151　BSP モデル上での最短経路問題を解くアルゴリズム

情報論理工学研究室　02-1-47-103　水口 貴裕

1．序 論

並列処理を行うための並列アルゴリズムは、多くの場合 PRAM(Parallel Random Access Machine)1) で設計・解析が行われる。しかし PRAM はアルゴリズムが必ずしも現実の計算機に効率良く適用できるとは限らない。そのため、より現実に近い並列計算モデルとして BSP (Bulk-Synchonous Parallel) モデル 2) が注目されている。　BSP モデルは、ネットワーク接続された複数のプロセッサから成る分散メモリ型並列計算モデルであり、メッセージの送受信やプロセッサ間での同期に掛かる時間を考慮することができる。

本研究では、BSP モデル上で全頂点最短路問題を解く並列アルゴリズムを提案する。　各辺が非負の重みを持つ連結無向グラフ G に対して、ある頂点 u からある頂点 v までの経路 (Path) のうち、経路上の辺の重みの和が最小となる経路を u,v 間の最短路 (Shortest Path) と言う。全頂点最短路問題とは、重み付無向グラフ G が与えられたとき、全ての頂点間で最短路を求める問題である。頂点数 n, 辺数 m のグラフに対し Froyd は $O(n^3)$ 時間で全頂点最短路問題を解くアルゴリズム 4) を提案した。

2．研究内容

本研究では、重み付連結無向グラフ G が与えられたとき、BSP モデル上で G の全頂点最短路を求めるアルゴリズムの提案を行った。また、そのアルゴリズムの BSP モデル上の動作をシミュレートするプログラムを作成し、アルゴリズムの BSP モデル上での時間計算量を実験的に評価した。

全頂点最短路は以下の手順で求めることができる。$n \times n$ 行列 $A = \{a_{i,j}\}, B = \{b_{i,j}\}$ $(0 \leq i, j < n)$ に対して、行列演算 $C = A \circ B$ を以下のように定義する

$$c_{i,j} = \min_{0 \leq k < n} \{a_{i,k} + b_{k,j}\}$$

このときグラフ G の隣接行列を A とすると、G の全頂点最短路は、A^n で求めることができる。ただし、$A^2 = A \circ A, A^{i+1} = A^i \circ A$ である。

行列演算 $\circ$ は行列積と同形の演算である。従って行列積を求めるアルゴリズムを繰り返し用いることにより、全頂点最短路を求めることができる。

本研究で作成したアルゴリズムは、$p \times q$ 台のプロセッサが全頂点最短路の隣接行列 A^n のサイズ $\frac{n}{p} \times \frac{n}{q}$ の部分行列を計算する。

3．結果・考察

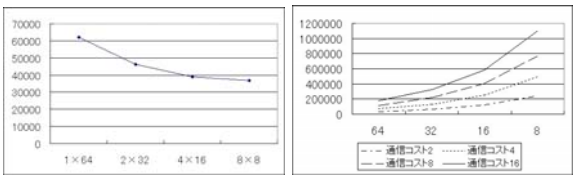

図 1: プロセッサと実行時間

図 1 に、本研究で提案したアルゴリズムの BSP モデル上での時間計算量のシミュレーション結果を示す。左図はグラフの頂点数を 64、1 メッセージ辺りの通信時間を 16、プロセッサ間の同期時間を 16 とし、プロセッサ数を変化させた場合の実行時間のグラフである。プロセッサ数の増加に伴い、計算時間は短縮される。

右図は各プロセッサが計算する部分行列のサイズを $(n \times \frac{n}{16})$, $(\frac{n}{2} \times \frac{n}{8})$, $(\frac{n}{4} \times \frac{n}{4})$ と変化させたときの実行時間のグラフである。グラフより、同じ台数プロセッサを用いても部分行列の縦横の長さにより通信時間が変化することが示される

4．結 論

本研究で提案した BSP モデル上で全頂点最短路問題を解く並列アルゴリズムは、プロセッサ数の増加に伴い実行時間が減少する。また、同じプロセッサ台数でもプロセッサの割り当て方によって実行時間が変化するため、割り当て方を考慮しなければならない。

参考文献

1) J.J 'aJ' a: "An Introduction to Parallel Algorithms," Addison-Wesley Publishing Company (1999).

2) L.G.Valiant: "A Bridging Model for Parallel Computing," Comm. Of the ACM (1990).

3) 渋沢進：並列分散処理入門, 培風館 (1998).

4) 浅野孝夫・今井浩　共著：計算とアルゴリズム, オーム社出版 (2000).