである。ノード間通信で受け渡される行列データはタイルに分割されているので、タイルサイズが最適化の一つのパラメータとなる（**タイルサイズチューニング**）。また、近年重要性が増大している**通信削減（communication avoiding）アルゴリズム**をタイルアルゴリズムに適用する研究がある。

本研究で想定している高並列計算環境は近年のマルチコアアーキテクチャのノードからなる

クラスタシステムであり、ノード内とノード間でそれぞれスレッド並列化と分散メモリ並列化の異なる流儀の並列化が必要なハイブリッド並列化を行う必要がある。また、高速な実装とするためにノード間通信の最適化を行うことが必須である。近年の複雑な計算環境においてこのような最適化を行うためには専門的な知識・技術が必要である。そのため東京大学情報基盤センターを共同研究拠点に選び、同センターのスーパーコンピュータ FX-10 を使用して研究を進めると共に、同センター教員に研究協力を依頼した。

このような研究テーマに関して、これまでアルゴリズム開発を行う数理的な分野の専門家自身が並列化を行うことが多かった。しかし近年の計算機アーキテクチャの複雑化により、効率的に並列化・最適化を行うために高度な知識・技術が必要となり、アルゴリズム開発者とこのような知識・技術を持った研究者との協業が必須となっている。

本研究の成果物は、大規模並列環境で効果的な実装であることが求められる。そのため、研究室レベルで所有するワークステーションによる小規模ベンチマーク的な実験結果ではなく、申請する共同研究拠点における計算資源を使用した実用的な規模の実験結果によって実装の性能を評価する必要がある。

3. 当拠点公募型共同研究として実施した意義

科学技術計算の大規模化に伴って行列を扱う数値線形代数計算の大規模化、高速化の要求が高くなっており、高い並列性を持つ近年のハードウェアの性能を最大限に引き出す行列分解アルゴリズムが現在求められている。特に LU 分解(不完全 LU 分解)、QR 分解は数値線形代数分野で多用される計算であり、高並列環境における大規模行列向けの高速な実装は有用性が高い。

クラスタシステムにおける標準的な数値線形代数ライブラリである ScaLAPACK が近年のハードウェアの性能を十分に発揮できていないことが指摘されて久しいが、クラスタシステム上の多くのアプリケーションが現在もこのライブラリを使用している。これに換わるクラスタシステム向け数値計算ライブラリを開発するという側面から、国内で最高クラスのスーパーコンピュータを使用した研究開発を行う意義は非常に高い。

前述の通り、タイルアルゴリズムのタイルサイズチューニングは負荷分散、通信最適化、誤差への影響の面から重要性が高いが、これに関する研究報告は現在ほとんど見られない。

上記をまとめると以下となる。

- **大規模並列環境における高速な行列分解ルーチンの提供**
- **国内最高クラスのスーパーコンピュータによる実装の性能評価**
- **未着手の研究課題への取り組み**

4. 前年度までに得られた研究成果の概要

タイルアルゴリズムは、大量の細粒度タスクをタスク並列プログラミングモデルとして実装し、これらを非同期に実行することで並列計算資源を有効に活用することを目的としている。共有メモリ環境におけるスレッド並列化のデファクトスタンダートな API に OpenMP がある。OpenMP は基本的には Fork-Join 型のデータ並列プログラミングモデルに基づく。これに対して、タイルアルゴリズムに必要となる非同期実行のための動的タスクスケジューリングを共有メモリ環境上で OpenMP により実装した。

京都大学学術情報メディアセンターの 2013 年度プログラム高度化支援事業に「動的 schedule 版タイル QR 分解の MPI/OpenMP ハイブリッド実装」(代表者：鈴木智博) が採択され、クラスタシステム向け実装の**ノード間通信の最適化**を行った。Cray XE6 の最大８ノード使用した実験では、この最適化により比較的小さい問題でも強スケールする実装が得られた。

東京大学情報基盤センターを拠点とした平成 27 年度 JHPCN 共同研究として「行列分解のタイルアルゴリズムの高並列環境における最適化」

（代表者：鈴木智博）が採択され、主として**タイルサイズチューニング**と東京大学情報基盤センターの FX-10 スーパーコンピュータ上で**縦長行列に対するタイルアルゴリズムの最適化**を行った。タイルアルゴリズムの実行時パラメータのチューニングに対して有効な手法に枝刈り探索がある。これはタイル QR 分解そのものを実行してパラメータの評価を行うのではなく、主要なタスクに対して実行したチューニング結果を最適パラメータ候補とし、この候補のみについてタイル QR 分解を実行するものである。我々は、共有メモリ環境においてタスク数の指標を導入することで最適パラメータ候補を更に削減することが可能であることを示した。これにより、枝刈り探索の時間はそれまでの半分近くに短縮された（7 研究成果, (2), 1)。

LU 分解、QR 分解のタイルアルゴリズムでは行列の横方向（行方向）にはタスクの並列性があるものの、縦方向（列方向）に依存関係があるためタスク実行の逐次性が強い。そのため縦長行列に対しては性能を発揮できない。これに対して小行列に分割された行列を縦方向の領域（**ドメイン**）に分割し、それぞれにドメイン内で並列に行われた処理結果をマージすることで行列分解を行う手法がある（**図 1**)。ドメイン内での QR 分解（ローカル QR 分解）を 1 ステップ実行した後、ドメイン間のマージを行う。複数のプロセスがそれぞれ独自のドメインを管理しているとき、ドメイン間のマージを二分木的に行うことでプロセス間の通信回数が最小となることが示されており、通信削減アルゴリズムと呼ばれている。

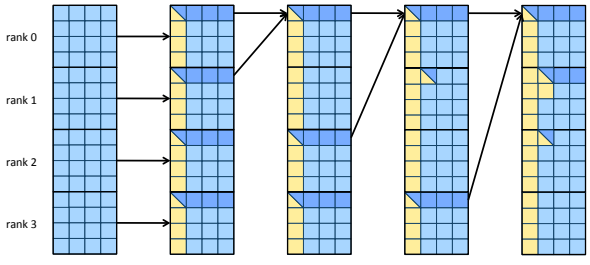

図 1 ブロックデータ分散とノード間マージ

1 ドメインを 1 プロセスに割り当て、ドメイン内のタスクをスレッド並列で処理するハイブリッド並列化し、これを共同研究拠点の FX-10 スーパーコンピュータ上に実装した。

ノード内での計算結果をノード間でマージする作業に対して幾つかの手法を試した結果、分解ステップ毎にマージ元のノードをシフトする手法が最も効果的であった（**図 2**)。図 1 では、行列が 16×4 のタイルで構成されており、すべてのステップにおいて rank0 のドメインに各ドメインのローカル QR 分解の結果がマージされる。これにより著しい負荷不均衡が生じるため良好な性能が得られなかった。

ステップ毎にマージ先のドメインを一時的にずらすことで負荷分散が可能となり（マージシフト)、更に縦方向に並列性のあるバイナリツリー方式のマージを行った結果、良好な性能を得た。

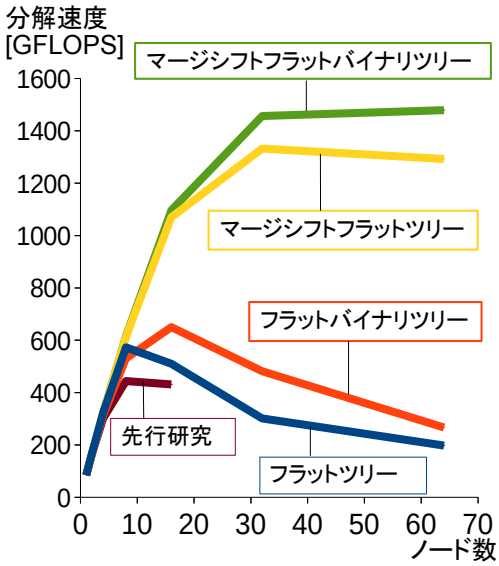

図 2 マージ方式による性能差

5. 今年度の研究成果の詳細

前年度の JHPCN 課題の成果として得られたノード間マージのマージ元のノードを分解ステップ毎にシフトする手法が効果的である理由は、負荷分散であった。今年度は、データ分散方式をブロック分散から 1D サイクリック分散とすることで、負荷分散を行う方式を評価した。**図 3**、

図 4 に評価結果を示す。

図 3 はノードあたりの行列サイズを 64000×8000 とした弱スケールでの台数効果を示す。図 4 は行列サイズを 1024000×8000 とした強スケールでの台数効果を示す。図中の「aPbT」という表記は、ノード内の MPI プロセス数が a、1MPI プロセスあたりのスレッド数が b であることを表す。今回使用した FX-10 スーパーコンピュータは、1 ノードに 16 コア CPU を 1 台搭載する。我々は MPI/OpenMP ハイブリッド並列でタイル QR 分解を実装し、ノード間通信が必要となるマージ操作を MPI で、ローカル QR 分解を OpenMP で並列化している。a×b＝16 の範囲でプロセス数、スレッド数を変化させ性能を比較した。（ただし、実装上の制限から 16P1T は実行できない）

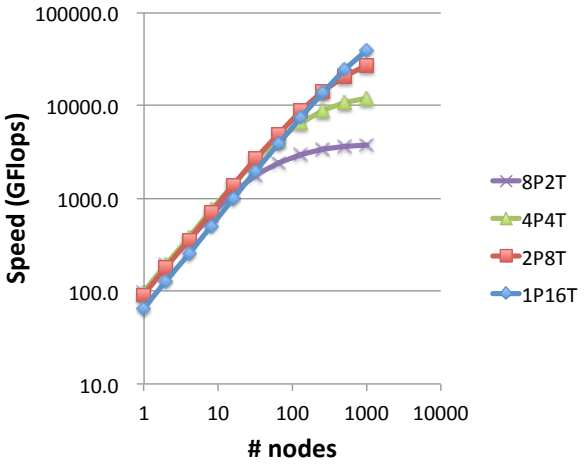

図 3 弱スケール

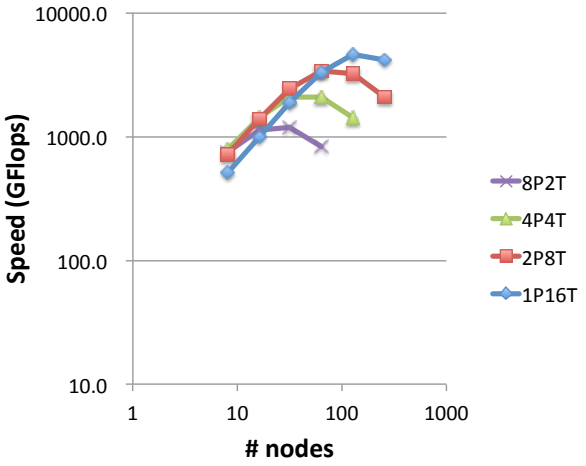

図 4 強スケール

データ分散を変更することで、マージ先のノードを変更することなく負荷分散が行われ、前年度の「マージフラットツリー」方式と同等の性能となっている。前年度の実装は、行列の形が正方行列に近くなると性能低下する特徴があったが、今年度のサイクリックデータ分散方式の実装はそれが見られない。これは前年度の手法よりも更に負荷分散が良好に行われていることが原因と考えられる。弱スケールについては概ね良好な結果であるが、強スケールについてはまだ十分な性能であるとは言えない。

プロセス数スレッド数の違いによる性能差について、ノード数が増加するにしたがって、MPI プロセス数 1 とした場合が高速となる。これは、MPI 通信回数の量が原因だと考えられる。今年度の実装においてマージ操作は図 1 のフラットツリー形状としている。8P2T ではマージの段数が大きくなり、これがノード数の増加に伴って顕著に性能に影響するものと予想している。

特定の行列サイズに対して、利用するノード数における最適なプロセス数スレッド数を得るための一つの手法として、実装の性能モデルを構築することが挙げられる。しかし、タイルアルゴリズムでは複数のタスク、通信がオーバーラップするため性能モデルの構築が難しい。そこで、呼び出し回数が大半を占めるタスクの実行時間と通信時間で構築したモデル（7 研究成果, (3), 4)、タスク、通信のオーバーラップのない 1 タイル列からなる行列の QR 分解のモデルを構築した。これらの性能モデルのパラメータの値を決定するためにはクラスタシステム向け QR 分解そのものを実行する必要はなく、実行時間が非常に短いサブプログラムを実行し、その実行時間、通信時間を測定すればよいので、モデルの構築コストは非常に低い。どちらも高い精度で実装の実行時間を予測することが可能であり、後者では利用するノード数における最適なプロセス数スレッド数を得ることが可能となった。

6. 今年度の進捗状況と今後の展望

本年度の研究計画は以下の通りであった。

1. 現在のクラスタシステム向け実装の FX-10 での性能評価
2. 分散メモリ環境におけるタイルサイズチューニング手法の検討
3. チューニング情報の収集と着目点選定
4. ノード間通信削減アルゴリズムの実装、評価

このうち、1 については終了し、更に新しい実装を追加してその評価も行った。

2 については、昨年度行った枝刈り探索の高速化を改良する作業を行っている。さらに、7 研究成果, (3), 4 で作成した性能モデルは、タイルサイズをパラメータとして持ち、複数のタイルサイズに対して予備実験を行うことでタイルサイズチューニングが可能となっている。タイルアルゴリズムではタイルサイズが重要な性能パラメータであることは広く認識されているが、実際のタイルサイズチューニングに関する報告は少なく、7 研究成果, (2), 1、同, (3), 4 は重要な研究成果である。

3 については、プログラム中にディレクティブを挿入し、実行トレースを収集することで通信処理の問題点を抽出しようと試みたが、通信処理そのものに性能低下を引き起こす問題点は見つけられなかった。

4 については、1D ブロックサイクリックデータ分散による通信削減アルゴリズム（communication avoiding QR, CAQR）を FX-10 システム上に実装し弱スケール基準で良好な性能であることを確認した。

今回の実装は FX-10 のネットワーク（TOFU）向けの特別なものではなく、他のシステムでも良好な性能を発揮すると予想される。

今回の成果で特筆すべきはクラスタシステム向け実装の性能モデルを構築したことである。2 つのモデルを作成し、タイルサイズチューニング、プロセス数/スレッド数のチューニングが可能となった。今後、国内外の会議等で詳細を報告する予定である。

今後の課題として以下を上げる。

- Intel Knights Landing 等のメニーコア環境での動的スケジューリングの有効性の検証
- GPU のアクセラレータとしての効果的な利用法の検討

7. 研究成果リスト

(1) 学術論文

なし

(2) 国際会議プロシーディングス

1. T. Suzuki, Faster method for tuning the tile size for tile matrix decomposition, Proceedings of IEEE 10th International Symposium on Embedded Multicore/Many-core System on Chip (MCSoC-16), 2016.9.

(3) 会議発表(口頭，ポスター等)

1. 高柳雅俊，鈴木智博，クラスタ型ヘテロジニアス環境におけるタイル QR 分解，日本応用数理学会 2016 年度年会，2016.9（北九州国際会議場）
2. 高柳雅俊，鈴木智博，京を用いたタイル CAQR アルゴリズムの性能評価，第 15 回自動チューニング研究会オープンアカデミックセッション（ATOS15），2016.10（山梨大学）
3. 高柳雅俊，鈴木智博，縦長行列におけるタイル CAQR アルゴリズムの性能評価，情報処理学会第 158 回ハイパフォーマンスコンピューティング研究会（HPC158），2017.03（大月ホテル和風館）
4. 鈴木智博，高柳雅俊，縦長行列に対するマルチコアクラスタ向け QR 分解アルゴリズム，第 22 回計算工学講演会，2017.05 （ソニックシティー）

(4) その他（特許，プレス発表，著書等）

なし