第二海洋研究所高性能集群管理员手册

部　门：技术支持部

日　期：2020-05-31

单　位：杭州得到信息技术服务有限公司

1 系统概况

1.1 系统概况

- 计算节点共有 3 台 DELL R740 双路机架式服务器，每台计算节点配置 2 颗 Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz 24 核处理器; 8*32G 2666 256GB 内存；
- 系统配置登陆管理 1 台，配置 PBS 作业调度系统；
- 系统配置 1 套 100Gb EDR Infiniband 数据互通网络；
- 系统配置 1 套存储系统(96TB 裸空间)。

1.2 节点主机名与 IP 地址

主机名	角 色	管理 IP(16)	所内部 IP 地址	IB 地址(24)	ipmi 地址(16)
node1	计算节点	11.11.11.1	--	14.14.14.1	--
node2	计算节点	11.11.11.2	--	14.14.14.2	--
node3	计算节点	11.11.11.3	--	14.14.14.3	--
node100	管理节点	11.11.11.200	172.16.10.133	14.14.14.100	--

1.3 访问账户管理

计算节点与管理节点	账户	密码
node1~3,node100	root	dedao@123456

1.4 共享存储

存储规划		
存储节点或客户端	挂载点	备 注
node100	/data	可用容量：72TB

1.5 软件资源列表

并行环境软件			
软件名		版本	安装路径
1	Intel compiler	2017.5.239	/data/compiler/intel/intel-compiler-2017.5.239
2	openmpi	2.12	/data/mpi/openmpi/intel/2.1.2
			/data/mpi/openmpi/gnu/2.1.2
	Impi	2017.4.239	/data/mpi/intelmpi/2017.4.239
3	FFTW3	3.3.7	/data/mathlib/fftw/intelmpi/3.3.7_float
			/data/mathlib/fftw/intelmpi/3.3.7_double
4	hdf5	1.8.12	/data/mathlib/hdf5/intel/1.8.12
5	lapack	3.4.2	/data/mathlib/lapack/intel/3.4.2
6	netcdf	4.1.3	/data/mathlib/netcdf/intel/4.1.3
7	petsc	3.4.3	/data/mathlib/petsc/intelmpi/3.4.3
8	plasma	2.6.0	/data/mathlib/plasma/intel/2.6.0

1.6 作业队列设置

计算系统	节点表	节点描述	节点数	可用作业队列
计算节点(默认)	node1-3	2颗Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz 24核; 8*32G 2666 256GB内存；	3	dedao

2 使用集群进行计算业务

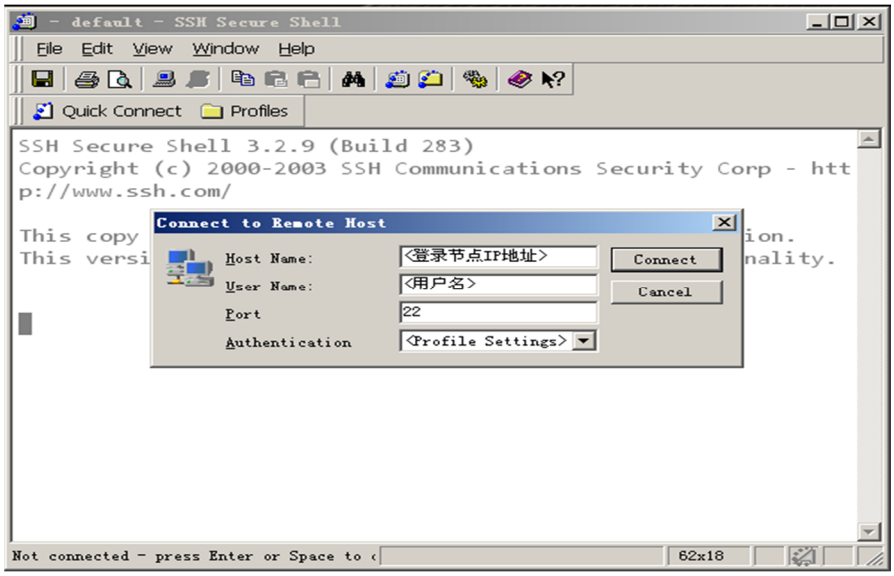
2.1 系统登陆

2.1.1 登陆管理 IP 地址

HPC 所有使用者登录到登入节点（172.16.10.133）上进行操作，并且供 HPC 使用者登录到管理节点上进行操作，作为集群的门户。

2.1.2 命令行终端登录

Windows 用户可以用 SSH Secure Shell Client，PuTTY，SecureCRT 等 SSH 客户端软件登录。推荐使用 SSH Secure Shell Client，它集成了 SFTP 文件上传下载功能。

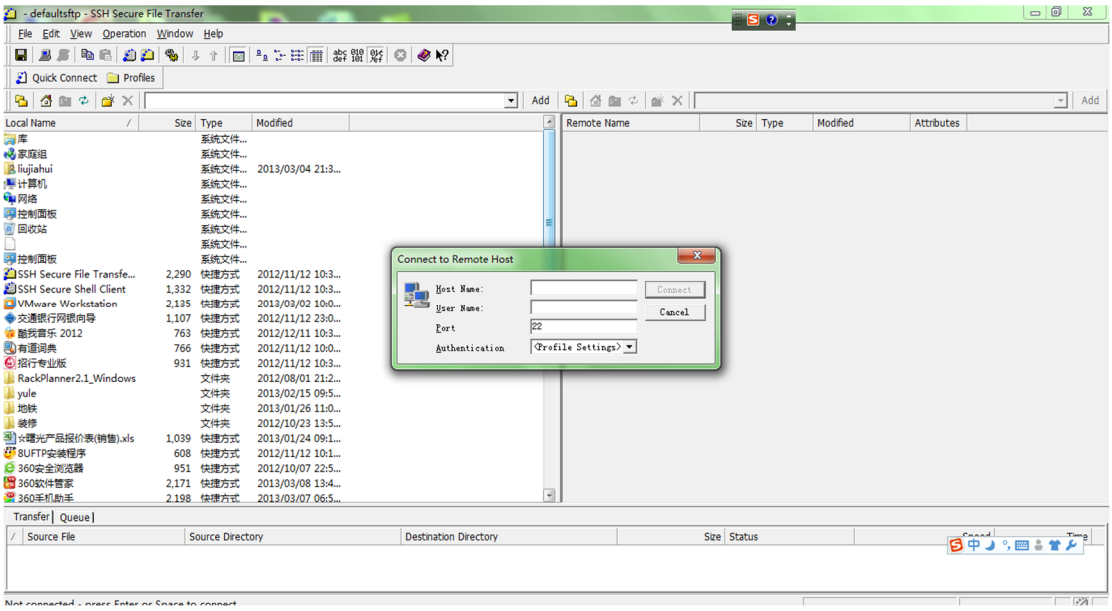


Linux 用户可以直接在命令行终端中执行 ssh 命令进行登录：

```
$ ssh username@登录节点 IP 地址
```

2.1.3 文件上传下载

Windows 用户可以用 SSH Secure Shell Client，winscp 等软件实现文件的上传下载。



Linux 操作系统直接使用命令

```
scp  filename  test@ip:/home/test
```

2.2 编译、安装 OpenMPI 示例

以 OpenMPI 1.6.3 为例：

```
$ tar zxvf openmpi-1.6.5.tar.gz
$ cd openmpi-1.6.3
$ ./configure --prefix=/public/software/mpi/openmpi-16-intel
--enable-mpirun-prefix-by-default  --without-psm CC=icc CXX=icpc FC=ifort F77=ifort
$ make -j 8 && make install
```

设置环境变量脚本：

```
vim /public/software/profile.d/openmpi-intel-env.sh
```

```
#!/bin/bash
export MPI_HOME=/public/software/mpi/openmpi-16-intel
export PATH=${MPI_HOME}/bin:${PATH}
export LD_LIBRARY_PATH=${MPI_HOME}/lib:${LD_LIBRARY_PATH}
export MANPATH=${MPI_HOME}/share/man:${MANPATH}
```

☑ **Tips：**

1. OpenMPI 安装会自动检测编译节点本地可用的通信网络设备，如需支持 InfiniBand 网络，请确保编译 MPI 前该节点已安装 OFED 驱动。

2. 执行 OpenMPI 安装目录$MPI_HOME/bin 下的 ompi_info 命令，可查询当前 OpenMPI 配置信息。

2.3 手动运行程序

2.3.1 运行串行程序

方法一

```
cd /home/your_account/your_workdir
./your_code
```

方法二

```
cd $HOME
vi .bashrc
```

添加

```
  export PATH=/home/your_account/your_workdir:$PATH
```

执行命令

```
your_code
```

2.3.2 使用 openmpi 运行并行程序

2.3.2.1 编译 MPI 程序

OpenMPI 提供了 C/C++，Fortran 等语言的 MPI 编译器，如下表所示：

语言类型	MPI 编译器
C	mpicc
C++	mpicxx
Fortran77	mpif77
Fortran90	mpif90

MPI 编译器是对底层编译器的一层包装，通过-show 参数可以查看实际使用的编译器。比如：

```
$ mpicc -show
icc -I/public/software/mpi/openmpi-16-intel/include -pthread
-L/public/software/mpi/openmpi-16-intel/lib -lmpi -ldl -lm -lnuma -Wl,--export-dynamic -lrt
-lnsl -lutil
```

编译程序示例：

```
$ source /public/software/profile.d/openmpi-intel-env.sh
$ mpicc -o hello hello.c
$ mpif90 -o hello hello.f90
```

2.3.2.2 运行 MPI 程序

OpenMPI 使用自带的 OpenRTE 进程管理器，启动命令为 mpirun/mpiexec/orterun，基本格式如下：

```
$ mpirun -np N -hostfile <filename> <program>
```

- -np N：运行 N 个进程
- -hostfile：指定计算节点，文件格式如下：

```
node1 slots=8
node2 slots=8
```

slots=8 代表可在该节点上执行 8 个进程，也可将 node1 和 node2 分别写 8 行。

2.3.3 使用 mvapich2 运行并行程序

2.3.3.1 编译 MPI 程序

```
$ source /public/software/profile.d/mvapich2-intel.sh
$ mpicc -o hello hello.c
$ mpif90 -o hello hello.f90
```

MVAPICH2 编译 C、C++、Fortran77 和 Fortran90 的编译器分别为 mpicc,mpicxx,mpif70 和 mpif90。

2.3.3.2 运行 MPI 程序

MVAPICH2 提供了两种进程管理器：mpirun_rsh/mpispawn 方式和 mpiexec/Hydra 方式。其中 mpirun_rsh/mpispawn 方式启动速度更快，支持集群规模更大，**但容易出现任务意外终止后计算节点存在僵尸进程的情况发生。虽然官方推荐使用 mpirun_rsh 方式，但曙光 HPC 推荐使用 mpiexec/Hydra 方式。**下面分别介绍：

1. **mpirun_rsh 命令**

```
$ mpirun_rsh -rsh -np 4 -hostfile hosts [ENV=value] ./program
```

- -rsh 或-ssh：指定使用 rsh 或 ssh 通信（默认 ssh）
- np：进程数，hostfile 文件格式与 MPICH2 相同（-np 和-hostfile 是必备选项，不可缺少）
- ENV=value 设置运行环境变量，如网络选择，进程绑定等，见下文。

2. **Hydra 方式**

```
mpiexe.hydra -launcher ssh -f hosts -n 4 [-env ENV value]./program
```

- -launcher ssh/rsh：指定启动远程任务的方式，默认 ssh
- -f hosts：格式同 MPICH2 相同，<node name>:<proc num>
- -n 4：指定进程数
- -env ENV=value：设置运行环境变量

2.4 命令行使用作业调度

2.4.1 PBS 的基本命令

在 PBS 系统中，用户使用 qsub 命令提交用户程序。

```
qsub xxx.pbs
```

用户运行程序的命令及 PBS 环境变量设置组成 PBS 作业脚本，提交格式如下：

注释,以“#”开头

PBS 指令,以“#PBS”开头

示例 OpenMPI 脚本：openmpi.pbs

```
#PBS -N openmpi
#PBS -l nodes=1:ppn=8
#PBS -j oe
#PBS -l walltime=2:00:00
cd $PBS_O_WORKDIR

echo my job id is $PBS_JOBID | tee  openmpi.log
echo run nodes is following: | tee -a openmpi.log
cat $PBS_NODEFILE | tee  -a openmpi.log

echo begin time is `date` | tee -a  openmpi.log
id=`echo $PBS_JOBID|awk -F. '{print $1}' `
NP=`cat $PBS_NODEFILE|wc -l`

mpirun -np $NP -hostfile $PBS_NODEFILE --mca orte_rsh_agent ssh  --mca btl
self,openib,sm  ./program  2>&1 | tee -a openmpi.log

echo end time is `date` | tee -a  openmpi.log
```

注意：算例规模的大小合理估算所需的 walltime 和 Mem，把其写进作业脚本里，这样有助于更快、更有效地分配资源；

2.4.2 查询队列信息

```
qmgr -c 'p s'
```

以 gpu 节点为例：

```
#
# Create queues and set their attributes.
#
#
# Create and define queue batch
#
create queue batch
set queue batch queue_type = Execution
set queue batch resources_default.cput = 99999:00:00
set queue batch resources_default.nodes = 1
set queue batch resources_default.walltime = 01:00:00
set queue batch enabled = True
```

```
set queue batch started = True
#
# Create and define queue dedao
#
create queue dedao
set queue dedao queue_type = Execution
set queue dedao resources_default.cput = 99999:00:00
set queue dedao resources_default.nodes = 1
set queue dedao resources_default.walltime = 99999:00:00
set queue dedao enabled = True
set queue dedao started = True
#
# Set server attributes.
#
set server scheduling = True
set server acl_hosts = node100
set server managers = root@node100
set server operators = root@node100
set server default_queue = dedao
set server log_events = 511
set server mail_from = adm
set server query_other_jobs = False
set server scheduler_iteration = 600
set server node_check_rate = 150
set server tcp_timeout = 6
set server job_stat_rate = 45
set server poll_jobs = True
set server mom_job_sync = True
set server keep_completed = 300
set server next_job_number = 1
set server moab_array_compatible = True
set server nppcu = 1
```

```
qmgr -c “set queue gpu acl_users += guest”
```

添加可使用该队列的用户 guest

2.4.3 查询节点信息 pestat

```
pestat |more
```

如下输出

```
 [root@node100 ~]# pestat
  node state  load    pmem ncpu   mem   resi usrs tasks  jobids/users
  node1  free  0.00  257184  96 273568   2863  0/0    0
  node2  free  0.00  257184  96 273568   2859  0/0    0
  node3  free  0.00  257184  96 273568   2869  0/0    0
```

excl ： 所有 CPU 资源已被占用；

busy ： CPU 已接近满负荷运行；

free ： 全部或部分 CPU 空闲；

offl ： 管理员手动指定离线状态；

2.4.4 查询作业运行状态

```
qstat -an |more
```

如下输出：

```
mgmt:
[root@node100 ~]# qstat -an 326

Node100:
                                                                  Req'd    Req'd
Elap
Job ID                  Username    Queue    Jobname         SessID  NDS   TSK   Memory
Time    S   Time
----------------------- ----------- -------- ---------------- ------ ----- ------ ------
--------- - ---------
326.node100             lihua       low      O-Pd             21541     1     16    --
240:00:00 R  75:08:02
   node86/0+node86/1+node86/2+node86/3+node86/4+node86/5+node86/6+node86/7
   +node86/8+node86/9+node86/10+node86/11+node86/12+node86/13+node86/14
   +node86/15
```

查询作业命令　qstat [参数]，其中参数可为:

-q　：列出系统队列信息

-B　：列出 PBS 服务器的相关信息

-Q ：列出队列的一些限制信息

-an：列出队列中的所有作业及其分配的节点

-r　：列出正在运行的作业

-f jobid　　：列出指定作业的信息

-Qf queue：列出指定队列的所有信息

2.4.5 删除作业

作业删除命令：qdel　作业号

注意事项

1、非 root 用户只能查看、删除自己提交的作业；

2、**强制删除作业**，当某些作业由于节点死机无法删除时，可由 root 用户登录，使用 qdel -p 作业号来强制删除作业

2.4.6 作业调度系统使用举例

2.4.6.1 串行作业

```
#!/bin/bash -x

#PBS -N serial
#PBS -l nodes=1:ppn=1
#PBS -l walltime=60:00:00
#PBS -j oe
#PBS -q serial

#
#define variables
#
echo "This jobs is "$PBS_JOBID@$PBS_QUEUE

cd ${PBS_O_WORKDIR}

date
sleep 100
hostname
date
```

#PBS -l nodes=1:ppn=1 表示申请 1 个节点上的 1 颗 CPU。

#PBS -q serial 表示提交到集群上的 serial 队列。

2.4.6.2 并行作业

- openmpi

并行作业的脚本以 cpi 为例。示例 OpenMPI 脚本：openmpi.pbs

```
#PBS -N openmpi
#PBS -l nodes=1:ppn=8
#PBS -j oe
#PBS -l walltime=2:00:00
cd $PBS_O_WORKDIR

echo my job id is $PBS_JOBID | tee  openmpi.log
echo run nodes is following: | tee -a openmpi.log
cat $PBS_NODEFILE | tee  -a openmpi.log

echo begin time is `date` | tee -a  openmpi.log
id=`echo $PBS_JOBID|awk -F. '{print $1}' `
NP=`cat $PBS_NODEFILE|wc -l`

mpirun -np $NP -hostfile $PBS_NODEFILE --mca orte_rsh_agent ssh  --mca btl
self,openib,sm   ./program  2>&1 | tee -a openmpi.log

echo end time is `date` | tee -a  openmpi.log
```

- mvapich2

MVAPICH2 示例脚本：mvapich2.pbs

```
#PBS -N mvapich2
#PBS -l nodes=1:ppn=8
#PBS -j oe
#PBS -l walltime=2:00:00
cd $PBS_O_WORKDIR

echo my job id is $PBS_JOBID | tee  mvapich2.log
echo run nodes is following: | tee -a mvapich2.log
cat $PBS_NODEFILE | tee  -a mvapich2.log

echo begin time is `date` | tee -a  mvapich2.log
id=`echo $PBS_JOBID|awk -F. '{print $1}' `
NP=`cat $PBS_NODEFILE|wc -l`

mpiexec.hydra -n $NP -launcher ssh -f $PBS_NODEFILE cpi  2>&1 | tee -a mvapich2.log

echo end time is `date` | tee -a  mvapich2.log
```

2.5 软件环境变量使用

2.5.1 脚本 stripts

环境变量存放目录：

```
/data/profile.d/
├── compiler_intel-compiler-2017.5.239.sh
├── mathlib_fftw-intelmpi-3.3.7_double.sh
├── mathlib_fftw-intelmpi-3.3.7_float.sh
├── mathlib_hdf5-intel-1.8.12.sh
├── mathlib_lapack-intel-3.4.2.sh
├── mathlib_netcdf-intel-4.1.3.sh
├── mathlib_petsc-intelmpi-3.4.3.sh
├── mathlib_plasma-intel-2.6.0.sh
├── mpi_intelmpi-2017.4.239.sh
├── mpi_mvapich2-intel-2.3b.sh
├── mpi_openmpi-gnu-2.1.2.sh
└── mpi_openmpi-intel-2.1.2.sh
```

使用参考：

source /data/profile.d/mathlib_plasma-intel-2.6.0.sh

数学库 plasma 环境变量激活

3 集群系统管理

2.6 系统开关机

集群系统设备的开启和关闭需要安装一定的顺序进行，如果不按照合理顺序进行，容易导致集群工作不正常。

2.6.1 集群系统开启的顺序

(一) 机柜上电。将机柜电源箱空开拨至“ON”状态，将每个机柜 PDU 的空开拨至“ON”状态。一般情况下，机柜上电后，会自动开启网络交换机、存储磁盘阵列、KVM 等设备。

(二) 确保已开启网络交换设备，包括以太网交换机、IB 交换机等。

(三) 开启存储（等待 5min）。

(四) 开启管理节点（node100），操作系统完全启动后，检查是否挂载上 IO 节点的网络共享存储。

(五) 开启计算节点（node1~3），其中开启刀片计算节点前，需要按刀片机箱电源按钮为刀片机箱上电。

2.6.2 集群系统关闭的顺序

集群系统关机上与开启顺序相反

(一) 关闭所有计算节点（node1~3）。

(二) 关闭登陆管理节点（node100）。

(三) 关闭存储。

2.7 用户管理

2.7.1 添加用户

添加用户需要使用 root 账户在管理节点(node100)上进行，需要确认添加的家目录为共享目录

```
$ [root@node100 dedao]# adduser-cluster test100 users 12345678
用户名：test100
组名：  users
密码：  12345678
```

2.7.2 删除用户

删除用户也需要使用 root 账户在管理节点上进行.

```
$ [root@node100 dedao]# userdel test100
```

也可以使用 Linux 自带命令，删除完成后进行用户同步

文档结束