

## 1. 序論

本研究では BSP モデル<sup>2)</sup> 上で効率のよいアルゴリズムを作ることを目的としている。BSP モデルは従来の PRAM モデル<sup>1)</sup> と異なり通信時間も考慮する並列モデルである。ゆえに PRAM モデルのアルゴリズムでは BSP モデル上で効率よく働くとは限らない。そのため通信も考慮した BSP モデル用のアルゴリズムを考えることが必要となる。本研究では基本的な問題であるソートングに対して、BSP 上でそれを効率よく解く並列アルゴリズムを提案している。

## 2. 研究内容

本研究で提案する BSP 上のソートングアルゴリズムはクイックソートアルゴリズムをベースにしている。

本研究で提案するアルゴリズムは入力データを各プロセッサに均等に割り当て、各プロセッサが並列にデータを分割することにより高速化を図っている。

まずデータ全体からランダムに選んだ値を基準値とし、全てのプロセッサが共通する基準値を用いてデータを分割する。分割後、プロセッサを 2 グループに振り分け、各プロセッサは保持する基準値以下のデータを片方のグループのプロセッサへ、基準値以上のデータをもう片方のグループのプロセッサに送信する。データを受信後、各グループでそれぞれ再帰的にデータを分割していく。データの分割は、各グループに属するプロセッサ数が 1 台になるまで行い、その後クイックソートアルゴリズムを用いて逐次にソートングを行う。

基準値の選択をランダムに行うことにより、基準値選択にかかる時間を短縮でき、それにより全体の平均処理時間の短縮を得られると考えられる。また、平均的には各プロセッサに割り当てられるデータ数はほぼ等しくなると考えられるため、各プロセッサの負荷は均等となり、全体の処理時間が短縮されることが考えられる。

本研究では、上記の検討を行うために、提案したアルゴリズムの BSP 上での実行をシミュレートするプログラムを作成し、その処理時間の実験的評価も行う。

## 3. 結果・考察

表 1 に 256 個のデータに対して本研究で作成したアルゴリズムの BSP 上の計算量のシミュレーション結果を示す。

表 1: ソートングの平均計算時間 (データ数 256)

プロセッサ数	内部計算時間	通信時間	同期時間
1	24239	0	0
4	8983	371*g	6*L
16	2767	296*g	12*L

(g: 1 メッセージの送受信時間, L: 同期時間)

シミュレーション結果の通り、プロセッサ台数が増加するに従いアルゴリズムの内部計算回数は減少している。一方、送受信メッセージ数および同期回数はプロセッサ台数の増加に伴い増加している。従って、1 メッセージ辺りの送受信時間および同期時間が小さい環境では、プロセッサ台数を増加させることにより計算時間を短縮できる。一方、メッセージの送受信時間または同期時間が大きい環境では、プロセッサ台数を増加させるとかえって計算時間は増えてしまう。

また、プロセッサ台数が大きいときは各試行毎の内部計算回数およびメッセージ通信数の差が大きい。これは基準値の選択にランダム値を用いていることに原因があると考えられる。

従って、通信メッセージ数および同期回数を減らしプロセッサ台数の大きいときにも有効なアルゴリズムとすること、基準値の選択のランダム性に起因する計算時間の不安定性を排除することが今後の課題である。

## 4. 結論

本研究では BSP モデル上で高速にソートングを行う並列アルゴリズムを提案した。本研究で提案したアルゴリズムは、1 メッセージ辺りの送受信時間および同期時間が小さい環境では、プロセッサ台数を増加させることにより計算時間を短縮できる。一方、メッセージの送受信時間または同期時間が大きい環境では、その環境に応じて適切な数のプロセッサ台数を選択する必要がある。

## 参考文献

- 1) J.JáJá: An Introduction to Parallel Algorithms, Addison-Wesley Publishing Company (1999).
- 2) L.G.Valiant: A Bridging Model for Parallel Computation, Communications of the ACM, Vol.33, No.8, pp.103-111 (1990).