

1. 序論

本研究では BSP(Bulk Synchronous Parallel) モデル 1) 上で高速にソートを行う並列アルゴリズムを提案する。

BSP モデルとは分散メモリ型の並列計算モデルであり、PRAM(Parallel Random Access Machine)1) と異なり並列アルゴリズムの通信および同期にかかるコストを考慮したモデルである。このため、BSP モデルはより現実の並列計算機に近いモデルとして注目されており、現在様々な問題に対して BSP モデル上で高速に実行できる並列アルゴリズムが求められている。しかし、従来の PRAM のアルゴリズムは通信が考慮されていないため、これを BSP モデル上で実行させても効率よく実行できるとは限らない。従って、通信や同期を考慮した BSP モデル用のアルゴリズムを設計する必要がある。

本研究では基本的な問題であるソートに対して、BSP モデル上で効率良く解く並列アルゴリズムを提案する。

2. 研究内容

本研究で提案するアルゴリズムはクイックソートをベースとしている。クイックソートはデータ内のある値を基準値とし、データをその基準値以下のデータからなる部分データと基準値以下のデータからなる部分データに分割し、各部分データを再帰的に分割していくことによりソートを行う。

分散メモリ型並列計算モデルである BSP モデルでは、あるプロセッサが持つデータを他のプロセッサが使うためには、プロセッサ間で通信および同期を行わねばならない。しかし、一般的に 1 メッセージあたりの通信時間 g および同期時間 L は内部計算時間に比べて極めて大きいとされる。従って高速なアルゴリズムを設計するには通信メッセージ数および同期回数を減らすことが必要となる。

BSP 上でクイックソートを行う場合、各グループでデータを分割後、あるプロセッサから他のプロセッサに対して部分データの送信が行われる。本研究では多くの場合において計算量の大きな部分を占める同期にかかる時間に着目し、同期回数を減らすことにより計算量の改善を図る。

本研究で提案するアルゴリズムは、各グループでデータを分割する際に $k (> 2)$ 個の部分データに分割することにより同期回数を減少させている。図 1 に本研究で提案するアルゴリズムの実行の様子を示す。

3. 結果 考察

表 1: BSP モデル上でのソートの計算量

	計算量	プロセッサ
2 分木	$ng + L \log p$	n
k 分木	$n(\log k + g) + L \frac{\log p}{\log k}$	n

表 1 に、2 分木を用いるソートの計算量および、本研究で提案した k 分木を用いるソートの計算量を示す。本研究で提案したアルゴリズムは、BSP モデル上で

$$O\left(\frac{n}{p} * \log \frac{n}{p} + n(\log k + g) + L \frac{\log p}{\log k}\right)$$

時間でソートを行う。 $(\log k)^2 = L * \log p$ のとき上式は最小となり、効率よくソートを行うことができる。

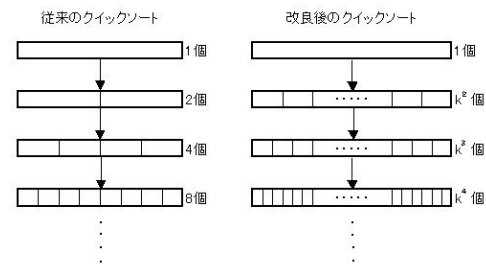


図 1: アルゴリズム実行の様子

4. 結論

本研究では BSP モデル上で効率良くソートを行う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、BSP モデル上で $O\left(\log \frac{n}{p} + n(\log k + g) + L \frac{\log p}{\log k}\right)$ 時間でソートを行う。このアルゴリズムは、同期回数の減少が得られるため、同期時間が長い環境では有用なアルゴリズムとなる。

参考文献

- 1) L.G.Valiant, A Bridging Model for Parallel Computation, Comm. of the ACM, Vol.33, No.8, pp.103-111, 1990.
- 2) 石畑 清, アルゴリズムとデータ構造, 岩波書店 pp.161-184, 1989.