



## MPI 環境の使い方 (2011/1) (クラスターシステム対応)

ビジュアルテクノロジー株式会社

### はじめに

MPI 環境は、Open MPI、MPICH1、MPICH2、MVAPICH2 がインストールされています。

いずれもジョブスケジューラ (torque) 経由、OS 直コマンドラインで実行可能です。  
通常の GNU コンパイラでビルドされた環境のほかに、Intel や PGI コンパイラご購入の場合は、それぞれのコンパイラでビルドされた MPI ライブラリもインストールされています。

コマンド「mpi-selector-menu」により、適宜切り替えて使用することが出来ます。

以下、実行コマンドなどは、特記なき場合は一般ユーザでの実行例として記します。

(root のパスワードは納品資料 (別ドキュメント) に記載しています)  
新規ユーザ作成の際は後記をご参照の上、ユーザを作成してください。

## MPI-selector の使い方

### 1 . 登録されている MPI ライブラリの確認

以下のコマンドを入力すると、登録されている MPI ライブラリが表示されます。  
(以下は例、お客様の環境、納入時期により表示が異なります)

```
$ mpi-selector-menu
```

```
Current system default: openmpi-1.4.1-pgi64-v10.2
```

```
Current user default: <none>
```

```
"u" and "s" modifiers can be added to numeric and "U"  
commands to specify "user" or "system-wide".
```

```
1. mpich-1.2.6-pgi64-v6.2  
2. mpich2-1.0.8-gnu64-4.1.2  
3. openmpi-1.4.1-pgi64-v10.2  
4. mvapich-1.1-pgi64-v10.2  
U. Unset default  
Q. Quit
```

```
Selection (1-16[us], U[us], Q):
```

1,2 行目から現在のシステムデフォルトとユーザのデフォルトがわかります。

### 2 . 選択

Selection の入力行で選びたい MPI の数字 (例えば 4) とユーザを表す 'u' を続けて

```
4u
```

と入力すると、

```
Current system default: openmpi-1.4.1-pgi64-v10.2
```

```
Current user default: mvapich-1.1-pgi64-v10.2
```

```
:
```

と表示され、ユーザデフォルトが正しく設定されたことが確認できます。

最後に 'q' と、入力して抜けます。

### 3 . 設定の反映

```
$ bash
```

または、ログインをやり直します。

X windows で使用の場合は exit で端末を閉じ、再度端末を起動し

```
which mpirun
```

とコマンドを入力し、使いたい MPI ライブラリのパスが表示されるか確認します。

MPI-selector を使用しない場合は、各ユーザーの「.bashrc」に MPI ライブラリの PATH, LD\_LIBRARY\_PATH を 適切に設定します。

## Open MPI での使用について

### [ コンパイル例 ]

```
$ mpicc -O3 -o cpi cpi.c      (コンパイラによって最適化などのオプションが異なります.)  
インテルコンパイラの例 (Xeon 5500 番台など)  
$ mpif77 -O3 -xSSE4.2 -o myapp myapp.f
```

=====  
・ジョブスケジューラ (torque) 経由での実行方法  
=====

### [ ジョブスクリプトの例 ]

(右括弧内は説明 (スクリプト中には記述しない))

```
#!/bin/sh  
#PBS -N test                (JOB 名, 任意の名称)  
#PBS -j oe                 (標準 / 標準エラーを出力)  
#PBS -l nodes=4:ppn=8     (ノード数, ノードあたり使用コア数) (必須)  
#PBS -q default           (デフォルトの JOB クラス)  
cd $PBS_O_WORKDIR        (実行するディレクトリに移動) (必須)  
mpirun ./cpi              (*)
```

(\*)torque 経由 Open MPI ジョブでは, -machinefile \$PBS\_NODEFILE  
は不要です。(Open MPI ビルド時に torque の位置を指定しています.)

(\*) mpirun -n 8 ./a.out のように並列数を指定してもよいが #PBS -l nodes=4:ppn=8 で指  
定しているので -n 8 は記載しなくても良い.

### [ ジョブの実行 (投入) ]

```
$ qsub run.sh
```

qsub run.sh にて投入したときに,  
###.HOST-NAME.DOMAIN-NAME (###が JOB 番号)  
で表示される JOB 番号###ごとに,  
JOB 名.o### (標準出力)  
JOB 名.e### (標準エラー出力)  
のファイルが, 実行後, 生成されます.  
但し, リダイレクションで標準 (エラー) 出力を指定した場合は,  
そのファイルに出力されます.

### [ ジョブの状況確認 ]

```
$ qstat -a  
(S(Status)の R が実行中, Q が実行待ち)
```

### [ ジョブのキャンセル ]

```
$ qdel ### (###は JOB 番号)
```

### [ 実行中のジョブの kill ]

```
$ qdel ### (###は JOB 番号)
```

### [ ジョブの実行結果の確認 ]

```
$ cat test.o### (###は JOB 番号)
```

=====  
・直コマンドでの実行方法 (torque 経由しない場合)  
=====

[ 実行例 ]

```
$ mpirun -hostfile host32 -n 32 ./cpi
```

hostfile の例 上記では host32 (各ノードの hostname を使いたいコア数羅列します.)

```
node1.localdomain
```

```
node1.localdomain
```

```
...
```

```
...
```

```
node4.localdomain
```

### MPICH1.2.系での使用について

[ コンパイル例 ]

```
$ mpicc -O3 -o cpi cpi.c
```

=====  
・ジョブスケジューラ (torque) 経由での実行方法  
=====

[ ジョブスクリプトの例 ]

(各項目の説明は Open MPI の場合と同様です)

```
#!/bin/sh
```

```
#PBS -N test
```

```
#PBS -j oe
```

```
#PBS -l nodes=4:ppn=8 (必須)
```

```
#PBS -q default
```

```
cd $PBS_O_WORKDIR (実行するディレクトリに移動) (必須)
```

```
mpirun -np 32 -machinefile $PBS_NODEFILE ./cpi .....(*)
```

(\*)torque 経由 MPICH1 ジョブでは, -machinefile \$PBS\_NODEFILE  
で torque から使用するノードを取得します.

ジョブの実行 (投入), 状況確認, キャンセル, kill, 実行結果の確認は  
Open MPI の場合と同様です.

=====  
・直コマンドでの実行方法 (torque 経由しない場合)  
=====

[ 実行例 ]

```
$ mpirun -machinefile host32 -np 32 ./cpi
```

machinefile (上記では host32) は OpenMPI と同じか, 以下のように, hostname の後ろに  
コア数を指定したもの.

```
node01.localdomain:8
```

```
node02.localdomain:8
```

## MPICH2 1.0,1.1,1.2 系での使用について

[ コンパイル例 ]

```
$ mpicc -o cpi cpi.c
```

=====  
・ ジョブスケジューラ ( torque ) 経由での実行方法  
=====

MPICH2 で , torque 経由での実行の場合は OSC MPIEXEC を使用します .

<http://www.osc.edu/~pw/mpiexec/index.php>

[ ジョブスクリプトの例 ]

( 各項目の説明は Open MPI の場合と同様です )

```
#!/bin/sh
#PBS -N test
#PBS -j oe
#PBS -l nodes=4:ppn=8          ( 必須 )
#PBS -q default
cd $PBS_O_WORKDIR             ( 必須 )
MPIEXEC="/usr/local/mpiexec-0.83/bin/mpiexec"
$MPIEXEC -comm=mpich2-pm ./a.out.          (*)
```

(\*)torque 経由 MPICH2 ジョブでは , -machinefile \$PBS\_NODEFILE  
は不要です . ( mpiexec ビルド時に torque の位置を指定しています . )  
OSC MPIEXEC では mpdboot は使用しません .  
mpiexec はフルパスで指定して下さい .

ジョブの実行 ( 投入 ) , 状況確認 , キャンセル , kill , 実行結果の確認は  
Open MPI の場合と同様です .

=====  
・ 直コマンドでの実行方法 ( torque 経由しない場合 )  
=====

・ \$HOME/.mpd.conf ファイル作成 , secretword 設定 ( 任意の文字列 )  
permission 設定 ( chmod 600 )

```
$ cat $HOME/.mpd.conf
```

( 例 )

```
secretword=12345
```

```
$ chmod 600 $HOME/.mpd.conf
```

・ \$HOME/mpd.hosts ファイル作成 , ( hostname を羅列したファイルを作成します )

```
$ cat $HOME/mpd.conf
```

( 例 )

```
node01.localdomain
```

```
node02.localdomain
```

・ 起動

```
$mpdboot -n 4 -r rsh -f mpd.hosts &
```

( 4 ノードを通信は rsh で行う 実行する host は mpd.hosts の内容 )

・ 実行 ( machinefile の内容は OpenMPI 使用時と同じ )

```
$mpirun -machinefile host32 -np 32 ./a.out
```

・ 終了

```
$ mpdallexit
```

## MVAPICH2.系での使用について

Infiniband 環境では OpenMPI より高速な場合もあります . Infiniband 向け MPICH2 環境です .

[ コンパイル例 ]

```
$ mpicc -O3 -o cpi cpi.c
```

```
=====
```

・ジョブスケジューラ ( torque ) 経由での実行方法

```
=====
```

[ ジョブスクリプトの例 ]

( 各項目の説明は Open MPI の場合と同様です )

```
#!/bin/sh
```

```
#PBS -N test
```

```
#PBS -j oe
```

```
#PBS -l nodes=4:ppn=8
```

( 必須 )

```
#PBS -q default
```

```
cd $PBS_O_WORKDIR ( 実行するディレクトリに移動 ) ( 必須 )
```

```
mpirun_rsh -rsh -np 32 -hostfile $PBS_NODEFILE ./cpi .....(*)
```

(\*)torque 経由 MVPICH ジョブでは , -hostfile \$PBS\_NODEFILE  
で torque から使用するノードを取得します .

ジョブの実行 ( 投入 ) , 状況確認 , キャンセル , kill , 実行結果の確認は  
Open MPI の場合と同様です .

```
=====
```

・直コマンドでの実行方法 ( torque 経由しない場合 )

```
=====
```

[ 実行例 ]

```
$ mpirun_rsh -rsh -hostfile host32 -np 32 ./cpi
```

hostfile ( 上記では host32 ) は OpenMPI と同じです .

MVAPICH2 では MPICH2 と同様の mpdboot を起動しての実行も可能です .

## mpi-selecto-menu での MPI ライブラリの表記

MPI 種類-MPI バージョン-コンパイラ種類-コンパイラバージョンで表記されています。  
openmpi-1.4.1-intel64-v11.1.046  
は以下を意味します。

MPI : OpenMPI  
MPI バージョン : 1.4.1  
コンパイラ : Intel 64bit コンパイラ  
コンパイラバージョン : 11.1.046

## ユーザ ID の追加などについて

別紙 クラスタ管理ガイド 6.2 節を参照してください。

**RedHat5 系 , CentOS5 系での OS に関するドキュメントの表示**  
OS 画面上部のメニューバーの  
System ( システム )

Document ( ドキュメント )

Deployment Guide(ja-JP)をクリックしますと  
システム全般の日本語マニュアルが出ます。