(付録 C)

実験環境(CSE クラスタ)

１．システムの運用

実験に用いるクラスタには CSE (Computer System Experiment)という名前が付けられている。これは 4 ノードの SMP-PC を Fast Ethernet で結合したシステムであり、各 SMP-PC ノードでは、さらに 2 つのプロセッサが SMP 結合されている。従って、システムは 2 プロセッサ×4 ノードで合計 8 プロセッサからなる。

このクラスタは HPCS (High Performance Computing System)研究室で管理されており、実体もそこに置いてある。従って、普段の実験は情報学類の教育用計算機システム室で行い、ここからクラスタにログインして用いる。ここで注意すべき点は、クラスタ内の 4 つのノードのうち、先頭のもの以外には直接 coins のマシンからのログインはできなくなっている点である。また、後述する msgb コマンド等では X-window を用いて出力が行われるため、DISPLAY を間違いなくセットするために、rlogin ではなく ssh によるログインを行うこと。

ログインするホスト名は cse.coins.tsukuba.ac.jp である。coins ドメインのマシンからは単に cse と指定するだけでもよい。

```
% ssh　cse
```

初めて ssh でログインした場合、いくつかのメッセージが出るが、これには yes で答えればよい（相手のホストを信頼するかどうか等が聞かれる)。cse は、クラスタの代表ノードで、クラスタ内部では cse0 という別名で参照できる。もし必要であれば、cse に ssh でログインした後、そこから他のノード（cse1, cse2, cse3）に対してログインすることが可能である。

テキスト本文中で説明したように、本実験ではプログラムの作成及びデバッグを行うフェーズと、性能評価を行うためにシステム全体を排他的に用いるフェーズを使い分ける。便宜上、前者を**プログラミングフェーズ**、後者を**性能評価フェーズ**と呼ぶことにする。両フェーズではプログラムのコンパイルと実行のやり方が異なるので十分注意すること。また、システムの機能上の制約から、**同じノード上で性能評価フェーズでプログラムを走らせている場合でも、プログラミングフェーズのプログラム実行が同時にできてしまう。**従って、性能評価フェーズは特定の日時に集中して行うこととし、それまでにプログラムのデバッグを終了して性能評価を待つような形でシステムを運用することとする。

２．両フェーズの環境

[プログラミングフェーズ]

プログラミングフェーズでは、MPI プログラムのコンパイル及び実行のため、コマンド

パスに /home/local/mpich/bin を通しておく。この下の mpicc 及び mpirun によって、プログラミングフェーズのジョブが実行できる。OpenMP に関しては、/opt/omni/bin をコマンドパスに通しておくことにより、omcc コマンドでコンパイルが可能となる。

[性能評価フェーズ]

性能評価フェーズでは、SCore 環境を用いた他のユーザとの排他制御機能(シングルユーザモード)を用いることにより、システムを独占し、正確な性能評価を行う。

まず、性能評価フェーズ全体において、必ず /opt/score/bin をパスの先頭に入れておく。このディレクトリの下にある mpicc は、プログラミングフェーズとは別の環境で MPI プログラムをコンパイルする。先述のプログラミングフェーズでコンパイルした MPI プログラムは、性能評価フェーズで実行することはできない。従って、両フェーズを使い分ける際に、コマンドパスを修正し、コンパイルを再実行する必要がある。

OpenMP に関してはパスを変更する必要はない。

性能評価環境においてプログラムを実行するには以下の手順で行う。

(1) msgb プログラムを実行する。これは、SCore 環境で動作する複数のノードの状況をモニタするプログラムである。SCore 環境では、システムの排他制御をコントロールするために、**並列実行グループ**という概念で並列プロセスをグルーピングする。msgb でもこのグループを指定する必要がある。

```
% msgb -group parallel
```

ここで、parallel というのはこの実験で用意したグループ名である。これ以外のグループはないため、必ずこの名前を指定すること。

(2) scout プログラムを実行する。scout は SCore 環境下でプロセスを制御するための一種のシェルである。

```
% scout -g parallel
```

msgb の場合と同様、-g オプションの後には並列処理グループ名、すなわち parallel を指定する。scout を実行すると、何らかのメッセージが表示された後、そのまま普通のコマンド入力状態になる。これは、scout がシェルとして機能しているためで、この上で通常の UNIX コマンドが実行できる。

もし他のユーザが scout で parallel グループを使っている場合、他のユーザの scout は失敗する。これにより、排他制御が実現される。

(3) scrun コマンドにより並列プログラムを実行する。使い方はプログラミングフェーズにおける mpirun とほぼ同じである。

```
% scrun -np プロセス数 プログラム名 [プログラムの引数…]
```

(4) システムの返却を行う。scrun により、一連のプログラムを実行し終えたら、速やかに scout 環境から抜ける。これは、exit コマンドを、普通の UNIX のように実行すればよい。scout 環境からいつまでも無駄に抜けずにいると、他のユーザが同じグループの下

で性能評価フェーズを実行することができなくなるので注意すること。