146　MPI を用いたグラフの並列計算

情報論理工学研究室　藤本　涼一

1．序 論

情報処理の高速化かつ多様化に従い様々な分野で高速化が追及されており高速化を行うためには複数の 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　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。検証方法として、MPI を用いて並列処理した場合、逐次処理した場合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、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、重み付グラフが与えられたとき、そのグラフの全ての頂点を含み連結かつ閉路を持たない部分グラフのうち、辺の重みの和が最小になるグラフを求める問題である。本研究では、頂点数が 5,10,20,40 それぞれの場合で、計算機数 1 から 5 台を用いた場合の最小全域木を解く時間を計測した。

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 のアルゴリズム 1) を用い、MPI 上でプログラム化した。以下に Sollin のアルゴリズムの概略を示す。

以下の操作を $\log n$ 回繰り返す

step1. 各頂点 v において、v に接続する辺の中から最も重みの軽い辺 (v, u) を選択する。このとき u を v の親として根付有向森を構成する。

step2. 各頂点 v において、v の根 $r[v]$ を求める。

step3. 各頂点 v に接続された辺および隣接する頂点の情報を根 $r[v]$ に送信する。このとき、各森を 1 つの頂点とする。

また、本研究では、MPI 上で上記のアルゴリズムを処理するために PC のうちの 1 台をメイン PC として、メイン PC が各サブ PC にデータを送信し、サブ PC が送られてきたデータを元に計算した結果をメイン PC へ戻す。

表 1　内部計算時間と計算機台数の関係

台数	頂点 5	頂点 10	頂点 20	頂点 40
1 台	0.000022	0.000059	0.000168	0.000440
2 台	0.000016	0.000052	0.000141	0.000385
3 台	0.000010	0.000034	0.000090	0.000255
4 台	0.000010	0.000028	0.000085	0.000200
5 台	0.000010	0.000028	0.000060	0.000148

(m 秒)

表 2　全体の計算時間と計算機台数の関係

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)

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 から 5 台の計算機を用いて MPI 上で最小全域木を求めた場合のメイン PC での通信を除いた内部計算時間の合計を示し、表 2 に各頂点数での通信を含んだプログラム開始から終了までの計算時間を示す。表 1 より、頂点数が増えると計算機数に応じて内部計算時間が減っていることが示される。表 1 および表 2 に示されているように、計算機台数の増加に伴い内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。この事から計算時間増大の原因は通信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を計測した。しかし本研究からは内部計算自体を早くすることはできたが、通信に時間がかかってしまったため、MPI の有用性が十分示せたとは言えない。通信環境を整備し、通信が少ないプログラムを作らなければ MPI の真価を発揮できないと考える。

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).