

CPU服务器并行运算指南

本小组有两个单机的CPU服务器：

222.195.74.10 服务器为处理器为**2×AMD Opteron 6272**十六核处理器**2.1GHz**，共32核；内存：**128G DDR3 1333**内存、支持高级内存容错功能的服务器。上面挂载有**12块2T**硬盘（共**24T**）的磁盘阵列。

222.195.76.220 服务器为处理器为**2×AMD Opteron 6376**十六核处理器**2.3GHz**，共32核；内存：**128G DDR3 1333**内存、支持高级内存容错功能的服务器。上面挂载有**4块2T**硬盘（共**8T**）的磁盘阵列。

只能通过校内**IP**登陆到**CPU**服务器,登陆方式为：

`ssh -X UserName@222.195.74.10` (启用X11图形模型)

`ssh UserName@ 222.195.74.10` (不启用X11图形模型)

或者：

`ssh -X UserName@222.195.76.220` (启用X11图形模型)

`ssh UserName@ 222.195.76.220` (不启用X11图形模型)

希望用户在使用时请特别注意：

废进程可能产生堆积，有时严重占用内存，降低**Cluster**的使用效率。用户应经常查找自己僵死掉的进程(**ps -aux**),并**kill**掉！

机器的故障是无法避免的，重要数据、中间结果.....用户自己要随时在本机以外的设备上备份，自动保护措施也可能因机器的故障而失去作用！

另外：

- 1、用户工作 **Home** 在 **/public** 下。
- 2、高性能计算软件 在 **/public/software** 下。
- 3、用户登录后使用,请注意环境变量是否符合自己的需求。
- 4、环境变量的**.SSH**脚本在 **/public/software/ profile.d** 中。
- 5、**/public** 组成磁盘阵列存储空间。
- 5、有问题请联系：**63601095** lsc@ustc.edu.cn

下面也许是用户要注意的：

MPICH1 已於2006年停止更新，对于现代多核CPU（例如：至强12核CPU），已不能适应。

请使用**MPICH2**，**MPICH2**支持最新的**MPI-2**接口标准，与**Mpich1**相比**MPICH2**具备更加严谨和合理的结构，可移植性和效率更好，支持**C/C++**和**Fortran**。

MPICH2的主页是：

<http://www-unix.mcs.anl.gov/mpi/mpich2/index.htm>

在 <ftp://ftp.mcs.anl.gov/pub/mpi> 下有**MPICH2**的

安装和使用指南，分别是**User's Guide, Installer's Guide**

以下是为计算机阵列而写（单机运行时，也可以使用**CPU**中的众多核心,做并行计算）可参照处理。

MPICH2

特点

支持**mpi2.0**,功能强大，效率高， 使用**rsh**或**ssh**可以自由切换，缺点是仅支持以太网。

使用

运行 `/public/software/mpi/ mpich2.sh`

设置临时环境变量，确认使用 `mpich2`

或参考: `/public/software/mpi/ mpich2.sh`的内容，

设置环境变量，确认长期使用 `mpich2`

`(#export PATH=/public/software/mpi/mpich2/bin:$PATH)`

编译**MPI**程序

```
#mpicc -o hello hello.c
```

```
#mpicc -show 可以看到实际调用编译器的情况
```

手动运行

```
vim ma
```

```
node1:8
```

```
node2:8
```

```
mpdboot -f ma -n 2 -r rsh
```

```
-n node number(节点数)
```

```
mpdtrace
```

```
mpirun -machinefile ma -np 16 ./ hello (或cpi等 )
```

```
mpdallexit
```

使用**pbs**

脚本内容如下， vi `mpich2.pbs`

```
#PBS -N mpi
```

```
#PBS -l nodes=2:ppn=8
```

```
#PBS -j oe
```

```
#PBS -l walltime=1000:00:00
```

```
cd $PBS_O_WORKDIR
```

```
NP=`cat $PBS_NODEFILE|wc -l`
```

```
cat $PBS_NODEFILE | uniq -c | awk '{ printf("%s:%s\n", $2, $1); }' >ma
```

```
NNODE=`cat hostfile|wc -l`
```

```
mpdboot -f ma -n $NNODE -r rsh
```

```
mpirun -machinefile ma -np $NP mpiprogram
```

```
mpdallexit
```

提交作业

```
qsub mpich2.pbs
```

为了快速高效地计算，对于CPU核心较多的节点，推荐使用 **openmpi**
OPENMPI

特点

支持**mpi2.0**,功能强大，灵活，支持各种网络，包括以太网和infiniband，效率高。使用rsh或ssh可以自由切换，路径可以自己标志，编译器也可以改，一个版本支持多种通讯方式。

使用

运行 /public/software/mpi/ openmpi.sh

设置临时环境变量，确认使用 OPENMPI

或参考： /public/software/mpi/ openmpi.sh的内容，

设置环境变量，确认长期使用 OPENMPI

(export PATH=/public/software/mpi/openmpi/bin:\$PATH

export LD_LIBRARY_PATH=/public/software/mpi/openmpi/lib:\$LD_LIBRARY_PATH

因为默认的openmpi编译出来的库为动态库，所以要设置

LD_LIBRARY_PATH变量，如果想要不设，在configure openmpi时加上

--disable-shared --enable-static 选项)

编译MPI程序

```
#mpicc -o hello hello.c
```

```
#mpicc -show 可以看到实际调用编译器的情况
```

手动运行：

```
#vim ma
```

```
node1 slots=8
```

```
node2 slots=8
```

(也可将node1和node2分别写8行)

slots=8代表每个结点上有8个cpu

```
mpirun -np 16 -machinefile ma --prefix /data/software/mpi/openmpi
```

```
--mca pls_rsh_agent rsh --mca btl self,tcp yourprogram
```

-np 16 起16个进程

-machinefile ma 文件写出在哪些节点上执行该命令，格式如下

```
--prefix /data/software/mpi/openmpi
```

如果有两个以上节点，告诉远端机器openmpi的安装路径

如果configure openmpi 时加 --enable-mpirun-prefix-by-default 则可不加--prefix

```
--mca pls_rsh_agent rsh 告诉节点间通讯用rsh
```

```
--mca btl self,tcp 使用以太网tcp/IP通讯
```

self,sm 当单节点运行的时候，使用内存通讯，效率高

self,openib 有infiniband设备时，使用IB通讯

```
--mac btl_tcp_if_include eth0 以太网通讯时用eth0通讯
```

使用pbs

脚本内容如下， vi mpich.pbs

```
#PBS -N mpi
```

```
#PBS -l nodes=2:ppn=8
```

```
#PBS -j oe
```

```
#PBS -l walltime=1000:00:00
```

```
cd $PBS_O_WORKDIR
```

```
NP=`cat $PBS_NODEFILE|wc -l`
```

```
mpirun -machinefile $PBS_NODEFILE -np $NP --prefix  
/data/software/mpi/openmpi --mca pls_rsh_agent rsh --mca btl  
self,tcp ./yourmpiprogram
```

注意：叙述中忽略了文件和目录的具体版本号，应在实际使用的环境中确认。

软件程序：

其他一些常用的软件和程序安装在 **/pub/software/**目录下，请先查看此目录中的应用程序再决定是否请管理员或自己安装其它应用程序。