TSUBAME 共同利用　平成 27 年度　学術利用　成果報告書

利用課題名　ポストペタ時代の大規模並列数値計算のための技術開発

英文：Research and Development on Large Scale Parallel Numerical Computations in the Post-Peta Era

利用課題責任者
Reiji SUDA

所属
Graduate School of Information Science and Technology, the University of Tokyo
URL http://sudalab.is.s.u-tokyo.ac.jp/~reiji/sudalab.html

邦文抄録（300 字程度）

TSUBAME，京コンピュータなど，ペタフロップスを達成するスーパーコンピュータが広がってきている．今後さらなる性能向上が期待されているが，その性能を十分に引き出すためには 100 万コア以上の並列性，深いメモリ階層，ネットワークの遅延などを手なずける必要がある．本研究ではこのような視点から，計算科学と計算機科学の融合・協力により次世代の超並列超高性能計算科学ソフトウェアの構成方式，アルゴリズム，実装技術を開発してゆく．本稿では，TSUBAME を用いたブロック化チェビシェフ基底共役勾配(BCBCG)法、Tall Skinny QR (TSQR)，通信削減行列冪カーネル（CA-MPK）の実装について報告する．

英文抄録（100 words 程度）

Supercomputers over peta-flops are getting widespread, such as TSUBAME and K-computer. The progress is expected to continue in the following years, but to attain their highest performance, we need to tame several problems such as high parallelism of million order, deep memory hierarchy, network latency, and so forth. In this research, in a collaboration of computational science and computer science, we are researching on construction methodology, algorithms, implementation techniques of extremely parallel high performance computational science software of the next era. In this report, research on Block Chebyshev Basis CG (BCBCG) method, Tall Skinny QR (TSQR), Communication-Avoiding Matrix Powers Kernel (CA-MPK) on TSUBAME.

Keywords: 5つ程度

Post-peta supercomputer, communication avoiding algorithms, block CG method, TSQR, CA-MPK

背景と目的

半導体技術の進歩により，TSUBAME，京コンピュータなど，ペタフロップスを超えるスーパーコンピュータが増えてきている．今後はさらに，2020 年ごろと言われるエクサフロップス計算機に向けて研究が進められている．エクサフロップス級の計算機では，その高い性能を高い並列性と深い階層性によって生み出すものと考えられている．エクサ級の性能を得るには，たとえば 100 程度の SIMD 並列性を持つコアが 100 万コア必要である．これは汎用コアではなくアクセラレータである可能性も高い．また，メモリ周りの性能維持や低消費電力化のために深いメモリ階層が用いられると考えられる．本研究では，特に計算科学をターゲットとして，次世代の超並列計算機のためのソフトウェア技術を研究している．本稿では，ブロック化チェビシェフ基底 CG（BCBCG）法、Tall Skinny QR（TSQR），通信削減行列冪カーネル（CA-MPK）の TSUBAME における実装について報告する。

CG 法は Krylov 部分空間法のひとつであり，対称正定値な疎行列を係数とする連立一次方程式の解法として広く用いられている．CG 法を分散メモリ型のスーパーコンピュータに実装すると，疎行列ベクトル積と内積において通信が必要となり，バンド幅と遅延の両面で通信オーバーヘッドがかかる．プロセッサ数が一段と増加したポストペタの計算機では，通信オーバーヘッドが非常に重要な問題になると考えられている．これを解決するために提案されているのが s ステップ法であり，これは CG 法の s 反復分に含まれる通信を一度にまとめて行うものである．しかし，s ステップ法には数値的な安定性という課題がある．我々は数値的に安定なアルゴリズムとし

て，ブロック化 Chebyshev 基底 CG 法（Block Chebyshev Basis Conjugate Gradient: BCBCG 法）を提案している．本稿では TSUBAME での実装を報告するが、ブロック化の効果は高くなく、右辺行列の生成に工夫が必要と思われる。また、BCBCG 法の数値的安定性を改善するために QR 分解を利用することを想定して、通信削減された QR 分解アルゴリズムである Tall Skinny QR（TSQR）を TSUBAME に実装した。Scalapack の QR 法よりも高いスケーラビリティが得られた。また、BCBCG 法の内部で必要となる通信削減行列冪カーネルとして、PA1 と呼ばれるアルゴリズムの実装に取り組んでいるところである。

概要

本プロジェクトは、ナノサイエンス，バイオインフォマティクス，コンピュータサイエンスの研究者の協力により，日本最大の GPU 搭載スーパーコンピュータである TSUBAME を用いて、共同研究により次世代の計算機のためのアルゴリズム、並列化手法，最適化，プログラミングモデル，アプリケーションの新たな展開に関して研究を行っている。以下では、平成27年に行った，通信削減アルゴリズムの成果について報告する。

まず、通信削減 CG 法の研究概要について説明する。従来型の CG 法では，Krylov 部分空間を拡張するのに Arnoldi 法を用いて1反復あたり1次元を拡張している。我々の Block Chebyshev Basis CG (BCBCG)法では，Chebyshev 多項式を用いて数値的に安定に s 次元を一度に拡大する。これは s ステップ法といて知られる方式であり，これにより大域的な集団通信を削減できる。しかし、この方法は Arnoldi 法によって探索方向を直交化する従来の CG 法のアイデアと一致しない。そこで，我々はブロック化によりこの問題を軽減することを提案している。ブロック化によっても1反復で拡大する Krylov 部分空間の次元を増加させることができるからである。

次に、Tall Skinny QR (TSQR) の概要について説明する。TSQR は Communication-optimal Parallel and Sequential QR and LU Factorizations, SIAM Journal on Scientific Computing, 2008 で提案されている通信削減 QR 分解のアルゴリズムである。通信削減 CG 法の数値安定性と収束の速度を維持するため、QR 分解のような方法は必要である。しかし、従来の QR 分解アルゴリズムは、並列化された後、集団通信の数は O(n) である（n は行列の行また列の数である）。よって、そのような QR 分解を使うと、通信削減 CG を使う意味がなくなってしまう。TSQR は 2 分木の Point-to-Point (P2P) の通信パターンで，プロセス間の情報を同期する。これにより、全部の通信のオーバーヘッドは1回の集団通信のオーバーヘッドと同等にすることができる。我々は，TSUBAME2.5 において Intel MKL を利用した TSQR 法を実装した。

最後には、行列冪カーネルの実装の進捗について、説明する。BCBCG 法では Chebyshev 基底も生成するので，行列冪カーネルも必要である。我々は、Avoiding communication in sparse matrix computations、IPDPS 2008 で提案されている Communication-avoiding Matrix Powers Kernels (CA-MPK) の PA1 を TSUBAME2.5 において実装した。原則的に CA-MPK PA1 は一回の集団通信で P0(A)x, P1(A)x,…, Pk(A)x を計算できる。k は行列多項式の次数で、x はベクトルである。CA-MPK PA1 の実装は四つの部分からなる。第1に、Breadth First Search (BFS) でデータ間の依存関係を計算する。Hyper-graph を計算するアルゴリズムも提案されているが、我々の実装では通常のグラフを用いて計算する。第2に、そのデータの依存関係に基づいて，行列 A の行をプロセス間で分散する。第3に、もとのプロセスで計算できる部分を計算して、同時に、他のプロセスに依存しているデータと依存されているデータをプロセス間で P2P 通信，あるいは一回の集団通信で交換する。第4に、もとのプロセスに残っている未処理データの処理を完成する。CA－MPK PA1 の実装は従来の MPK の実装より複雑で，ソースコードの行数は大幅にも増えている。高い性能を出すためには、十分なチューニングが必要である。

結果および考察

まず，BCBCG 法について実験の結果および考察を説明する。実験において使用した行列は The

University of Florida Sparse Matrix Collection からの bmw7st_1 という行列である。行数と列数は 141,347 で非零元素の数は 7,318,399 である。CG 法と通信削減 CG 法の Tsubame2.5 における性能評価を表1に示す。表1に示している実行時間（sec.）は完全に収束した時間ではなく、残差（residual）が 2,000 程度までの実行時間である。m はブロック数で、s はステップ数である。表 2 は残差が 2,000 程度まで収束するまでの反復回数を示す。表3はそれぞれの solver が1反復にかかった時間を示す。

表1から、512 プロセスまで、CG 法は BCBCG 法より速いことが分かる。1024 プロセスから BCBCG－m1s5 は CG 法より早くなった。原因は、プロセスの数が多くなる場合、使ったノードの数も増えるという点にある。ノードが増える時、通信オーバーヘッドも増え、通信削減 CG 法のメリットも見える。だが、BCBCG-m5s5 は BCBCG-m1s5 より3倍ぐらい遅い。m の数が増えると、計算に参加した行列のサイズも増える。そのゆえ，ブロックのサイズは収束にあまり役立たない場合は、m の数が増えれば増えるほど性能は悪くなる。このことは表2に見える。表2に示すように m の値を大きくしても収束に全然利かない、むしろ、悪い点があると見える。ブロックという技術は入力となる右辺行列の性質によって大きく左右される。この実験に使った右辺行列はランダムで生成されたものであった。右辺行列を工夫することでmの効果を高める技術の開発もこれからの重要な課題である。

表3から、BCBCG 法の1反復のかかった時間は CG 法より多いことがわかる。それは、通信削減法の共通している問題である。通信の数を減らすため、計算量は増えるようになる。だが、通信削減法の収束の速度は CG 法より速いから、通信オーバーヘッドは大きい場合、計算量のオーバーヘッドが補償できる。なお、これまでに実装した BCBCG 法は十分にチューニングされていない。今後、BCBCG の実装の性能をチューニングする予定である。

	128	256	512	1024
CG	4.53	4.77	5.81	105.51
BCBCG-m1s5	11.50	14.14	19.76	51.50
BCBCG-m5s5	53.88	52.41	60.40	185.38

表1．CG 法と通信削減 CG 法の性能、単位は sec. 列はプロセス数.

	128	256	512	1024
CG	200	200	201	201
BCBCG-m1s5	43	43	43	42
BCBCG-m5s5	44	45	45	44

表2．残差が 2,000 になるまでの反復の回数

	128	256	512	1024
CG	0.022	0.024	0.029	0.525
BCBCG-m1s5	0.267	0.329	0.460	1.226
BCBCG-m5s5	1.224	1.165	1.342	4.213

表3．１反復の平均時間

実験をした時，我々はプロセスの数が多い場合、測った時間が不安定にある現象があった。測った時間の最小値は表1に記入したが、最大値は表4に載っている。その原因は、ネットワークの性能が変わっていたと見られる。ネットワークの性能の不安定性も通信削減を提案した原因の一つである。通信の数をできるだけ減らせるとその不安定性は solver に与える影響も減らせられることからである。

	512	1024
CG	323.70	211.91
BCBCG-m1s5	45.53	208.77
BCBCG-m5s5		217.38

表4．測った時間の最大値

次に、TSQR の性能を評価する。図１に Tsubame2.5 において、我々は実装した TSQR 法の性能と Intel の MKL に実装した Scalapack QR の性能を示す。MKL の Scalapack QR の実装の性能は，最初に我々の実装した TSQR より速かったが，プロセス数が 128 を越えた後、プロセス数を増やすほど悪くなる。原因は，MKL の Scalapack QR の実装にプロセス数が増えた時、集団通信のオーバーヘッドも増えるためである。だが、我々の実装した

TSQR 法は、プロセス数が 2048 まで、よい性能を維持できる。TSQR 法の実装では，主に P2P 通信を使った。従って，通信オーバーヘッドはプロセス数と一緒に増えることはない。

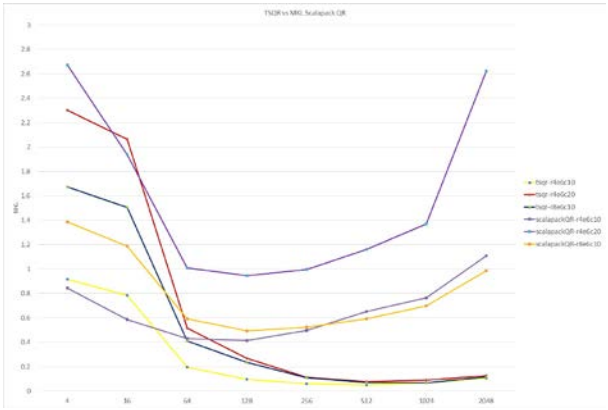

図 1. TSQR と MKL Scalapack QR 性能比較。縦軸は性能，単位は sec. 横軸はプロセスの数。r は行列の行数，c は列数を示している

我々は CA－MPK PA1 も実装した。CA－MPK PA1 法は計算量が普通の MPK より増え、メモリも多く必要である。したがって、MPI+X のような hybrid programming という技術を使わなければ性能が出せないと予想される。我々は MPI＋OpenMP という組み合わせを試した。Tsubame2.5 に OpenMPI, MPICH2 と MVAPICH の 3 つの MPI バージョンがある。その中、Tsubame2.5 の OpenMPI では multithreading の機能が使えないようである。我々の MPICH2 と MVAPICH でもまだ成功していない。原因はまだ調査中であり、目下努力している。

まとめ、今後の課題

本研究では、次世代ポストペタスーパーコンピュータにおける計算科学の手法開発を行っている。本稿では、ブロック化した Chebyshev 基底 CG 法による通信削減と,通信削減 CG 法にとって重要通信削減カーネルについて報告した。

BCBCG 法では，Chebyshev 基底のサイズに比べてブロックサイズの効果は小さい。今後はブロック化サイズの効果を高める手法の開発に取り組みたい。そして、実装の性能をチューニングしたい。

TSQR 法では，今の実装の性能のスケーラビリティが高いが。通信パターンの変更や multithreading による性能の最適化がまだ実施できていない。これらの性能の最適化もこれから取り組みたい課題である。

CA-MPK については，hybrid programming という技術を使っての実装に取り組みたい。そして、実装の性能の最適化もこれからの課題である。