

卒業研究報告書

題目

BSP*モデル上での行列積アルゴリズム

指導教員

石水 隆 助教

報告者

04-1-47-041

山田直人

近畿大学工学部情報学科

平成 21 年 1 月 31 日提出

概要

本研究では、BSP*(Bulk-Synchronous Parallel)モデル上で行列積を行う効率の良いアルゴリズムの提示を行う。

BSP*モデルは Valiant により提案された並列計算モデルである BSP モデルの拡張モデルであり、Bäumker らによって提案された。BSP モデルは並列計算において重要とされる通信コストを同期周期 L 、通信命令実行時間 g といったパラメタにより表すことを可能にしたモデルである。そして BSP*モデルでは、通信パケットサイズを表すパラメタ B を導入することにより、より実際に即したアルゴリズムの計算量の検証が可能となっている。

本研究では、BSP*モデル上で、入力行列の各要素のプロセッサへの割り当て方及び解行列の各プロセッサの計算担当範囲の割り当て方を検証することにより、行列積を求めるのに最適なアルゴリズムの検証を行う。また、その分割パターンにおいて最適となるプロセッサ数を求める。

目次

1	序論	1
1.1	並列アルゴリズム	1
1.2	並列計算モデル	1
2	BSP モデルと BSP*モデル	2
3	BSP*モデル上で行列積を求める並列アルゴリズム	4
4	結論	9
	謝辞	10
	参考文献	11

1 序論

1.1 並列アルゴリズム

アルゴリズムとは、ある特定の目的を達成するために必要な作業の倫理や手順を述べたものである。この手順は決まっており、その指示通りに従うことが必要である。

ある問題を解くアルゴリズムの評価において、計算時間を基準に評価する方法がある。この方法で一番簡単なのは、実際にプログラムを書いてそれを実行させることである。しかし、複数のアルゴリズムを比較する場合、計算機等がまったく同じ環境で実行しなければならないことになる。また、計算機が変わったときに、動作の優劣が逆転してしまうことも起こり得てしまう。

そこで、アルゴリズムを評価するにあたって、計算機のモデルを考えてその上で評価する方法が用いられている。

並列アルゴリズムは、複数のプロセッサが協調してデータを処理することにより、問題を短時間で解け、またより複雑な問題を解くことを可能にするためのアルゴリズムである。このとき、通常アルゴリズムの評価に用いられる計算時間や記憶領域に加えて、プロセッサ数も評価の対象になることが多い。これは、アルゴリズムを計算機のモデル上で考えているために、膨大なメモリやプロセッサを必要とするアルゴリズムの場合、実際の計算機では事実上解けなくなってしまう。この為に、これらの基準も重要な要素となってくる。

1.2 並列計算モデル

並列アルゴリズムの設計、解析は並列計算機を抽象化した並列計算モデル上で行われる。代表的な並列計算モデルとして PRAM(Parallel Random Access Machine)^[4]、Mesh^[4]、Hyper-cube^[4]、BSP(Bulk-Synchronous Parallel)^[2]等がある。

PRAM は共有メモリ型並列計算のモデルであり、通信や同期に掛かるコストを無視でき、また、1命令ごとに全てのプロセッサで同期を取る細粒度同期式モデルである。このため PRAM 上でのアルゴリズムの設計、解析は比較的容易に行うことができる。しかし、プロセッサ能力の向上に伴い、PRAM では無視していた通信や同期のコストが実行時間全体において無視できないほど大きな割合となってきたため、PRAM 上で設計したアルゴリズムが現実の並列計算機では必ずしも効率よく実行できるとはかぎらなくなっている。また、昨今ではプロセッサが他のプロセッサと同期せずに処理を行う非同期式処理が主流となりつつある。このため、非同期式処理に対応したモデルとして注目されているのが BSP モデルである。

BSP モデルは分散メモリ型の非同期式並列計算モデルであり、通信のオーバーヘッドや同期のオーバーヘッドを考慮することができるモデルである。このため BSP モデルは多くの実在する並列計算機に対応する汎用性の高いモデルである。また、BSP*モデルは BSP モデルの拡張モデルであり、通信計算量にパケットの概念が取り入れられている。今日の TCP/IP 通信ではパケット通信が主に行われていることから、このモデルはより現実のネットワークに近いモデルであると言える。

2 BSP モデルと BSP*モデル

BSP(Bulk-Synchronous Parallel)^[2]モデルは Valiant によって提案された非同期式計算モデルであり、以下の構成要素から成る。

- 局所メモリを持つ複数のプロセッサ
- プロセッサ間の 1 対 1 メッセージ通信を行なう完全結合網
- プロセッサ間の同期を実現するための同期機構

図 1 に BSP モデルの概念図を示す。

また、BSP モデルは以下のパラメータを持つ

p: プロセッサ数

g: 通信路帯域幅: 1 個のメッセージを送信あるいは受信するためにかかる時間。

L: 同期時間、バリア同期周期: プロセッサ全体の 1 回の同期にかかる時間、これは同時に通信延期時間を表すパラメータでもある。

BSP モデル上の並列アルゴリズムの基本的命令の実行時間について、以下のように仮定されている。

- 各プロセッサは 1 単位時間に 1 内部計算命令を局所メモリについてのみ基づいて実行する。
- メッセージ 1 個の送信命令または受信命令の実行は g 単位時間で行われる。ただし、1 メッセージは 1 語からなるものとし、サイズ 1 のメッセージと呼ぶ。
- あるスーパーステップにおいて、全てのプロセッサで命令の実行を終了してから L 時間以内にバリア同期が取られ、次のスーパーステップの実行に移る。よって、あるスーパーステップにおいて、各プロセッサが高々 w 個の内部計算命令、高々 h 個の送信命令または受信命令を割り当てられた場合、そのスーパーステップの実行には $O(w+gh+L)$ 時間かかる。図 2 に BSP モデル上でのアルゴリズム実行の概念図を示す。

以降では簡単のために、各スーパーステップは内部計算命令のみ、あるいは送信命令および受信命令のみからなるとし、内部計算命令のみからなるスーパーステップの実行時間を内部計算時間、送信命令および受信命令のみからなるスーパーステップの実行時間を通信時間と呼ぶ。

BSP*モデル[3]は、Bäumker らによって提案された BSP モデルの拡張モデルであり、上記の BSP モデルのパラメータに加えて以下のパラメータを持つ

B(≥ 1): 通信パケットの最小サイズ

この拡張により、BSP モデルと比べて以下の点に変更される。

- 同じプロセッサに対する s 語のメッセージをサイズ s のメッセージとして送信または受信できる。
- サイズ s のメッセージの送信命令または受信命令の実行は $g \left\lceil \frac{s}{B} \right\rceil$ 単位時間で行われる。よって各

プロセッサが高々 w 個の内部計算命令、各サイズ s_1, s_2, \dots, s_h である高々 h 個のメッセージの送

信命令又は受信命令からなるスーパーステップの実行には、 $O \left(w + g \sum_{i=1}^h \left\lceil \frac{s_i}{B} \right\rceil + L \right)$ 時間かかる。

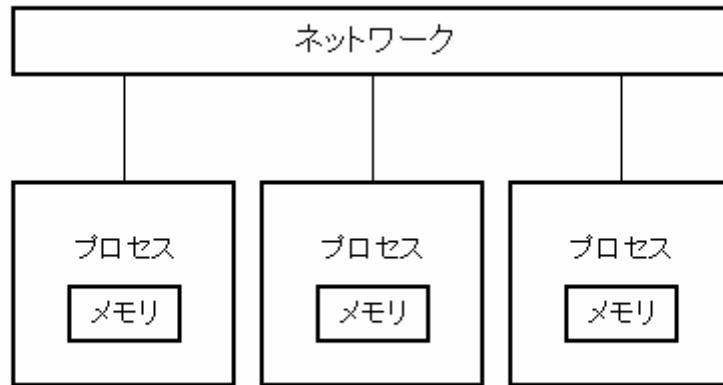


図 1 BSP モデルの概念図

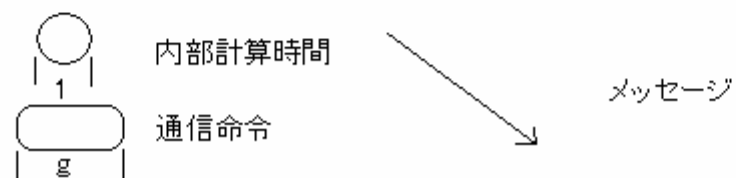
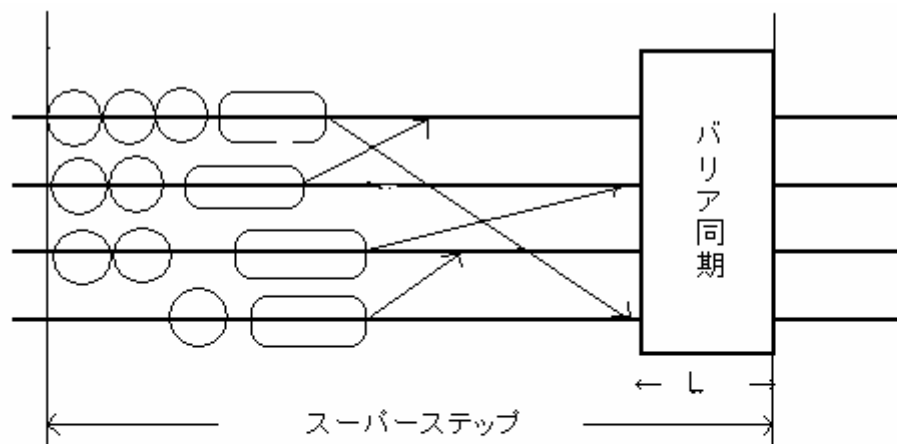


図 2 BSP アルゴリズム上での計算機の動き

3 BSP*モデル上で行列積を求める並列アルゴリズム

本研究では BSP*モデル上で $n \times n$ の行列積を求める効率の良い並列アルゴリズムを設計する。

BSP モデル上で行列積を求める場合、入力行列 A,B は p 個の部分行列に分割され、各プロセッサのメモリ上に割り当てられる。また解行列 C は p 個の部分行列に分割されて各プロセッサに割り当てられ、各プロセッサは割り当てられた部分行列の計算を担当する。

行列積を計算するアルゴリズム中で各プロセッサは、保持する部分行列の要素を、それを必要とする解行列を計算するプロセッサに送信する。このとき、入力行列 A,B データの各プロセッサへの割り当て方により、データ送信に必要となるデータの個数及び送信先のプロセッサの台数が異なる。したがって、効率の良い並列アルゴリズムを得るには入力行列 A,B のデータの各プロセッサへの割り当て方及び解行列 C の計算担当範囲の各プロセッサへの割り当て方を検証しなければならない。そこで以下には、入力行列 A,B 及び解行列 C のプロセッサへの最適な割り当て方を求める。

分割のパターンとしては図 3 の(a),(b),(c)に示す 3 通りが考えられる。

図 3 の(a)では、 $n \times n$ の行列はそれぞれサイズ $\frac{n}{\sqrt{p}} \times \frac{n}{\sqrt{p}}$ の $\sqrt{p} \times \sqrt{p}$ 個の部分行列に分割される。同様

に、図 3 の(b)ではそれぞれサイズ $n \times \frac{n}{p}$ の $1 \times p$ 個の部分行列に、図 3 の(c)ではそれぞれサイズ $\frac{n}{p} \times n$

の $p \times 1$ 個の部分行列に分割される。

入力行列 A,B 及び解行列 C がそれぞれ(a),(b),(c)の 3 通りの分割パターンがあるため、全体では $3^3 = 27$ 通りの組み合わせがある。

行列 A の要素の送信しなければならないデータ及びデータの送り先は、A の部分行列のデータの各プロセッサへの割り当て方及び C の部分行列の計算担当範囲の各プロセッサへの割り当て方のみ依存し、B の割り当て方には影響されない。同様に、B の送信が必要なデータは、B と C の割り当て方のみ依存する。また、A と B は縦横が異なるだけであり、送信が必要となるデータ数とデータの送信先数は変わらない。よって以下では、A と C の組み合わせ、すなわち $3^2 = 9$ 通りの組み合わせについてのみ検証する。

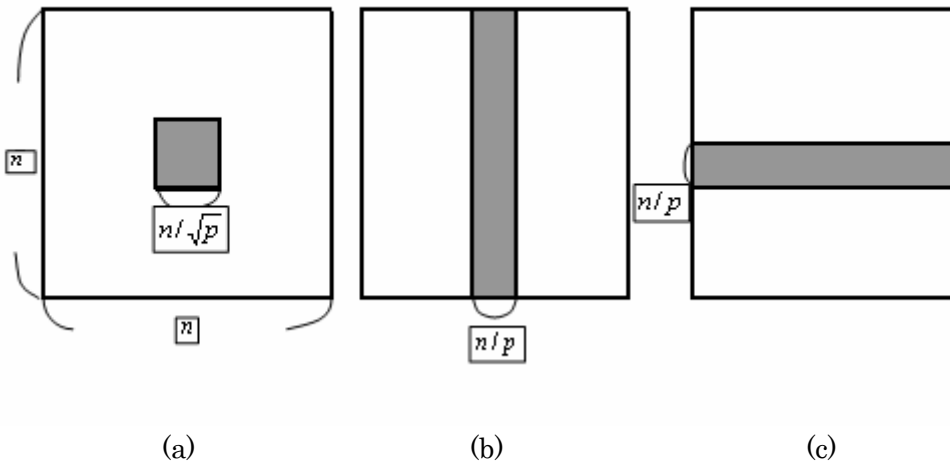


図 3 部分行列への分割の仕方

定理 1)

BSP*モデル上で行列積を解くアルゴリズムの計算量は

$$O\left(\frac{n^3}{p} + g\left(\frac{n^2}{Bp} + \sqrt{p}\right) + L\right)$$

となる。

(証明)

以下は入力行列 A の部分行列へのデータの各プロセッサへの割り当て方及び解行列 C の部分行列の計算担当範囲の各プロセッサへの割り当て方を(a),(b),(c)それぞれの分割の仕方にした場合の送信データ量及び送信先のプロセッサ台数について検証する。

(1) A のデータの割り当て方(a)、C の計算範囲の割り当て方(a)の場合

A のデータを保持する各プロセッサは、 $\frac{n}{\sqrt{p}} * \frac{n}{\sqrt{p}}$ 個のデータを \sqrt{p} 個のプロセッサに対して

送信する。よって通信計算量は

$$\begin{aligned} & O\left(g\left[\frac{n}{\sqrt{p}} * \frac{n}{\sqrt{p}} * \frac{1}{B}\right] * \sqrt{p} + L\right) \\ & = O\left(g\left(\frac{n^2}{\sqrt{p}B} + \sqrt{p}\right) + L\right) \end{aligned}$$

となる。

(2) A のデータの割り当て方(a)、C の計算担当範囲の割り当て方(b)の場合

(1)と同様に通信計算量は

$$\begin{aligned} & O\left(g\left[\frac{n}{\sqrt{p}} * \frac{n}{\sqrt{p}} * \frac{1}{B}\right] * p + L\right) \\ & = O\left(g\left(\frac{n^2}{B} + p\right) + L\right) \end{aligned}$$

(3) A のデータの割り当て方(a)、C の割り当て方(c)の場合

$$\begin{aligned} & O\left(g\left[\frac{n}{p} * \frac{n}{\sqrt{p}} * \frac{1}{B}\right] * \sqrt{p} + L\right) \\ & = O\left(g\left(\frac{n^2}{pB} + \sqrt{p}\right) + L\right) \end{aligned}$$

(4) A のデータの割り当て方(b)、C の計算範囲の割り当て方(a)の場合

$$\begin{aligned} & O\left(g\left[\frac{n}{\sqrt{p}} * \frac{n}{p} * \frac{1}{B}\right] * p + L\right) \\ & = O\left(g\left(\frac{n^2}{\sqrt{p}B} + p\right) + L\right) \end{aligned}$$

(5) A のデータの割り当て方(b)、C の計算範囲の割り当て方(b)の場合

$$\begin{aligned} & O\left(g\left[n * \frac{n}{p} * \frac{1}{B}\right] * p + L\right) \\ & = O\left(g\left(\frac{n^2}{B} + p\right) + L\right) \end{aligned}$$

(6) A のデータの割り当て方(b)、C の計算範囲の割り当て方(c)の場合

$$\begin{aligned} & O\left(g\left[\frac{n}{p} * \frac{n}{p} * \frac{1}{B}\right] * p + L\right) \\ & = O\left(g\left(\frac{n^2}{pB} + p\right) + L\right) \end{aligned}$$

(7) A のデータの割り当て方(c)、C の計算範囲の割り当て方(a)の場合

$$\begin{aligned} & O\left(g\left[\frac{n}{p} * n * \frac{1}{B}\right] * \sqrt{p} + L\right) \\ & = O\left(g\left(\frac{n^2}{\sqrt{p}B} + \sqrt{p}\right) + L\right) \end{aligned}$$

(8) A のデータの割り当て方(c)、C の計算範囲の割り当て方(b)の場合

$$\begin{aligned} & O\left(g\left[\frac{n}{p} * n * \frac{1}{B}\right] * p + L\right) \\ & = O\left(g\left(\frac{n^2}{B} + n\right) + L\right) \end{aligned}$$

(9) A のデータの割り当て方(c)、C の計算範囲の割り当て方(c)の場合

$$\begin{aligned} & O\left(g\left[\frac{n}{p} * n * \frac{1}{B}\right] * 1 + L\right) \\ & = O\left(g\left(\frac{n^2}{pB} + 1\right) + L\right) \end{aligned}$$

$\frac{n^2}{p} \gg p$ の場合、通信計算量は、(1),(4),(7)が $O\left(g\frac{n^2}{\sqrt{p}B} + L\right)$ 、(2),(5),(8)が $O\left(g\frac{n^2}{B} + L\right)$ 、(3),(6),(9)

が $O\left(g\frac{n^2}{pB} + L\right)$ となり、通信計算量は C の計算担当範囲の割り当て方に依存することが示される。

また、B のデータを送信する通信計算量は、A,C の縦横を入れ替えたものと同じ、すなわち、C の計算担当範囲の割り当てをそれぞれ(a),(b),(c)とする場合、B のデータを送信する通信計算量は $O\left(g\frac{n^2}{\sqrt{p}B} + L\right)$ 、 $O\left(g\frac{n^2}{pB} + L\right)$ 、 $O\left(g\frac{n^2}{B} + L\right)$ となる。したがって、全体の通信計算量は C の計算担

当範囲の割り当てをそれぞれ(a),(b),(c)とする場合、 $O\left(g\frac{n^2}{\sqrt{pB}}+L\right)$ 、 $O\left(g\frac{n^2}{B}+L\right)$ 、 $O\left(g\frac{n^2}{B}+L\right)$ となり、(a)とした場合が最も通信計算量が小さくなる。

Cの計算担当範囲を(a)とした場合、Aのデータの割り当て方は(a)または(c)、Bのデータの割り当て方は(a)または(b)にした場合が最も通信計算量が小さくなり、 $O\left(g\frac{n^2}{\sqrt{pB}}+L\right)$ となる。

解行列の各要素は $O(n)$ 時間で計算でき、各プロセッサは $\frac{n^2}{p}$ 個の要素の計算を行うので、解行列の

計算は $O\left(\frac{n^3}{p}\right)$ 時間で行うことが出来る。よって BSP*モデル上で行列積を解く並列アルゴリズムの計算量は

$$O\left(\frac{n^3}{p}+g\left(\frac{n^2}{pB}+\sqrt{p}\right)+L\right)$$

□

となる。

定理 2)

BSP*モデル上で行列積を解くアルゴリズムは

$$p \leq \frac{n^2 B^2}{g^2} \quad \left(\frac{n^2}{p} \geq B \text{ のとき}\right)$$

$$p \leq \frac{n^2}{\sqrt[3]{g^2}} \quad \left(\frac{n^2}{p} < B \text{ のとき}\right)$$

のとき最適となる

(証明)

並列アルゴリズムが最適となるのは、通信計算量が内部計算量よりも小さいときである。よって

$$g\left(\frac{n^2}{\sqrt{pB}}+\sqrt{p}\right) \leq \frac{n^3}{p} \quad \dots(1).$$

$\frac{n^2}{p} \geq B$ のとき、(1)式において $\frac{n^2}{\sqrt{pB}} \geq \sqrt{p}$ である。したがって

$$g\frac{n^2}{\sqrt{pB}} \leq \frac{n^3}{p}$$

$$\sqrt{p} \leq \frac{nB}{g}$$

$$p \leq \frac{n^2 B^2}{g^2}$$

$\frac{n^2}{p} < B$ のとき(1)式において $\frac{n^2}{\sqrt{p}B} < \sqrt{p}$ である。したがって

$$g\sqrt{p} \leq \frac{n^3}{p}$$

$$p\sqrt{p} \leq \frac{n^3}{g}$$

$$p \leq \frac{n^2}{\sqrt[3]{g^2}}$$

よって成り立つ

□

4 結論

本研究では BSP*モデル上で効率よく行列積を求めるアルゴリズムを提案した。

BSP*モデル上で行列積を求める並列アルゴリズムの計算量は $O\left(\frac{n^3}{p} + g\left(\frac{n^2}{\sqrt{pB}} + \sqrt{p}\right) + L\right)$ となる。ま

た、 $p \leq \frac{n^2 B^2}{g^2}$ ($\frac{n^2}{p} \geq B$ のとき)、 $p \leq \frac{n^2}{\sqrt[3]{g^2}}$ ($\frac{n^2}{p} < B$ のとき) のとき最適な並列アルゴリズムとなる。

今後の課題としては、行列以外にも 2次元配列になっているデータが存在するような問題に対するデータ分割の仕方を求めていくことが必要となる。

謝辞

本研究するにあたり、様々な助言、温かい御指導をしていただいた近畿大学工学部情報学科 石水隆先生には心より感謝申し上げます。
また、研究室の皆様には論文を書く上で助言等をしてくださり大変感謝いたします。

参考文献

- [1] 選択問題を解く BSP モデルおよび BSP* モデル上の並列アルゴリズム、電子情報通信学会論文誌 (DI)、Vol. J82 -D-1No. 4, pp. 522-542, Apl. 1999
- [2] L. G. Valiant, " A Bridging Model for Parallel Computation," , Communications of the ACM, Vol. 33, No. 8, pp. 103-111, 1990.
- [3] A. Baumker, W. Dittrich, and F.M. Heide, " Truly efficient parallel algorithms: 1-optimal multisearch for an extension of the BSP model," Technical Report TR-RSFB-96-008, University Paderborn, Department of Mathematics and Computer Science, Jan. 1996.
- [4] Joseph JáJá : "An Introduction to Parallel Algorithms" Addison Wesley, 1992