並列処理教育用 PC クラスタの構築

山崎勝弘　三浦誉大
立命館大学理工学部情報学科

1. はじめに

PC16 台からなる PC クラスタを並列処理教育用に構築し，分散メモリ並列プログラミングモデル(PVM)と共有メモリ並列プログラミング言語(OpenMP)による並列プログラミング環境を実現した．本稿では，PC クラスタ構築の経緯と並列プログラミングの現状を述べる．

2. PC クラスタ

PC クラスタの構成を図 1 に示す．

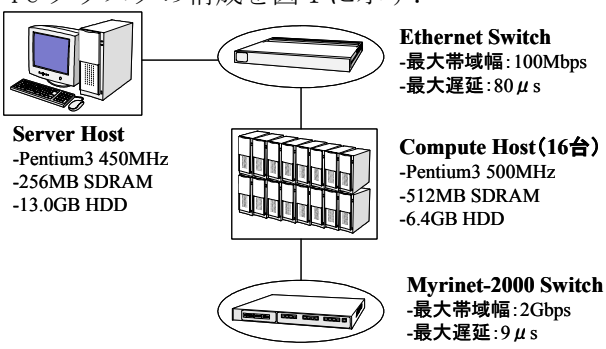

図 1　PC クラスタの構成

本研究室の PC クラスタは，計算ノードとして PC16 台と，それらを管理するサーバを 1 台設置し，研究室内の 100BASE-T Ethernet に接続してある．また，Myrinet2000 を用いて PC16 台をつないでいる．Myrinet2000 では 4Gbps(片方向 2Gbps，双方向通信)の高速通信が可能である．

3. システムソフトウェアのインストール

各 PC にまず Linux をインストールし，その上に RWC が開発したクラスタシステムソフトウェア SCore[1]をインストールした．途中 SCore の機能拡張などもあり，以下の 3 段階を経た．

(1) Kondara1.1 + SCore3.3.2

Kondara を選択したのは，通常使われそうなサービスが on になっているので，初めにやるには使いやすいからである．

まず，PC4 台を Ethernet と Myrinet で接続し，動作を確認した後，8 台で組んで動作を確認した．その後，Omni OpenMP コンパイラをインストールした．

(2) Turbo6.1 + SCore3.3.2

Kondara では，ソースコードからインストールしたので，手間がかかり煩雑であった．そこで，SCore で用意されているイージーインストレーションツール(EIT)を利用した．これにより，インストールによる手間を大幅に削れるが，OS やカスタマイズし難いなどの制約を受ける．

Turbo に変更したのは，EIT が Kondara を対象外としていたからである．Ethernet と Myrinet2000 を使い，8 台で動作確認を行った．

(3) RedHat7.1 + SCore4.2

これまでは，サーバと計算ノードを兼用していたので，計算時間にばらつきがあり，不安定であった．そこで，サーバと計算ノードをそれぞれ専用にすることにより安定を図った．

Turbo はインストールもほぼ自動でできるが，細かい設定は自分で行わなければならず，設定の書式が RedHat 系の Linux の中でも異なっていてやりにくかった．そこで，最も普及しており，ドキュメントが入手しやすい RedHat に変更した．現時点では RedHat7.2 に SCore5.2.0 に更新されされている．

4. インストールとテストの手順

4.1 Linux のサーバへのインストール

計算ノードへのインストールは EIT を使うのであれば必要なく，サーバだけに Linux をインストールすればよい．

4.2 Linux カーネルをあてる

Linux カーネルは CPU のコントロール，タスク管理などの基本的な部分を受け持っている．動作を安定させるために最新バージョンをあてておく．

4.3 SCore のインストール

EIT をサーバで実行し，細かな設定を EIT の指示に従って行う．計算ノードへのインストールはこのときに Ethernet 経由で行う．EIT はサーバで必要なファイルの設定と，計算ノードに必要なファイル(OS，SCore 関連)のコピーを行ってくれる．

Development of a PC Cluster for Parallel Processing Education
Katsuhiro Yamazaki and Takahiro Miura
Ritsumeikan University

4.4 システムテスト

クラスタの正常動作を確認するために，以下の3つのシステムテストを行った．

(1) SCOUTテスト

SCore用シェルの動作の設定と確認をする．計算ノードの状態を監視するCompute Host Lock Clientが正常に動くか確認した．

(2) PMテスト

サーバと計算ノード間のPMⅡ通信ライブラリが正常に動作するか確認した．Ethernet，Myrinet，UDP，Shmem(共有メモリ)それぞれについてテストした．Myrinetのゼロコピー(メモリを介在しないコピー)機能もここでテストした．

(3) SCoreテスト

最後に，サンプルでついているマンデルブローのプログラムを走らせて動作確認した．また，PVMとOpenMPで簡単なテストプログラムを作成し，コンパイル・実行して動作確認を行った．PVMはEthernetのみを使用し，OpenMPでは制御情報はEthernet経由で送られ，計算はMyrinetを介して行われる．

5. 並列プログラミングの現状

PCクラスタの動作確認，OpenMP[2]の学習，及び並列プログラミングのノウハウの蓄積を目的として，比較的小規模な並列プログラムの作成を行っている．今まで作成したのは，積分計算（サイクロイド），マンデルブロー，文字列照合(KMP法，BM法)，ランレングス圧縮，ビジネル暗号，ハフ変換，細線化などである．同じ問題でも直ちに並列効果が得られない場合，データサイズやアルゴリズムを工夫して，どのような場合に効果が得られるかを調べている．

以下，Hilditchの細線化アルゴリズム[3]を並列化した実験の要点のみ述べる．この問題では単純なブロック分割では境界が正しく処理されないので，ある線幅だけ境界部分を重複させる必要がある．並列アルゴリズムはプロセッサファーム(マスタースレーブ)，データ分割はブロック／サイクリック，スケジューリングとして静的／動的を行った．

(実験A) 1000x5000ピクセルの画像中に線幅70の文字がある．縦方向のみサイクリック分割で行うと，PC16台で不均質画像では6.8倍，均質画像では7.6倍の速度向上が得られた．

(実験B) 3000x3000ピクセルの画像中に線幅30の文字がある．縦横それぞれ15個ずつ計225個のユニットに分割し，動的・静的両方のスケジューリングで実験した。不均質画像では動的スケジューリングが最善で，16台で6.8倍，均質画像では静的サイクリック分割が最善で，16台で14.8倍の速度向上が得られた．

6. クラスタ使用上の留意点

今までのテストの結果，クラスタ使用上の制約や留意点として，以下のことが分かっている．

(1) ユーザが使用可能なメモリ空間

SCoreでは分散共有メモリをソフトウェア(SCASH)で実現している．このため，ユーザが使用可能なアドレス空間には制限がある．細線化プログラムで試した結果，メモリ容量が256MB/ノードの場合90MB程度，512MB/ノードで170MB程度のデータまで扱えることが分かっている．

(2) 共有引数スタックオーバーフロー (shared arg stack overflow)

引数メモリサイズは，デフォルトでは4KBになっている．関数内で宣言する変数のサイズの合計がこれを超える場合には，環境変数(OMNI_SCASH_ARGS_SIZE)で引数メモリサイズを指定しなければならない．

(3) Myrinetの通信性能

現在，計算ノードのPCIバスが32bit/33MHz(1056Mbps)のため，Myrinetの性能が十分出ていない．このため，細線化プログラムを不均質画像を対象に，サイクリック分割で実行した場合，sendとreceiveの警告が出続けて，プログラムが終わらない場合がある．これはMyrinetのメッセージの取りこぼしが発生しているためであると考えられる．この問題を回避し，十分な通信性能を出すためには，64bit/66MHzのPCIバスで接続することが必要である．

7. まとめ

本稿では我々の研究室におけるPCクラスタ構築の経緯と並列プログラミングの現状を述べた．PCクラスタを比較的安価に構築でき，並列処理の教育・研究ができることは喜ばしいことであるが，使いこなすにはかなりの努力が必要である．現在，JPEG，流体問題などの並列化に取り組んでいる．

参考文献

[1]石川裕，他："Linuxで並列処理をしよう"，共立出版，2002.

[2]R. Chandra, et al. : "Parallel Programming in OpenMP", MORGAN KAUFMANN PUBLISHERS, 2001.

[3]安居院，長尾：C言語による画像処理入門，昭晃堂，2000.