

## 1. 序論

近年ではPCの高性能化やインターネットシステムの発達により、日常のネットワークにおいても、取り扱われている情報の量が増え、複雑な計算処理時間を長く必要とする問題を処理する必要が高まってきた。計算処理高速化の方法の一つに、一つの処理に対して、ネットワーク上で接続した複数の計算機で対応する仮想並列計算機のシステムの採用による解決が考えられる。そこで本研究では、仮想並列計算機環境を構築するソフトウェアであるMPI[Message Passing Interface]1)を用いて仮想並列計算を行い、その実用性を検証する。

## 2. 研究内容

### 2.1 目的

本研究では、MPIを用いた並列計算を行いその有効性を検証する。

### 2.2 研究環境

本研究では、4台の家庭用の計算機をネットワーク接続しMPI環境を作る。また、MPIの有効性を検証するための問題としては最小全域木問題を用いる。最小全域木問題とは、与えられた重み付き無向グラフ(ネットワーク)の部分グラフとなる全域木の中で、辺の重みの和が最小のようになる木を見付ける問題である。本研究では頂点数10,20,30,40の重み付無向グラフが与えられたとき、MPI上でその最小全域木を求める。本研究で作成した最小全域木問題を求めるMPIプログラムはSollin's Algorithm2)を元になっている。

### 2.3 アルゴリズム

本研究では、最小全域木問題を解く並列アルゴリズムとしてSollin's Algorithm2)を用い、それをMPI上でプログラム化した。以下にSollin's Algorithm概要を示す。

1. 各頂点  $v$  において、 $v$  に接続する辺のうち、最も重みの軽い辺  $(v,u)$  を選択し、 $u$  を  $v$  の親とする根付有向森を構成する。
2. 各頂点  $v$  において、 $v$  の根  $r[v]$  を求める。
3. 各頂点  $v$  に接続されている辺と隣接する頂点の情報を  $r[v]$  に送り、各森を1つの頂点としてグラフを構成し直す。上記操作を  $\log n$  回繰り返す。

## 3. 結果・考察

頂点数10,20,30,40のグラフに対して1~4台の計算機を用いてMPI上で最小全域木を求めたときに、その内部計算

表1 内部計算時間と計算機数の関係

台数 \ 頂点数	10	20	30	40
1	0.000010	0.000017	0.000027	0.000038
2	0.000008	0.000012	0.000022	0.000031
3	0.000006	0.000009	0.000015	0.000024
4	0.000005	0.000009	0.000014	0.000021

(m秒)

表2 全体の処理時間と計算機数の関係

台数 \ 頂点数	10	20	30	40
1	0.0045	0.006	0.012	0.023
2	0.012	0.013	0.018	0.039
3	0.016	0.023	0.0055	0.064
4	1.3	2.0	3.0	3.2

(m秒)

にかかった時間を表1に示す。また、通信時間を含む全体の処理時間を表2に示す。表1より計算機の数が多いほど内部計算時間が減っていることが示されている。しかしながら表2より通信時間を含むプログラムの全体処理の時間は、計算機台数増加に伴い増えていることが示されている。計算機数の増加により内部計算時間が減少しているにもかかわらず、全体処理時間が増加していることは、全体の計算処理において通信時間の割合が大きいということが示されていると言える。

## 4. 結論

本研究ではMPIによる並列計算の実用性を求めるためにMPIを用いて最小全域問題を解く時間を検証した。しかしながら内部計算の計算速度を早めることができたのだが、重要な全体処理時間短縮を十分にすることができず、実用性を十分に示せたことにはならない。この事より、実用性のあるMPIによる並列計算をする為には通信時間短縮を考えなければならないことがわかる。その為には通信環境を良くしたり通信速度に負荷の掛からないプログラムを作ることが必要であり今後の課題と考えられる。

### 参考文献

- 1) P パチエコ 著, 秋葉博 訳: MPI並列プログラミング, 培風館(2001)
- 2) J.JáJá 著: An Introduction to Parallel Algorithms, Addison-Wesley Professional (1992)