TSUBAME 共同利用 平成 26 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 須田礼仁
Reiji SUDA

東京大学 情報理工学系研究科
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，通信削減 CG 法のブロック化と，TSUBAME を用いた全点対最短経路問題の並列解法について報告する．

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 version of communication avoiding CG method and parallel solver of all-pairs shortest path problem on TSUBAME.

Keywords: 5つ程度

Post-peta supercomputer, communication avoiding algorithms, graph algorithms, all-pair shortest path problem

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，ブロック化した通信削減 CG 法と，TSUBAME を用いた全点対最短経路問題の並列解法について報告する．

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．BCBCG 法の通信を真に最小にするには，通信削減された行列分解アルゴリズムと通信削減された行列冪カーネルが必要である．我々はこれらの実装を進めている．

また，従来の並列計算においては並列化効率が重要視されてきた．そのためにはプロセッサあたりの問題サイズを大きく保つ必要があり，弱スケーリングが広く採用されてきた．しかし，TSUBAME や京コンピュータをはじめとする大規模並列マシンの登場によってこの考えには限界がきた．さらに応用分野においても，非常に多くのプロセッサを用いて与えられた問題サイズの最短時間での解決，すなわち強スケーリングが要請としてある．非常に多くのプロセッサを用いればプロセッサ１つあたりの問題サイズは非常に小さくなり，計算自体にかかる時間に対してプロセッサ間の通信にかかる時間の割合が大きくなってくる．よって通信にかかる時間をいかに減らすかということを考えることは重要である．本研究では全対最短経路問題を例題として，TSUBAME における強スケーリング実装を行った．通信オーバーヘッドを通信隠蔽によって軽減し，ナイーブな実装では達成できなかったプロセッサ数の増加とともに実行時間が小さくなるという結果および，どのような実装が問題サイズに対して適しているかという知見を得た．

概要

本プロジェクトでは，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 26 年度に行った，通信削減 CG 法の成果と，全点対最短経路問題の強スケーリング実装について報告する．

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，1 反復で 1 次元を拡張している．我々の 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 ステップ法とし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．

しかしブロック化の効果，特に反復回数が少ないところでの効果は，入力となる右辺行列の性質によって大きく左右される．このため，QR 分解のような方法により，ブロック探索方向ベクトルを改善する必要が，しばしばある．この際，通信削減という性質を維持するためには，この QR 分解などに TSQR（Tall Skinny QR）法などの通信削減アルゴリズムを用いる必要がある．我々は，TSUBAME2.5 において Intel MKL を利用した TSQR 法を実装した．

また，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．行列冪カーネルの実装は係数行列の非零パターンにより最適な実装方式が大きく異なる．現在行列冪カーネルの実装は開発中であり，今後これに注力して進める予定である．

次に，全対最短経路問題の並列解法について報告する．我々の実装は Buluc らによる Solving path problems on the GPUs, Journal of Parallel Computing, 36, pp. 241-253 (2010) で提案されている 1GPU 実装をもとにしている．基本的なアルゴリズムは以下のようになる．

Function APSP(A)
if n < b
  then
    Floyed-Warshall 法で計算する
  else
    A を 2 等分して $\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ とする

    $\mathrm{A}_{11} \coloneqq \mathrm{APSP}(\mathrm{A}_{11})$
    $\mathrm{A}_{12} \coloneqq \mathrm{A}_{11}A_{12}$ //*1
    $\mathrm{A}_{21} \coloneqq A_{21}A_{11}$ //*2
    $\mathrm{A}_{22} \coloneqq \mathrm{A}_{22} \oplus A_{21}A_{12}$ //*3

$A_{22} \coloneqq \mathrm{APSP}(A_{22})$
$A_{21} \coloneqq A_{22}A_{21}$ //*4
$A_{12} \coloneqq A_{12}A_{22}$ //*5
$A_{11} \coloneqq A_{11} \oplus A_{12}A_{21}$ //*6
end if
end function

このアルゴリズムでは,コード中で*1 の計算と*2 で示した計算,および*4 と*5 で示した計算が自明に並列化できる．さらに，APSP を完了した行列 A については冪等性 $A^2 = A$ がなりたつことに注意すると，*3 における A_{21} は *2 で計算した結果ではなく，*2 で計算する前の結果でもよいことがわかる.これにより，*2 と*3 も並列化することができる．これらの考察に基づき，我々の 2GPU 実装では，*1 と*3 を 1 つの GPU が計算している間にもう 1 つの GPU が*2 を計算し，また，*4 と*5 を 1 つの GPU が計算している間にもう 1 つの GPU が*6 を計算するという方式を採用した．これにより，通信遅延が隠ぺいできるようになった．

また，複数のノードを用いた実装では 2 分割ではなく$\sqrt{p} \times \sqrt{p}$ の分割を用いている．そして，1 つの GPU に 1 つの MPI プロセスを割り当て，2 次元正方グリッドで分割することによって並列化を施した．複数ノード実装についても 2GPU の場合と同じ原理を用いて通信隠蔽を施した．この方式ではプロセッサの相当数が遊ぶため，並列化効率は高くないが，できるだけ多数のノードに計算を分散させることにより，強スケーリングを目指した．

結果および考察

まず，BCBCG 法については，概要で説明したように，TSQR 法は実装が完了したが，行列冪カーネルについての実装を進めているところである．これまでに Matlab を用いた小規模計算で数値的な性質を分析しているので，ここではそれについて報告する．

BCBCG 法では，CG 法に比べて少ない反復回数で収束することができている．すなわち，ブロック化により，分散メモリ並列計算器における大域的な通信を削減することができる.ただし計算量は CG 法よりも多く，将来のスーパーコンピュータにおいて，通信コストに比べて計算コストが時間・電力ともに劇的に少ない場合において有効である．

丸め誤差に伴う安定性の問題を軽減するには，Chebyshev 基底のサイズとともに，ブロック化サイズもてきせつに選択してやる必要がある.Chebyshev 基底サイズの増加とブロック化サイズの増加の 2 つによる反復回数の削減程度を比較すると，多くの場合において Chebyshev 基底サイズの増加のほうが効果が大きい．しかし，右辺行列を適切に選ぶことにより，ブロック化サイズの効果を高めることができることが観測されている．

TSQR の実装においては，プロセッサ数の適切な選択が重要であることがわかっている．すなわち，プロセッサ数を多く選びすぎると，TSQR の性能が急速に低下する．最適なプロセッサ数の自動的な決定は今後の課題である．

次に，全対最短経路問題の並列解法について報告する．1GPU 実装（1GPU と称する）と，通信隠蔽を用いない 2GPU 実装（2GPU と称する）と，2GPU で通信隠蔽を用いてさらに問題サイズが 512 以下になると 1GPU に切り替えるような実装（2GPU opt と称する）の 3 つの実装につき，を問題サイズごとに実行時間(ms)を計測したものを表 1 に示す．

頂点数	1GPU	2GPU	2GPU opt
1024	11.951	18.950	11.000
2048	44.166	75.269	35.776
4096	249.25	239.71	178.98
8192	1784.1	1403.3	1215.0
16384	13831.5	9827.3	9275.3

表 1. ノード内並列化による性能．単位は ms．下線は，実行時間最小のものを示す．

表 1 から，2GPU で通信隠ぺいを行った実装が高速であり，通信削減の効果が高いことがわかる．

また，複数のノードを用いた実装について通信隠蔽を用いた場合の性能を，行にプロセッサ数(p)，列に頂点数(n)をとり，実行時間(s)をまとめたものが表 2 である．表 2 において$n/\sqrt{p}$の値が 1 プロセッサあたりに割り当てられる問題サイズを表

しているが，これが 1024 以上の場合においては，プロセッサ数の増加に伴って実行時間がより短くなることがわかる．1024 未満になる場合にそうならないのは，1 プロセッサあたりの問題サイズが十分な大きさが無いため GPU の計算効率が低下してしまうためである．

p \ n	2048	4096	8192	16384
4	0.05285	0.22647	1.36530	9.63585
16	0.08674	0.17608	0.57560	3.17227
64	0.22080	0.23008	0.35139	1.20455
256		0.52978	0.77171	0.96503

p \ n	32768	65536	131072	2622144
4	72.3535			
16	21.6361	161.133		
64	6.72522	45.6037	338.770	
256	2.52223	13.8227	93.5249	694.174

表 2. ノード間並列化による全点最短経路問題の並列解法の性能．行はプロセス数，列は問題サイズを示す．単位は ms．下線は，各サイズで実行時間最小のものを示す．

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，全対最短経路問題の強スケーリング実装について報告した．

BCBCG 法では，ブロック化により収束性を安定化することができた．ブロック化サイズを増加させることにより，数値的な安定性を損なうことなく，大域的な通信のオーバーヘッドを削減することができる．しかし，Chebyshev 基底のサイズに比べてその効果は小さい．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．また，BCBCG 法の並列実装に必要な TSQR 法は実装したが,行列冪カーネルは現在進行中である．TSQR の最適化とともに，行列冪カーネルに取り組む予定である．

全対最短経路問題に対する TSUBAME 2.5 上での並列実装では，1 ノード上の 2 つの GPU を使用した実装と，複数のノードを使用した実装の 2 つを実装した．そしてそれぞれの実装について，冪等性を用いて通信隠蔽を施した実装も提案した．実験結果としては全対最短経路問題に対して、頂点数が 512 以下であれば 1 つの GPU を用いた実装が，頂点数 1,024 以上 4,096 以下では 1 ノード上の 2 つの GPU を用いた実装が，それ以上の頂点数の場合には複数のノードを用いた実装がそれぞれ最短時間を記録した．今後の課題としては，行列積を GPU 上で計算する際に，行列の大きさがある程度大きくないと GPU での計算の効率が著しく下がってしまうということが分かっているので，小さなサイズの行列に特化したような GPU 実装を考えることで全体の実行時間を小さくすることに寄与できると思われる．