15-NA06

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

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行列分解に対する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り、高並列な計算環境で効果的に負荷分散が行えるアルゴリズムとして近年多くの研究が行われている。タイルサイズを小さくすることによって、計算資源に対し十分な量のタスクを生成可能である一方で、クラスタシステムでのノード間通信ではサイズの小さな通信が大量に発生し、通信遅延の原因となる可能性がある。また、LU 分解においてはタイルサイズが小さいほど丸め誤差が大きくなることが報告されており、計算環境、問題規模に応じた適切なチューニングが必要である。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、タイルサイズチューニング、ノード間通信の最適化を行うことである。また、高並列な計算環境下において高速な実装とするために、アルゴリズムの改良と共に、通信処理の最適化を行う。

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（東京大学・情報基盤センター）
  プログラム並列化、最適化
- 高坂知寛（山梨大学・大学院総合教育部）
  実証実験

2.　研究の目的と意義

科学技術計算の中で、線形方程式や、行列の固有値問題、特異値問題の数値解法が重要であることは言うまでもない。一般行列に前処理を適用することで生成された特殊な形状の行列に対して、これらの問題を高速、高精度に解くいくつかの数値解法が確立されている。しかし、これらの数値解法が高速であっても、その前処理を行う Cholesky、LU、QR などの行列分解に多くの時間を要し、これが全計算時間の大部分を占めることが多い。そのため大規模な行列分解を高速に実行するソフトウェアが求められている。

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り、高並列な計算環境で効果的に負荷分散が行えるアルゴリズムとして海外では多くの研究が行われている。

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、

- タイルサイズチューニング
- ノード間通信の最適化

を行うことである。タイルサイズを小さくすることによって、計算資源に対し十分な量のタスクを生成可能である一方で、クラスタシステムでのノード間通信ではサイズの小さな通信が大量に発生し、通信遅延の原因となる可能性がある。また、LU 分解においてはタイルサイズが小さいほど丸め誤差が大きくなることが報告されており、計算環境、問題規模に応じた適切なチューニングが必要である。また、高並列な計算環境下において高速な実装とするために、アルゴリズムの改良と共に、通信処理の最適化を行う必要がある。

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、現在クラスタシステム上の多くのアプリケーションが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

3. 当拠点公募型共同研究として実施した意義

本研究は、大規模な問題に対して、高並列な計算環境下において高速な行列分解の実装を得るために、

- アルゴリズム開発を行う代表者
- 実計算環境において実装を最適化する副代表者

の共同研究を行う。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での並列化が必要な**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メモリ階層を考慮した古典的な最適化以外に、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。

これまで、このような研究テーマに関して、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。

4. 前年度までに得られた研究成果の概要

なし

5. 今年度の研究成果の詳細

マルチコアクラスタシステム向けのタイル QR 分解のこれまでの我々の実装の特徴を以下に挙げる。

- 縦方向 1D ブロックサイクリックデータ分散
- 1 ノード 1MPI プロセス
- OpenMP スレッド化（ノード内）
- 動的タスクスケジューリング（ノード内）
- 通信専用スレッド(1)と計算用スレッド(他)

この中でノード内の OpenMP スレッドを動的にスケジューリングする機構は、タスクキューとプログレステーブルから成るオリジナルの実装であり、スレッドの同期処理が多用されているもののタイルアルゴリズムの実装として高い性能を発揮できることが示されている。他のスーパーコンピュー

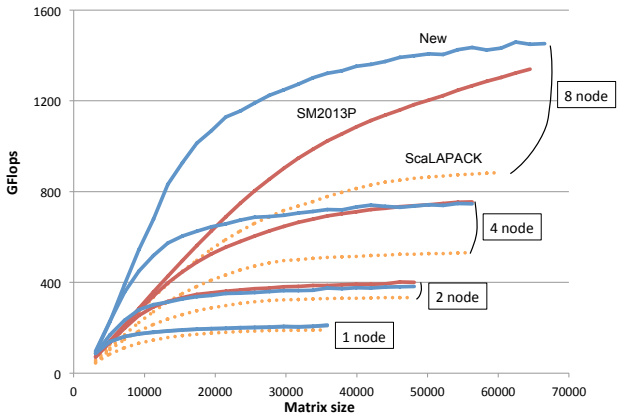


図 1 タイル QR 分解の性能（青線が研究開始時の実装の性能）

タ環境における最適化により、通信処理を最適化し、本研究を開始する時点で少ないノード数ではスケールする実装が得られていた（**図 1、国際会議プロシーディングス 1**）。この実装に対して、共同研究拠点の計算資源上で評価、最適化を行う。

以下では、これまで本年度に行った研究の 3 つの部分について報告する。

I. 縦方向の並列化（国内会議発表 1）

行列分解に対するタイルアルゴリズムのうち、タイル QR 分解、部分ピボット選択付きタイル LU 分解は行列の縦方向のタスクに依存関係がある。そのため、特に縦長の行列に対して高い性能を発揮しにくい。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 2**）。

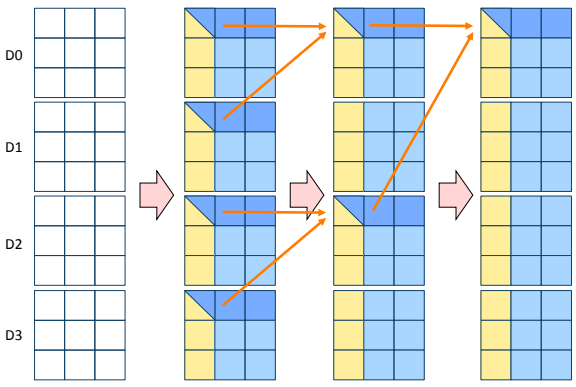


図 2 ドメイン分割とマージ操作

共同研究拠点の FX-10 スーパーコンピュータ上にこれを実装した。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を行った（後述）。この実装方式では、マージの操作にドメイン間通信が必要であり、これが性能のボトルネックとなる。

マージ操作に対して幾つかの手法を実装し、性能評価を行った。

1. フラットツリー：最上ドメインから下方向のドメインに向かって逐次的にマージを行う。
2. フラットバイナリツリー：ドメインを複数のグループに分け、グループ内でフラットツリーマージを行った後に二分木的にマージを行う。
3. マージシフトフラットツリー：フラットツリーでは、各ステップで最上ドメインに向かってマージをしていたが、マージシフトフラットツリーでは、ステップが進む毎にひとつ下のドメインに向かってマージを行う。これにより、最上ドメインを扱うプロセスへの負荷の集中を避けることが出来る。
4. マージシフトフラットバイナリツリー：フラットバイナリツリーにおけるドメイングループ内のフラットツリーマージをマージシフトフラットツリー方式で行う。

上記の手法について、行列サイズ 96,000×12,000 の縦長行列の QR 分解を 1 から 64 ノードで行い、強スケールの並列性能を調査した結果を**図 3** に示す。単純なバイナリツリー方式のドメイン間マージ（先行研究）に比べて、マージシフトフラットバイナリツリー方式は格段に性能向上していることが分かる。16 ノード以降ではやや速度向上が鈍くなるものの、16 ノードで理想性能の 80%を達成している。

II. ハイブリッド並列化の性能評価（国内会議発表 1）

上述の縦方向の並列化を行った実装において、FX-10 の 1 ノード上で、MPI プロセス数を変化させた性能評価を行った。ここで、各プロセス内のス

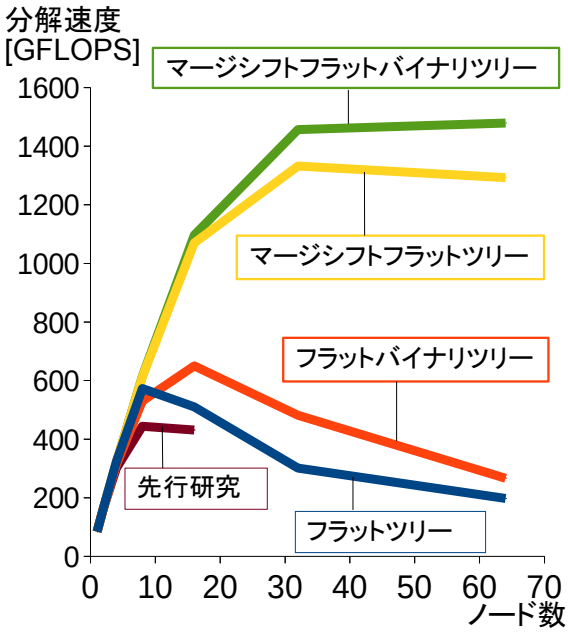


図 3 各マージ方式の性能比較

レッド数=16/MPI プロセス数とした。分解対象の行列は行数：列数=10:1 の縦長行列である。

図 4 より、MPI プロセス数が少ない場合、縦方向分割を行わない場合（block）よりも分割をした場合の方が高性能であり、逆に、MPI プロセス数が多い場合には縦方向の分割が少ない方が高性能であることが分かる。これは、ドメイン間のマージを行う際の MPI プロセス間通信が大きなボトルネックとなっていることが原因だと考えられる。

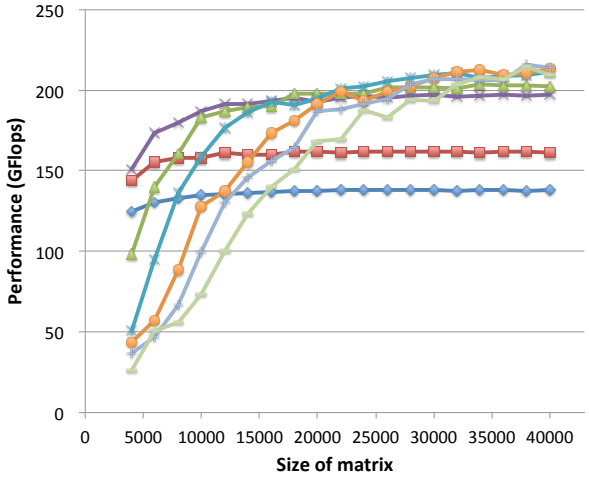


図 5 枝刈り探索（8 つの候補点について 4000 から 40000 までタイル QR 分解を実行）

（ここまでの内容をまとめたものが図 3 の「先行研究」。中間報告後にこの部分について改良を行った）

III. タイルサイズチューニング（国内会議発表 3）

タイルアルゴリズムにおいてタイルサイズは実行時に調整可能な性能パラメータであり、そのチューニングは非常に重要である。また、タイル内のタスクはブロック化されており、そのブロック幅（**内部ブロック幅**）も重要な性能パラメータである。タイルサイズを b、内部ブロック幅を s として問題サイズに応じて適切な組(b, s)を選ぶことが求められる。これに対して Agullo 等は**参考**[1]で(b, s)の枝刈り探索を行った。

タイル QR 分解に対する枝刈り探索では、複数の組(b, s)に対して網羅的に QR 分解を実行するのではなく、主要な小タスクに対して組(b, s)の探索を行う。主要な小タスクの実行時間は QR 分解と比較して非常に短い。主要な少タスクが高速に実行できる組(b, s)をいくつか抽出し、これを最適パラメータの候補とし、この候補についてのみ QR 分解を実行して問題サイズに応じた最適な組(b, s)を選択する。以上が枝刈り探索である。（**図 5**）

枝刈り探索では、最適パラメータ候補の抽出は比較的高速に行えるが、その後、想

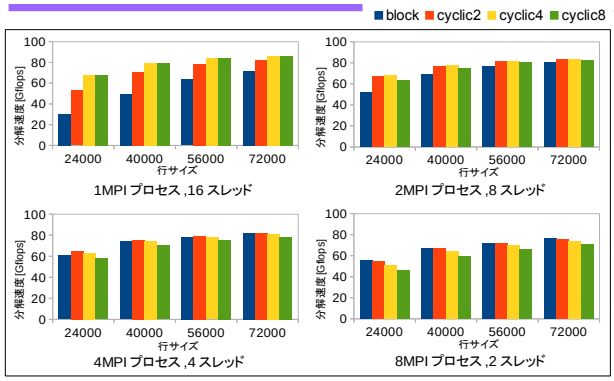


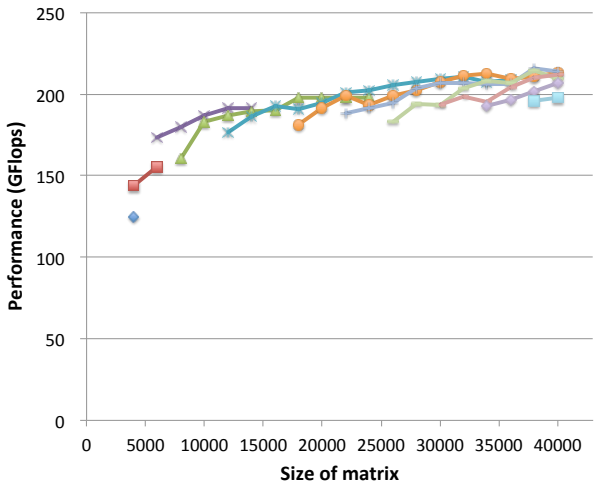
図 4 縦方向並列化実装の性能

定される問題サイズの区間で、すべての候補に対してタイル QR 分解を実行する段階で非常に時間がかかる。

タイルアルゴリズムにおいて、タイルサイズを適切に選択することで、実行環境に応じたタスク

数を生成できることを既に述べた。我々はタスク数の指標を導入し、これを一定の範囲内とするようなタイルサイズの範囲を問題サイズに応じて導出することで、特定の問題サイズにおいて試行されるパラメータ候補を削減することに成功した。（**図 6**）これにより、組(b, s)のパラメータ探索時間を半分以下とすることができた。

図 6 候補点の削減（各行列サイズにおいてタイル QR 分解を実行する候補点が削減された）



参考[1] Agullo, E., Hadri, B., Ltaief, H., Dongarra, J.: Comparative study of one-sided factorizations with multiple software packages on multi-core hardware, In Proceedings of SC'09: International Conference for High Performance Computing, Networking, Storage and Analysis (2009).

6. 今年度の進捗状況と今後の展望

申請時の研究計画のうち、タイルサイズチューニングに関しては以下の通り。

1-1 タイルサイズの実行時間への影響評価

1-2 タイルサイズの誤差への影響評価

1-3 タイルサイズの通信への影響評価

ノード間通信の最適化に関しては以下の通り。

2-1 現在のクラスタシステム向け実装の FX-10 での性能評価

2-2 ノード間通信の最適化

2-3 ノード間通信削減アルゴリズムの開発

現在まで 1-1 は完了、1-2 は途中、1-3 は未着手、2-1、2-2、2-3 は完了である。特に 2-2 により、FX-10 64 ノードまでで強スケールするクラスタシステム向け実装が得られた。

本申請課題を終えて、ドメイン間通信（マージ）について新たな着想が得られ、今後、これについても共同研究拠点の設備を使用した実装、評価を行いたい。

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, Improved internode communication for tile QR decomposition for multicore cluster systems, Proceedings of IEEE 29th IEEE International Parallel & Distributed Processing Symposium (IPDPS2015), pp. 1214 - 1220.

(3) 国際会議発表

なし

(4) 国内会議発表

1. 高坂知寛，鈴木智博，タイル CAQR の MPI/OpenMP ハイブリッド並列化，日本応用数理学会 2015 年度年会, 2015.9（金沢大学）.
2. 高柳雅俊，鈴木智博, CPU/GPU 混在環境における再帰的タイル QR 分解，日本応用数理学会 2015 年度年会, 2015.9（金沢大学）.
3. 鈴木智博，共有メモリ環境上でのタイル QR 分解のタイルサイズチューニング，情報処理学会第 151 回ハイパフォーマンスコンピューティング研究会, 2015.9（那覇市沖縄産業支援センター）.

4. 高柳雅俊，鈴木智博, CPU/GPU 混在環境における再帰的タイル QR 分解の動的スケジューリング実装，日本応用数理学会 2016 年度研究部会連合発表会, 2016.2（神戸学院大学）.

(5) その他（特許，プレス発表，著書等）

なし