

1. 序論

近年、CPU の処理性能の増加に伴い計算時間が長くなっている。この問題の解決のために、複数の CPU に仕事を分けて並列処理を行う並列計算機が必要とされている。しかし、一般に並列計算機は非常に高価であるため容易に利用できない。そこで、複数の計算機をネットワーク接続して仮想的並列計算機とする手法が注目されている。本研究では、MPI(Message Passing Interface)²⁾ 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 研究目的

本研究では、MPI を用いた並列計算の有用性を検証する。その検証方法として、逐次処理した場合と MPI を用いて、複数の計算機で並列処理した場合の所用時間を比べ、どの程度処理時間が短縮できるかを計測する。

2.2 実験環境

本研究では、MPI の実装に、MPICH2³⁾ を用いた。これは、MPI 規格を実装したフリーのライブラリ群である。

今回の検証では、同一機種の計算機 5 台を LAN 接続し MPI 環境を構築する。また、MPI の性能検証用の問題として、本研究では最小全域木問題を用いる。最小全域木問題とは、重み付無向グラフ G が与えられたとき、全ての頂点を含む G の部分木のうち、重みの和が最小となるものを求める問題である。本研究では、グラフ頂点数が 5,10,20,40 それぞれの場合で、計算機数 1~5 台を用いた場合の最小全域木を解く時間を計測した。また、本研究で用いた最小全域木を求める MPI プログラムは、Sollin のアルゴリズム¹⁾ を使用している。本研究で作成した MPI プログラムは、PC のうち 1 台をメイン PC として用いる。メイン PC は重み付グラフ G を作成し、 G を頂点数が均等となるようにサブ PC に割り当てる

3. 結果・考察

表 1 に頂点数 5,10,20,40 のそれぞれのグラフに対して 1~5 台の計算機を用いて MPI 上で最小全域木を求めたときのメイン PC のプログラムの実行にかかる時間を内部計算時間として示し、表 2 に全体の処理時間を示す。表 1 よりメイン PC の内部計算時間はサブ PC の台数の増加に応じて処理時間の短縮が得られていることが示されるが、表 2 より全体の処理時間はサブ PC 増えるごとに遅くなっていることが示される。すなわち、計算機台数の増加に伴い内

表 1 メイン PC の内部計算時間と計算機数の関係

台数	頂点 5	頂点 10	頂点 20	頂点 40
1 台	0.000004	0.000009	0.000018	0.000040
2 台	0.000004	0.000008	0.000011	0.000030
3 台	0.000004	0.000006	0.000010	0.000025
4 台	0.000004	0.000006	0.000010	0.000020
5 台	0.000004	0.000006	0.000010	0.000018

(m 秒)

表 2 計算全体の処理時間と計算機数の関係

台数	頂点 5	頂点 10	頂点 20	頂点 40
1 台	0.003	0.004	0.0056	0.02
2 台	0.009	0.011	0.01	0.04
3 台	0.013	0.017	0.025	0.07
4 台	1.4	2.0	2.6	3.2
5 台	1.2	2.0	2.6	2.4

(m 秒)

部計算時間が減少しているにも関わらず全体の処理時間は大きくなることから計算時間増大の原因は通信にかかる時間であると考えられる。

4. 結論

本研究では、MPI による並列化の有用性を検証するために MPI を用いて最小全域木を解く時間の計測を行った。しかし本研究の結果からは MPI の有効性を検証できたとは言えない。本研究の結果では、サブ PC の台数の増加に応じて内部時間は時間が短くなったが、通信に時間がかかるため全体の処理時間は計算機数の増加に反して長くなった。このことより、MPI による並列化の真価を發揮するためには、通信環境を整備し、通信が少ないプログラムを作ることが必要であると考えられる。

参考文献

- 1) J.JáJá 著, An Introduction to Parallel Algorithms, Addison-Wesley Professional (1992).
- 2) P.Pacheco 著, 秋葉博訳, MPI 並列プログラム, 培風館 (2001).
- 3) MPICH2, <http://www.mcs.anl.gov/research/projects/mpich2>, Argonne national laboratory.