

1. 序論

大容量の記憶デバイスの登場や、ネットワークの高速化などにより、近年、大量のデータを高速に処理することが求められている。データの高速処理には、複数のプロセッサを持つ並列計算機が必要とされる。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) ¹⁾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

MPI とは Message Passing Interface の略であり、分散メモリ型の並列計算機において、複数のプロセッサ間で、データのやりとりをするために用いる、メッセージ通信操作の仕様標準である。

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

今回使用した環境としては、LINUXOS の PC 一台で、MPI を用い、仮想的な並列計算環境を構築し、検証した。MPI の仕様言語としては、C や Fortran がよく用いられるが、今回は慣れ親しんだ Java を使うため、mpiJava³⁾を用いた。

本研究では、mpi の性能を検証するためにシンプルなか算プログラムを作り、検証した。これは、1 から 10ⁿ (n=2,3,4,5) までの和を求めるプログラムである。また、プロセッサ数は 4 台と仮定している。

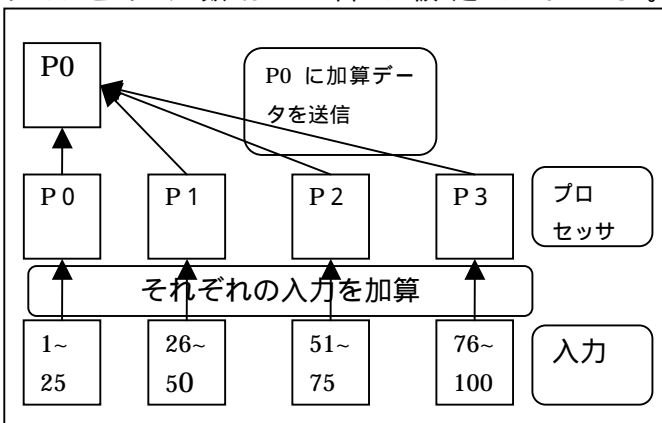


図 1. 並列プログラムの実行概念図

3. 結果・考察

本研究で得られた結果を表 1 に示す。

表 1: mpi による和演算の実行時間(秒)

	1~100	1~1000	1~10000	1~100000
逐次の実行時間	0.019	0.159	1.52	15.191
並列の実際の実行時間	0.878	0.803	2.188	16.289
並列に動作している時間	0.2	0.281	1.321	11.689
並列の実行時間	0.678	0.522	0.867	4.6

並列の実行時間とは、並列プログラムの実際の実行時間から並列動作部分の時間を引いたものである。これにより、仮想的ではあるが、並列プログラムの実行時間を出すことができる。

最大整数が 1000 から 10000 までの間は、実行時間において、逐次プログラムよりも遅かった並列プログラムであるが、10000 を超えてからは速くなっている。また、数値の伸びから、更に数を増やすことにより、差は開いていくと容易に推測できる。

4. 結論

今回のプログラムにおいては、通信回数、時間ともに最小に近いものであった。よって、最大数を増やすことにより、実行時間は、逐次の実行時間の 4 分の 1 に近づいていった。これは、理想的な環境における並列処理の有用性を示している。しかし、実際にプロセッサを繋いで通信した場合、どうしても通信時間を取られてしまう。つまり、通信回数と一台のプロセッサの計算の負担率、これが大きなトレードオフになってくる。つまり、プロセッサの性能、通信速度、これらを考慮した設計をする必要がある。

また、アルゴリズムや MPI を実装する環境作りも、これからの重要な課題である。

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) 渡邊真也 著: MPI による並列プログラミングの基礎, <http://mikilab.doshisha.ac.jp/dia/smpp/cluster2000/PDF/chapter02.pdf>
- 3) 梶原 広輝, 廣安 知之, 三木 光範: mpiJava の利用法範, <http://mikilab.doshisha.ac.jp/dia/research/report/2004/0809/005/report20040809005.html>