研究開発ノート

ネットワークを用いた並列計算システムの応用

井門俊治，長谷部浩樹
(埼玉大学工学部)
(1993年12月8日受理)

Applications of the Parallel Computing System Using Network

IDO Shunji and HASEBE Hiroki
(Received 8 December 1993)

Abstract

Parallel programming is applied to multiple processors connected in Ethernet. Data exchanges between tasks located in each processing element are realized by two ways. One is socket which is standard library on recent UNIX operating systems. Another is a network connecting software, named as Parallel Virtual Machine (PVM) which is a free software developed by ORNL, to use many workstations connected to network as a parallel computer. This paper discusses the availability of parallel computing using network and UNIX workstations and comparison between specialized parallel systems (Transputer and iPSC/860) in a Monte Carlo simulation which generally shows high parallelization ratio.

Keywords:

parallel virtual machine (PVM), parallel programming, socket, network computing, Monte Carlo simulation,

1．はじめに

従来の我々の研究[1]において，MIMD型並列計算機に対して並列化されたプログラムが，インタフェースプログラムを作成することによって，ネットワーク（イーサネット）を介して接続された複数台のワークステーション上で利用可能となることを示した．また，このシステムが，MIMD型並列計算機として実用できることを示した．ここで，MIMDとは，"Multiple Instruction Stream - Multiple Data Stream（複数命令－複数データ流れ）"の略で，独自のCPUとメモリを持った計算ユニットが，独立して計算動作を行い，CPU間の通信（またメッセージ交換とも言う）によるデータ交換によって並列動作を行うシステムを示す．

著者らは，MIMD型の並列計算機であるトランスピュータ（T800）を用いて，並列プログラムの開発を行ってきた[2-8]．並列プログラムは，

Faculty of Engineering, Saitama University, Urawa 338.

ANSI-C，FORTRAN77にチャネル通信を行なうサブルーチンを付加したものであるトランスピュータ用の並列C，並列FORTRANを用いて記述されている．これらの言語を用いて開発されたプログラムは，データ通信のサブルーチンをさしかえることで，ネットワーク接続された複数のワークステーション上や別のMIMD型並列計算機であるiPSC/860でも，直ちに用いることができた[9]．

以前の研究においては，UNIXに装備されたプロセス間通信のためのライブラリであるソケット(socket)を用いたデータ交換により並列計算を行ったが，ここでは，パブリックドメインソフトのParallel Virtual Machine (PVM)[10]を用いて，ネットワーク接続された複数の計算機の集合をMIMD型の並列計算機として用いる方法と，他のシステム（トランスピュータ，インテルiPSC/860）との比較について述べる．このネットワークを用いた並列計算の方式を以下ではネットワークコンピューティングと呼ぶ．ネットワークコンピューティングの利点としては，第1に複数のスーパーコンピュータを用いて，従来1台のスーパーコンピュータで行っていた時よりも高速の演算が実行しうることが挙げられる[11,12]．第2の利点としてはネットワーク内に存在する計算機資源の有効活用がある．即ち，並列処理の動的負荷分散（CPU能力，CPU稼動状況に応じた計算量の割り当て）を行い，使用頻度の小さいCPUの能力を動員することが可能となる．また第3の利点としては従来並列計算機として用いられていなかった計算機をも並列システムにとり込むことが可能となることである．

2．並列アプリケーション

2.1 ネットワークコンピューティングシステムの概要

我々の用いているネットワークにおいて，計算機は図1に示すようにイーサネットと呼ばれる共通の通信媒体で接続されている．各計算機は自分だけアクセスできるローカルメモリをもち，計算機間には共有メモリは存在しない．計算機間のデータ交換は，UNIXのソケットまたは，PVMによって行うことにより，ネットワークコンピューティングシステムをMIMD型の並列計算機とみなすことができる．

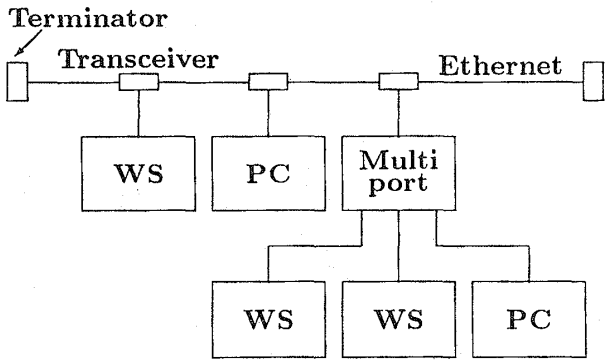

Fig.1 A schematic view of Ethernet. Workstations and PCs are connected to a common line. By using socket librarry or PVM to realize the communications among them, we can use the whole network system as a parallel computer.

ソケットにより，1台のワークステーション内でのタスク間通信はもとより，ネットワークで接続されたワークステーション間のタスク間通信も可能となる．ソケットとは，ユーザプログラムで通信機能を扱うために用意された関数を持つライブラリであり，現在のほとんどのUNIXにおいて，標準のライブラリとなっている．このため，ソケットを用いたプログラムでは，ほとんどのワークステーションで利用できるという利点があるが，低レベルの関数を扱うため，複雑なプログラミングが必要となる場合がある．

PVMとは，アメリカのOak Ridge National Laboratory (ORNL)で開発された並列処理，分散処理を行うことをネットワーク接続された計算機で可能とするソフトウェアである．メッセージ交換のライブラリとデーモンから構成され，多種のワークステーションや並列計算機に対応している．PVMにおける数値データの交換は，データの型を考慮して行われており，数値データの内部表現の違うワークステーション間のデータ交換が，自動的に処理される．我々は，PVMバージョン3.1.2を日立のRISCワークステーション

3050R (PA-RISC 50MHz) に移植して使用した．移植は，ヒューレットパッカードのワークステーション用のソースコードを用いた．作業は単純なもの（リモートシェルの変更）であった．ただし，FORTRAN については，日立の最適化FORTRAN77ではなく，GNU の FORTRAN-C コンバータ (f2c) を利用することにした．他に，サンの Solaris に移植を行ったが，こちらは複雑なものとなった．ただし，PVM のバージョン3.2以降では，Solaris 対応となっているため移植作業は必要ない．

2.2 並列モンテカルロシミュレーション

本研究で用いたテストプログラムは，プラズマスパッタリング装置においてスパッタされた粒子の蒸着のシミュレーションを行うためのモンテカルロコードである．モンテカルロ法を並列化するときに問題となるのは乱数系列の並列化である．本研究では乱数の種（たね）を100個用意し，各計算タスクがシミュレーションで用いる乱数の種を制御タスクが指定するようにして並列化を行った．また，それぞれの粒子の運動は，それぞれ独立に扱うことができるため，手法としては粒子の運動（事象）の並列化を用いた．

われわれがこれまで用いてきた並列計算機（トランスピュータ，iPSC/860）は，同一の能力を持ったプロセッサが複数集まったものであった．このため，各プロセッサに対する負荷の配分は均等にしておくことで並列化の効果は一般的に高くなっていた．しかし，数種の計算機の存在するネットワークコンピューティングシステムにおいて効率のよい並列処理を行うためには，各計算機の計算処理能力に応じた計算負荷の割り当てを行う必要がある．また，各計算機の負荷状態は時間ごとに異なっているため，この負荷の割り当てはリアルタイムに行う必要がある．

今回のテストプログラム（モンテカルロコード）においては，この動的負荷分散を実現する方法として，粒子を100の組に分け，1組ずつ各計算タスクに割り当て，その組の計算が終了したタスクに対し，次の組を割り当てるようにした．これにより計算が速いタスクに対しては多くの粒子の計算が割り当てられることになり，各計算機の計算時間の均一化を行うことができる．

並列化プログラムは，複数の計算プログラムとそれを制御する制御プログラムから構成される．各計算プログラム間でデータ交換の必要がないため，プログラム間の通信は，各計算プログラムと制御プログラム間のみである．トランスピュータの接続は，パイプライン接続を採用していたため，通信は隣接したトランスピュータ間でしか実現できないので，データを制御プログラムまで中継する必要があった．ネットワークコンピューティングシステムでは，全てのプロセッサは，共通の媒体で接続されているため，データの中継の必要はない．iPSC/860では，ハード的に直接接続されていないプロセッサの間でも，ソフトウェアにより，データ転送がなされるので，並列化プログラムでは，データの中継を考慮する必要はない．よって，トランスピュータ用のプログラムに，データ中継を行うプログラムを加えることにより，同一のプログラムを MIMD 型並列計算機で利用することにした．

3．結果及び考察

3.1 通信性能比較

今回用いたトランスピュータ，イーサネットでは，通信速度はどちらも10Mbps（メガビット毎秒）となっている．実際の通信性能を得るために2台の計算機間でデータ交換を行い，実行にかかった時間（通信時間）から通信性能を比較した．

図2に通信データバイト数に対する通信時間を示す．図中の実線は，式(1)によって描かれる．

$$T = T_0 + \frac{s}{v} \tag{1}$$

ただし，T [秒] は通信時間，T_0 [秒] はセットアップ時間，s [バイト] は送受信するデータのサイズ，v [バイト毎秒] は，通信速度である．式 (1) は，通信がセットアップと指定したデータ量だけの通信を行っているという仮定のもとで成立する．セットアップ時間とは，通信関数を実行している時間のうちで実際に通信を行っていない時間，つまり通信を行なうまでにかかる時間と通信

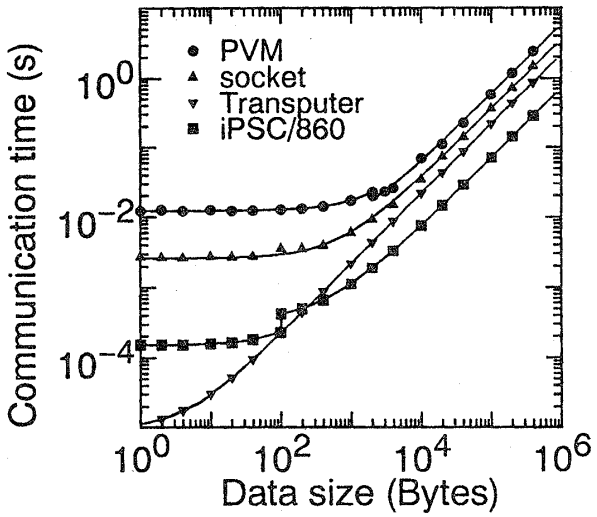

Fig. 2 The time taken to exchange data between two processing elements, in the case of network systems they mean workstations. The solid lines are the theoretical curves calculated by Eq.(1). At the small data size, network systems (socket and PVM) are slower than other systems. For the large data size, network systems show the similar performance to other systems.

が終了して次の動作に移るまでにかかる時間を示す．

少ないデータを送る場合（例えば1000バイト以下の時），ソケットまたは，PVMを用いたネットワークにおけるデータ通信に要する時間は，トランスピュータやiPSC/860が通信に要する時間よりも大きい．UNIXで取り入れられているTCP/IP (Transmission Control Protocol/Inetnet Protocol)においては，イーサネットを介しての通信を行うまでに，送信データにさまざまな情報が付加され，より大きなデータ量となってネット上を転送されている．こうした情報の付加は，ソフトウェアレベルで実現されているため，セットアップ時間（T_0）は，他のシステムと比較してはるかに大きくなっている．PVMにおいては，デーモンとの通信などもあるため，セットアップ時間は，さらに大きくなっている．また，測定においては，ネットワークの使用状況で通信性能が大きく変化するため，測定を行うマシンを全体のネットワークから切り放し，そのマシンだけでネットワークを構成した．よって実際のネットワークで測定した場合には，この結果よりも通信性能は低くなる．

3.2 速度向上特性

トランスピュータ用に開発された蒸着シミュレーションのMIMD型並列化コードを用いてトランスピュータ，ソケットを用いて複数のSPARC station-1を接続したネットワークコンピューティングシステム，日立3050Rで構成されたPVMによるシステム，iPSC/860の4種類のシステムで並列計算を行った．そのときの速度向上特性を図3に示す．図3の縦軸の速度向上比は，SPARCstation-1を1台使用した時の計算時間を1として，計算時間の速度向上を示している．図中の実線は，アムダールの式[5]によって描かれた速度向上特性であり，SPARC，トランスピュータ，iPSC/860の場合において，並列化率99.5%，3050Rにおいて，並列化率97.5%である．アムダールの式は，次式(2)で表される．

$$t_n = t_1\left(1-\alpha+\frac{\alpha}{n}\right) \tag{2}$$

ただし，t_n [秒] は，n台のプロセッサを使って並

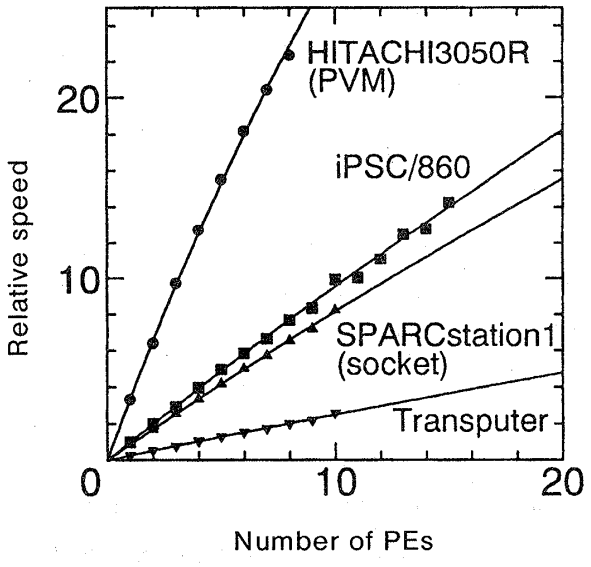

Fig. 3 The acceleration performances by parallel computing. The calculation time is normalized to the time required in a calculation by using one SPARCstation-1.

列計算したときの計算時間，t_1 [秒] は，1台のプロセッサでの計算時間，α は，並列化率をそれぞれ表している．式 (2) は，1台のプロセッサのみを用いて行った処理のうち，α だけが並列化可能な部分であり，n 台のプロセッサを用いることにより，その部分の処理時間が，$1/n$ に短縮され，残りの $(1-\alpha)$ が並列化不可能な部分で，n 台のプロセッサを用いても処理時間の短縮が不可能であることを示している．テストプログラムとしたモンテカルロコードの並列化率は，ほぼ100%であるが，3050Rにおいては，計算能力が高い分だけ，計算時間が相対的に少なくなり，通信によるオーバーヘッドの割合が増したため，式 (2) で高速化性能を近似した時の並列化率 α が減少した．

図3からわかるように，ネットワークコンピューティングにおいても，直線的な速度向上を得ていることから，ネットワークコンピューティングによる並列計算は，有効な手段であることがいえる．

4．まとめ

ネットワーク接続された複数のワークステーションをMIMD型並列計算機として用いる方法として，ソケットとPVMを用いる例を示した．通信関数名の置き換えを行うライブラリを作成することにより，テストプログラムとしたシミュレーションコードをトランスピュータ，iPSC/860，ソケット，PVMで実行し，各システムにおける速度向上性能を比較した．その結果，ネットワークコンピューティングが，MIMD型並列計算機として，実用できることを示した．ただし，通信性能の比較では，並列計算機として設計されたシステムより劣っているため，通信のオーバーヘッドが生ずる場合がある．

参考文献

[1] 井門俊治，長谷部浩樹：プラズマ核融合学会誌**69**, 343 (1993).
[2] 井門俊治，辻　龍介：日本原子力学会誌**30**, 1103 (1988).
[3] 井門俊治，辻　龍介：日本原子力学会誌**30**, 999 (1988).
[4] 井門俊治，辻　龍介，中井泰明：日本原子力学会誌**31**, 273 (1989).
[5] 井門俊治，久保田昌晴：核融合研究**64**, 506 (1990).
[6] 井門俊治，青木一夫：核融合研究**65**, 71 (1991).
[7] 井門俊治，石根正久：核融合研究**66**, 69 (1991).
[8] S. Ido, K. Aoki, M. Ishine and M. Kubota, Computational Mechanics **9**, 305 (1992).
[9] S. Ido and S. Hikosaka, Computational Mechanics **10**, 151 (1992).
[10] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM3 USER'S GUIDE AND REFERENCE MANUAL*, Oak Ridge National Laboratory, (1993).
[11] 吉岡　顕，矢川元基，吉村　忍，曽根田直樹：日本機械学会論文集**57**, 1964 (1991).
[12] 吉岡　顕：プラズマ核融合学会誌**69**, 1002 (1993).