南方科技大学
启明集群普通用户手册

南方科技大学科学与工程计算中心

2020 年 10 月

E-mail: hpc@sustech.edu.cn

目录

1.集群简介

集群采用 CPU+GPU 的异构计算架构，共 230 个双路刀片计算节点，7 个八路胖节点，6 个 GPU 计算节点，总体计算峰值高达 326.64 万亿次，其中 CPU 峰值 256.8 万亿次，GPU 峰值 69.84 万亿次。每个刀片计算节点配备两颗主频为 2.6GHz 的 Intel Xeon E5-2690v3 12 核处理器、64GB DDR4 内存，每个胖节点配备 8 颗主频为 2.3GHz 的 Intel Xeon E7-8880v3 18 核处理器、6T DDR4 内存，每个 GPU 节点配备两颗主频为 2.6GHz 的 Intel Xeon E5-2690v3 12 核处理器、144GB DDR4 内存、4 块 Nvidia Kepler K80 GPU 加速卡，管理网络采用的是以太万兆交换网络，并行数据交互采用是业界最快的 Mellanox 100GB Infiniband 网络。

集群的存储系统上层采用 lustre分布式文件系统、下层由24块1.8TB,53块10TB硬盘组成，可用空间将近400T，其高可用、高性能的特点满足了校级平台的存储需求。

2.系统软件环境

计算节点和管理登录节点的操作系统均为 64 位 CentOs-6.5，提供标准的 64 位 Linux 操作系统环境，用户需要熟悉一些基本的 Linux 命令行操作，并能熟练使用一种编辑器，如 vi 编译器。集群支持 OpenMP 和 MPI 两种并行方式，OpenMP 为共享内存方式，仅能在一个计算节点内并行，MPI 是分布式内存并行，可以在一个或多个节点上执行作业。集群已安装 intel 编译器，方面用户调用标准化数学库。

3.数学库

开放源代码程序往往要调用大量的数学函数进行各种计算，经过长期积累，已经有一些比较成熟的标准化数学库，其中最常见的诸如线性代数方面的 BLAS、LAPACK 等。集群已部署 intel 编译器，在调用相关的数学库时可用如下命令来设定环境变量：

```
source /opt/intel/composer_xe_2015/bin/compilervar.sh intel64
source /opt/intel/mkl/bin/intel64/mklvars_intel64.sh
source /opt/intel/impi/5.0.2.044/bin64/mpivars.sh
```

在使用 GPU 程序时需添加可执行文件和库:

```
export PATH=/usr/local/cuda-9.0/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-9.0/lib64:$LD_LIBRARY_PATH
```

4.集群节点设置说明

1 整个集群包含4 个登录节点，230 个刀片计算节点，7 个胖节点，6 个gpu 节点。

2 整个集群所有节点共享目录：/scratch,/home,/work,/opt。

3 登陆ip：172.18.6.67。

4 刀片计算节点：cu001-cu230。

5 胖节点：fat01-fat07。

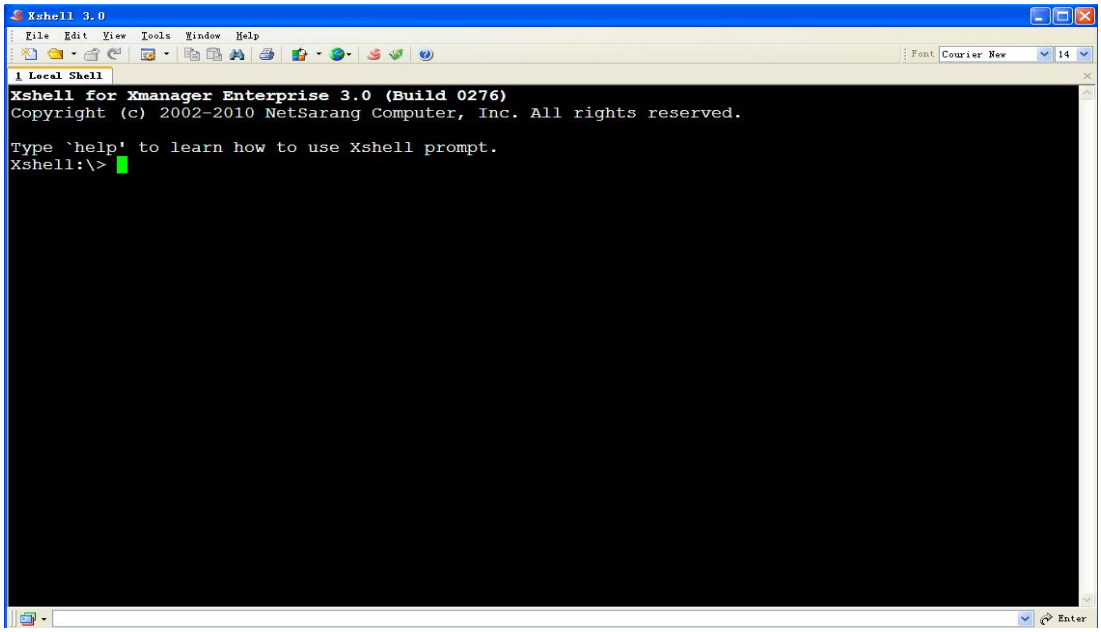
6 gpu 节点：gpu01-gpu06。

5.登录集群

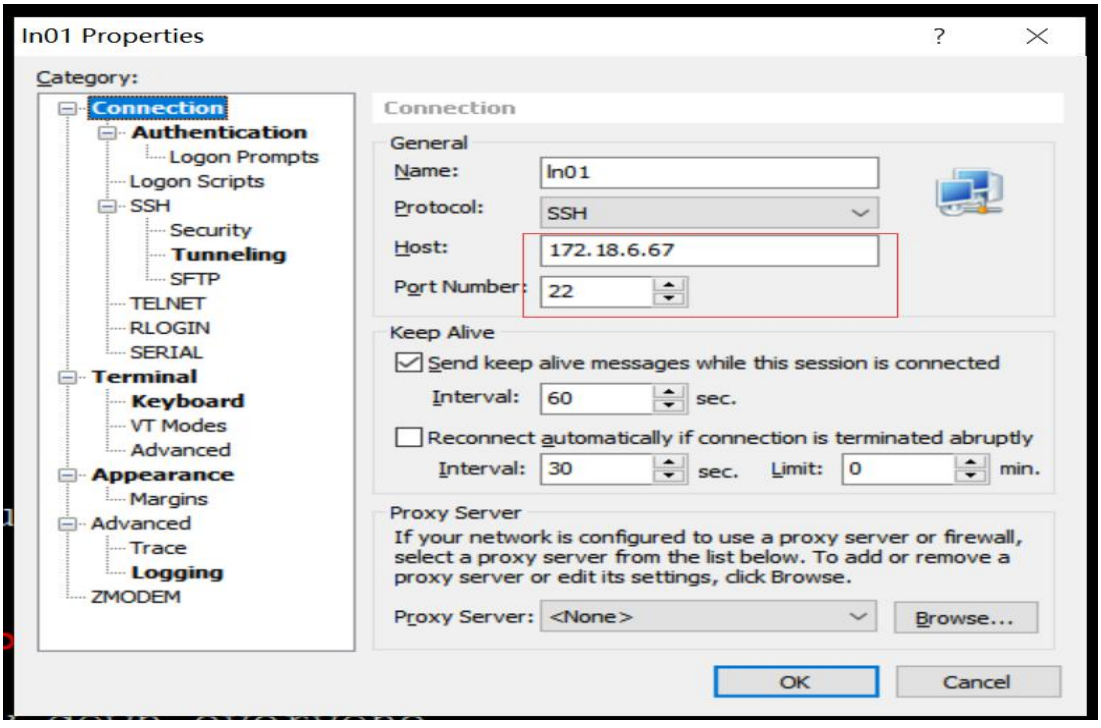
安装 PC 端 Xmanager 软件

直接默认下一步安装即可，注意中间输入注册码，若中间没输入注册码启动 xshell 会报错，可以在 help 里选项的 register Xshell 里再次输入注册码注册激活。

安装完后点击 Xshell 选项



添加主机：点击 file 里的 new 选项

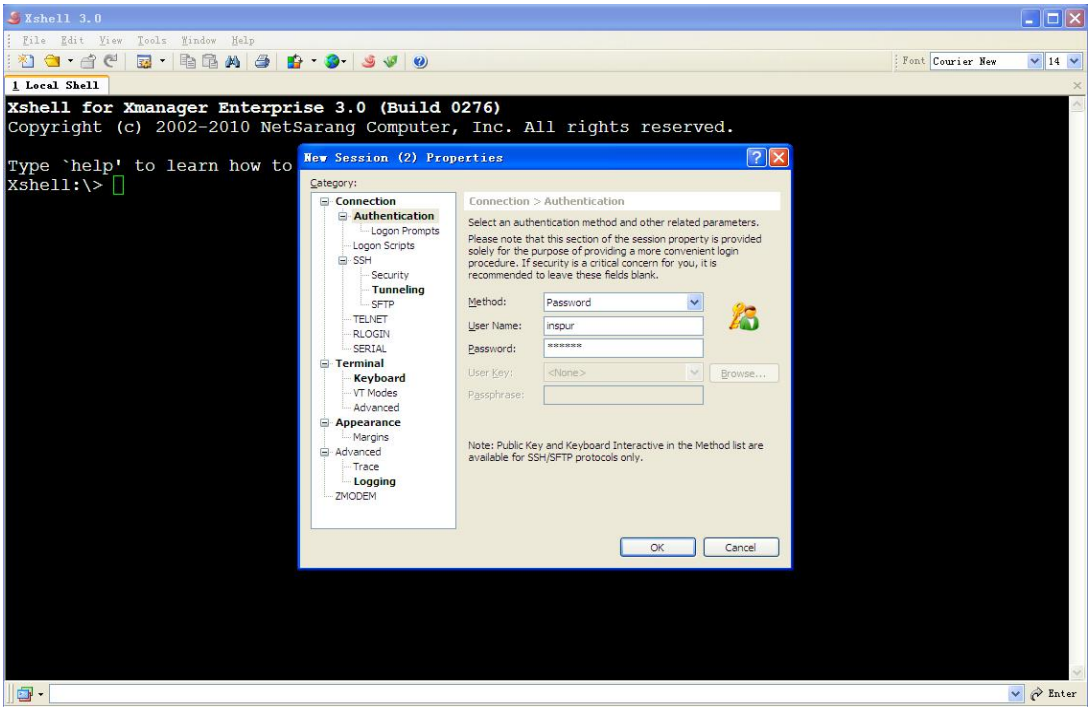


在此界面下，Connection 选项里，Name 填写一个名字用来识别你所添加的机器即可。

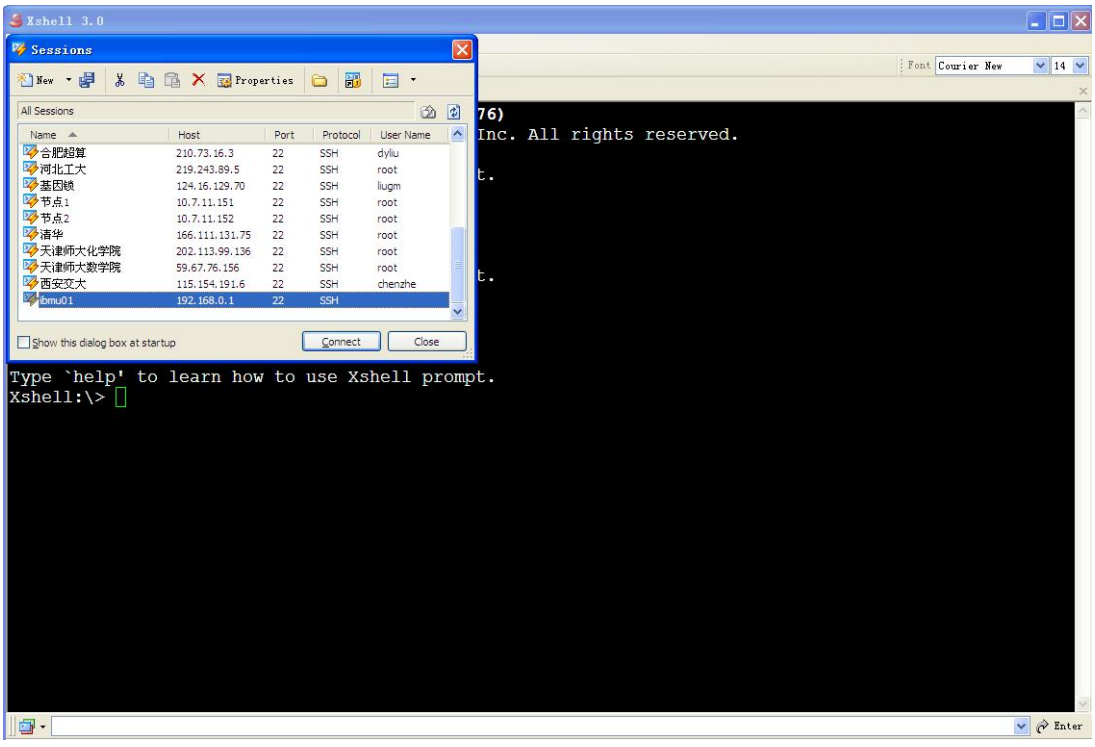
Host 选项登陆IP地址:172.18.6.67

Port Number: 22。

然后点击 Authentication 选项



此选项里，user Name 填写登陆用户名，password 填写登陆密码，填完后点击 OK，添加主机完毕



直接点解 connect 即可连上远程主机的 shell 里

以后连接主机，直接点击 open 选项里所保存的主机即可直接登陆

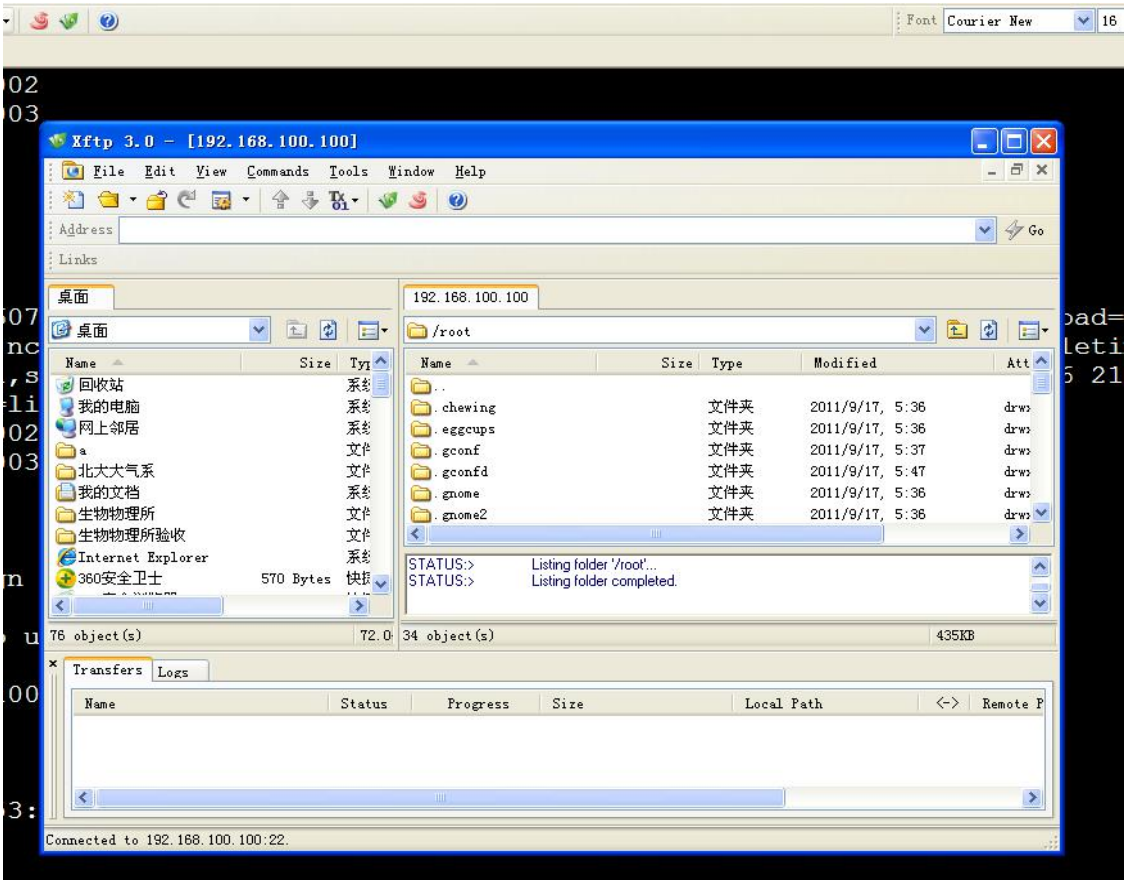
6.修改账号密码

ssh 登录集群后执行 yppasswd，按照提示输入自己的旧密码和新密码即可。

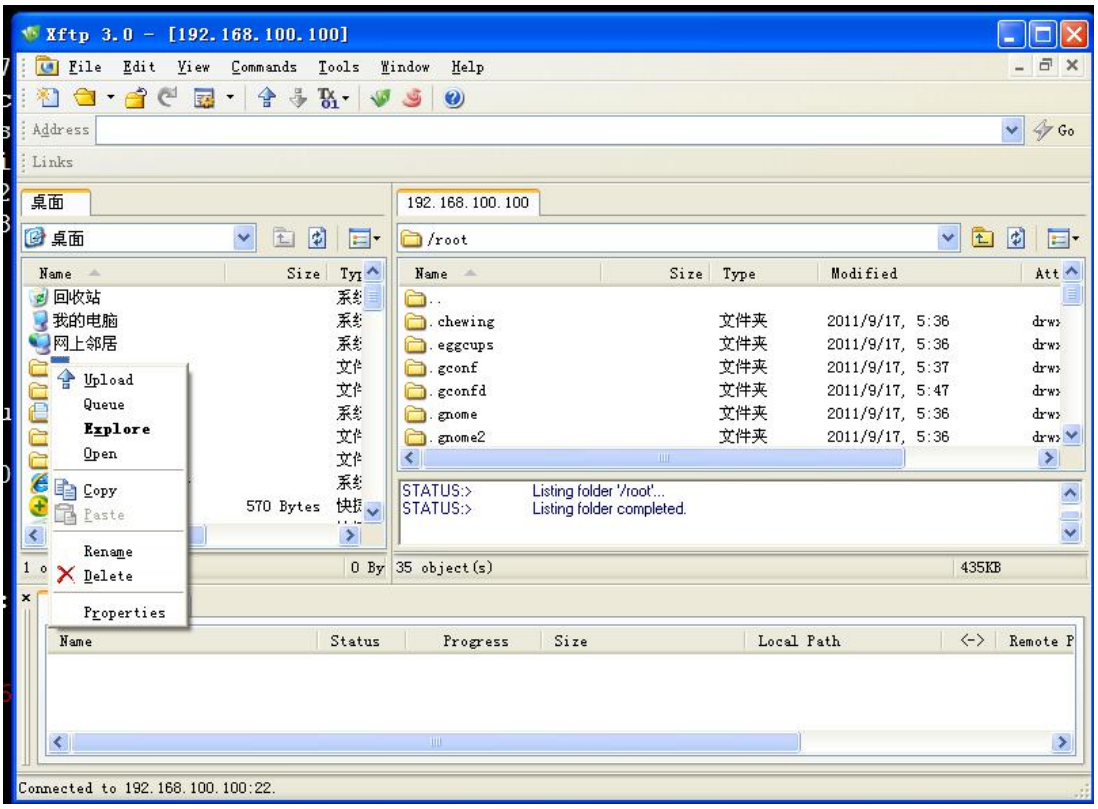
```
[inspur@ln01 ~]$ yppasswd
Changing NIS account information for inspur on mu01.
Please enter old password:
```

7.上传文件到集群

点击绿色的 new file transfer 按钮打开 xftp 工具



右键单击需要上传的文件或者文件夹，选择 upload 即可把文件上传到用户家目录下。



由于用户空间已经做了限制，查询自己硬盘使用量的方法为：（以用户 inspur 为例）

```
[          @ln03 ~]$ lfs quota -u          /home/ -h
Disk quotas for user          (uid 515):
     Filesystem    used   quota   limit   grace   files   quota   limit   grace
          /home/    264k     50G     56G       -      17       0       0       -
```

其中 used 参数为已经使用量。limit 参数为使用空间上限。

如果超过上限则无法继续写入文件，请及时清理。

8.pbs 系统队列划分

ser 队列：资源限制在 cu041-cu062，cu116-cu120，cu211-cu230,共47个节点，每个节点 2 个 CPU，每个 CPU 12 核，总核数为 1128，队列作业运行时间和核数没有限制。

cal-s 队列：资源限制在 cu001-cu040，cu081-cu115,cu121-cu150，cu191-cu210,共125个节点，每个节点 2 个 CPU，每个 CPU 12 核，总核数为 3000，队列作业最长运行时间为 240 小时，只能运行核数大于等于24核的作业。。

cal-m 队列：资源限制在 cu063-cu080，cu151-cu190，共58个节点。每个节点 2 个 CPU，每个 CPU 12 核，总核数为 1392。队列作业最长运行时间为 240 小时，只能运行核数大于等于120的作业。

fat 队列，资源限制在 fat01-fat07，每个节点 8 个CPU，每个 CPU 18 核，总核数为 1008，作业最长运行时间为 360小时，作业核数没有限制。

gpu 队列，资源限制在 gpu01-gpu06，每个节点 2 个CPU 和 8 个GPU 加速卡，每个 CPU 12 核， 总核数为 144，作业最长时间为 24 小时，作业核数没有限制。

9.通过 pbs 脚本提交作业

一般计算任务是通过脚本文件提交到作业管理系统的，脚本文件是一个常规文本文件，可以直接在登入节点使用 vi 编辑器编写，也可异地编写上传至用户作业工作目录，但要注意 dos2unix 转换一下。

脚本文件名无特殊规定，起一个可识别的名字即可。编辑完成脚本文件后，即可提交。例如对一个名称为 mytest.pbs 的作业脚本文件，编辑完成后用 qsub mytest.pbs 来提交，待作业提交成功后，会给出提交成功的作业 ID 号，用 qstat 命令可以查看作业的运行情况。

在提交作业以前可以在登陆节点上通过“pestat”命令查看当前集群资源使用情况，查看节点是否是空闲状态,tasks 表示分配的核数。

```
[inspur@mu01 ~]$ pestat
  node state  load    pmem ncpu   mem   resi usrs tasks  jobids/users
cu001  free  0.01   64374  24 128374   1275  1/1    0
cu002  free  0.01   64374  24 128374   1273  1/1    0
cu003  free  0.01   64374  24 128374   1274  1/1    0
cu004  free  0.00   64374  24 128374   1277  1/1    0
cu005  free  0.03   64374  24 128374   1270  1/1    0
cu006  free  0.01   64374  24 128374   1272  1/1    0
cu007  free  0.06   64374  24 128374   1269  1/1    0
cu008  free  0.00   64374  24 128374   1271  1/1    0
cu009  free  0.00   64374  24 128374   1273  1/1    0
cu010  free  0.00   64374  24 128374   1276  1/1    0
cu011  free  0.04   64374  24 128374   1275  1/1    0
cu012  free  0.00   64374  24 128374   1268  1/1    0
cu013  free  0.06   64374  24 128374   1279  1/1    0
cu014  free  0.00   64374  24 128374   1263  1/1    0
cu015  free  0.00   64374  24 128374   1264  1/1    0
cu016  free  0.02   64374  24 128374   1264  1/1    0
cu017  free  0.07   64374  24 128374   1265  1/1    0
cu018  free  0.00   64374  24 128374   1262  1/1    0
cu019  free  0.00   64374  24 128374   1269  1/1    0
```

以下以并行程序为例

1、

```
#!/bin/bash
#PBS -N test
#PBS -l nodes=1:ppn=24
#PBS -l walltime=12:00:00
#PBS -q cal-s
#PBS -V
#PBS -S /bin/bash

source /opt/intel/composer_xe_2015/bin/compilervars.sh intel64
source /opt/intel/mkl/bin/intel64/mklvars_intel64.sh
source /opt/intel/impi/5.0.2.044/bin64/mpivars.sh

date
mpirun -genv I_MPI_DEVICE rdssm -machinefile $PBS_NODRFILE -np 24 ./hello
date
```

脚本参数含义：

#PBS -N 指的是作业名

#PBS -l 指的是调用的系统资源，nodes=1：ppn=24 表示随机启动 1 个节点每个节点 24 核心

若要指定节点运行任务可写为 nodes=cu001：ppn=12+cu002：ppn=12,,,,

#PBS -l walltime=12:00:00 申请 12 小时的工作，不满足将无法继续进行计算

#PBS -q 表示要选择的队列

#PBS -V 表示将环境变量同步到计算节点

#PBS -S /bin/bash 表示让 pbs 脚本识别到 bash 命令

source /opt/intel/composer_xe_2015/bin/compilervars.sh intel64

source /opt/intel/mkl/bin/intel64/mklvars_intel64.sh

source /opt/intel/impi/5.0.2.044/bin64/mpivars.sh

表示的是环境变量

mpirun -genv I_MPI_DEVICE rdssm -machinefile /home/inspur/hello/host -n 24 ./hello

mpirun 为执行并行程序的程序，通过 pbs 为每个进程分配多个线程

-genv I_MPI_DEVICE rdma 指定 infiniband 设备

-machinefile $PBS_NODEFILE 指定所运行的节点，默认在/opt/tsce/server_priv_nodes

10. PBS 常用命令

qsub 命令

qsubpbs 脚本名称，提交作业

qstat 命令

用于查询作业状态信息

命令格式：qatat [-f][-a][-i] [-n][-s] [-R] [-Q][-q][-B][-u] +作业号

参数说明：

-f jobid 列出指定作业的信息

-a 列出系统所有作业

-i 列出不在运行的作业

-n 列出分配给此作业的结点

-s 列出队列管理员与 scheduler 所提供的建

-R 列出磁盘预留信息

-Q 操作符是 destination id，指明请求的是队列状态

-q 列出队列状态，并以 alternative 形式显示

-B 列 出 PBS Server 信 息

-r 列出所有正在运行的作业

-u 若操作符为作业号，则列出其状态。

例：# qstat 查看作业运行状态

```
[inspur@ln01 c7test]$ qsub pbs.linpack
2663.mu01
[inspur@ln01 c7test]$ qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
2663.mu01                  linpack          inspur                 0 Q cal
[inspur@ln01 c7test]$ qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
2663.mu01                  linpack          inspur                 0 R cal
```

qstat -f 查询作业的具体信息。

```
[inspur@ln01 c7test]$ qstat -f 2663
Job Id: 2663.mu01
    Job_Name = linpack
    Job_Owner = inspur@ln01
    job_state = R
    queue = cal
    server = mu01
    Checkpoint = u
    ctime = Thu Jun 16 16:10:35 2016
    Error_Path = ln01:/home/inspur/c7test/linpack.e2663
    exec_host = cu121/0+cu122/0+cu123/0+cu124/0+cu125/0+cu126/0+cu127/0+cu128/
        0+cu129/0+cu130/0+cu131/0+cu132/0+cu133/0+cu134/0+cu135/0+cu136/0+cu13
        7/0+cu138/0+cu139/0+cu140/0
    exec_port = 15003+15003+15003+15003+15003+15003+15003+15003+15003+15003+15
        003+15003+15003+15003+15003+15003+15003+15003+15003+15003
    Hold_Types = n
    Join_Path = n
```

qstat -an 查询当前所有作业所在的执行节点

```
                                                                  Req'd
Job ID                Username    Queue    Jobname          SessID NDS   TSK
--------------------- ----------- -------- ---------------- ------ ----- ------
992.mu01              inspur      fat      linpack           26061     1      0
   fat05/0
993.mu01              inspur      cal      linpack           20198   230      0
   cu001/0+cu002/0+cu003/0+cu004/0+cu005/0+cu006/0+cu007/0+cu008/0+cu009/0
   +cu010/0+cu011/0+cu012/0+cu013/0+cu014/0+cu015/0+cu016/0+cu017/0+cu018/0
   +cu019/0+cu020/0+cu021/0+cu022/0+cu023/0+cu024/0+cu025/0+cu026/0+cu027/0
   +cu028/0+cu029/0+cu030/0+cu031/0+cu032/0+cu033/0+cu034/0+cu035/0+cu036/0
   +cu037/0+cu038/0+cu039/0+cu040/0+cu041/0+cu042/0+cu043/0+cu044/0+cu045/0
   +cu046/0+cu047/0+cu048/0+cu049/0+cu050/0+cu051/0+cu052/0+cu053/0+cu054/0
   +cu055/0+cu056/0+cu057/0+cu058/0+cu059/0+cu060/0+cu061/0+cu062/0+cu063/0
   +cu064/0+cu065/0+cu066/0+cu067/0+cu068/0+cu069/0+cu070/0+cu071/0+cu072/0
   +cu073/0+cu074/0+cu075/0+cu076/0+cu077/0+cu078/0+cu079/0+cu080/0+cu081/0
   +cu082/0+cu083/0+cu084/0+cu085/0+cu086/0+cu087/0+cu088/0+cu089/0+cu090/0
   +cu091/0+cu092/0+cu093/0+cu094/0+cu095/0+cu096/0+cu097/0+cu098/0+cu099/0
   +cu100/0+cu101/0+cu102/0+cu103/0+cu104/0+cu105/0+cu106/0+cu107/0+cu108/0
   +cu109/0+cu110/0+cu111/0+cu112/0+cu113/0+cu114/0+cu115/0+cu116/0+cu117/0
   +cu118/0+cu119/0+cu120/0+cu121/0+cu122/0+cu123/0+cu124/0+cu125/0+cu126/0
```

qdel 命令

用于删除已提交的作业

命令格式：qdel 作业号

命令行参数：

例：# qdel 2624

```
[inspur@ln01 c7test]$ qdel 2663
[inspur@ln01 c7test]$ qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
2663.mu01                  linpack          inspur          00:00:57 C cal
```

其中，有一下几种状态

C：作业结束

Q：作业排队中

R：作业运行中

用户在使用时可用“du –sh”命令来查看当前空间、

```
[inspur@ln01 ~]$
[inspur@ln01 ~]$ du -sh
23G     .
[inspur@ln01 ~]$
```

如果不确定作业是否运行，可以使用 qstat -an 查看该作业所运行的节点，然后 ssh 到此节点上，执行 top 来查看是否有作业在运行，并且通过 cpu 利用率和进程数来判断程序运行情况。

11. 集群应用软件

集群应用软件安装在/opt/software 下。

```
[inspur@ln01 software]$ ls
abinit-7.10.5  MATLAB             openmpi-1.10.1-intel  phonopy-1.6.2      vasp.5.2
gcc-5.3        netcdf-4.3.2       openmpi-1.6.3-intel   python2.7          wannier90-1.2
hdf5-1.8.13    openmpi-1.10.1-gnu openmpi-1.6.5-gnu     QE-5.4.0-intelmpi
[inspur@ln01 software]$ pwd
/opt/software
[inspur@ln01 software]$
```

12. 注意事项

1、请不要在登陆节点上跑任务

2、如需要使用 openmp 方式跑作业，请把 pbs 申请的核心数与 opemmp 线程数设为一样。例如：

bwa 这是个典型的生命科学的应用，是个典型的 openmp 的应用，即一个进程调用多个线程的运行模式。这种模式，虽然只申请一个进程资源，但是在运行过程中会调用多个线程， 每个线程对应一个物理计算核心。所以在 top 中可以看到该进程 cpu 的使用率为 100%以上。

然而像这种 openmp 程序有自己的控制线程的参数。拿 bwa 来说：

```
bwa-0.7.5a mem -t 24
/useropt/public/IRGSP1.0/IRGSP ./R10.read1.fq ./R10.read2.fq >./test.sam
```

其中“-t 24”就是指定用 24 个线程来跑。

如果你在 pbs 脚本中申请资源为 1，但是下面申请 24 个线程，这样实际使用的为 24 个物理计算核心，但是 pbs 调度器却认为你只用了一个，这样会造成资源分配异常。为了避免这种情况，对于 openmp 程序，还请按照以下方式来提交作业。

```
#PBS -N bwa
#PBS -l nodes=1:ppn=24
#PBS -q cal-s
#PBS -V
#PBS -S /bin/bash

cd $PBS_O_WORKDIR
NP=`cat $PBS_NODEFILE | wc -l`
NN=`cat $PBS_NODEFILE | sort | uniq | tee /tmp/nodes.$$ | wc -l`
cat $PBS_NODEFILE > /tmp/nodefile.$$

mpirun -genv I_MPI_DEVICE rdssm -machinefile /tmp/nodefile.$$ -np 1
/useropt/bin/bwa-0.7.5a mem -t 24 /useropt/public/IRGSP1.0/IRGSP
$PBS_O_WORKDIR/R10.read1.fq $PBS_O_WORKDIR/R10.read2.fq >$PBS_O_WORKDIR/test.sam

rm -rf
/tmp/nodefile.$$ rm -rf
/tmp/nodes.$$
```

解析：就是你申请和你线程一样多的核数，但是在实际运行的时候-np 的值只跑一个进程。

附录：Linux 基本命令

目录操作

名称：cd

语法：cd [directory]

说明：把当前工作目录转到” directory”指定的目录。

实例：进入目录 /usr/bin/：

cd /usr/bin

名称 : ls

语法：ls [options] [pathname-list]

说明：显示目录内的文件名和“pathname-list”中指定的文件名

实例：列出目前工作目录下所有名称是 s 开头的文件：

ls s*

名称：pwd

语法：pwd

说明：显示当前目录的绝对路径。

名称：mkdir

语法：mkdir [options] dirName

说明：创建名称为 dirName 的子目录。

实例：在工作目录下，建立一个名为 AA 的子目录：

mkdir AA

名称：rmdir

语法：rmdir [-p] dirName

说明：删除空的目录。

实例：将工作目录下，名为 AA 的子目录删除 :

rmdir AA

文件操作

名称：cp

语法：cp [options] file1 file2

说明：复制文件 file1 到 file2。

常用选项：-r 整个目录复制

实例：将文件 aaa 复制(已存在)，并命名为 bbb ：

cpaaabbb

名称：mv

语法：mv [options] source... directory

说明：重新命名文件，或将数个文件移至另一目录。

范例：将文件 aaa 更名为 bbb ：

mv aaabbb

名称：rm

语法：rm [options] name...

说明：删除文件及目录。

常用选项：-f　强制删除文件实

例：删除除后缀名为.c 的文件rm

*.c

名称：cat

语法：cat[options] [file-list]

说明：在标准输出上连接、显示文件列表 file-list 里的文件

实例 1：显示 file1 和 file2 的内容

cat file1 file2

实例 2：将 file1 和 file2 合并成 file3

cat file1 file2 > file3

名称：more

语法：more[options] [file-list]

说明：在标准输出上连接、分页显示文件列表 file-list 里的文件

实例：分页显示文件 AAA

more AAA

名称：head

语法：head[options] [file-list]

说明：显示文件列表 file-list 中的文件的起始部分，默认显示 10 行；

实例：显示文件 AAA 起始部分

head AAA

名称：tail

语法：tail[options] [file-list]

说明：显示文件列表 file-list 中的文件的尾部；默认显示 10 行；

实例：显示文件 AAA 尾部

tail AAA

名称：ln

语法：ln[options] existing-file new-file

ln[options] existing-file-list directory

说明：为“existing-file”创建链接，命名为 new-file

在 directory 目录，为 existing-file-list”中包含的每个文件创建同名链接

常用选项：-f 不管 new-file 是否存在，都创建链接

-s 创建软链接

实例 1：建立软连接 temp.soft,指向 Chapter3

ln -s Chapter3 temp.soft

实例 2：为 examples 目录下的所有文件和子目录建立软连接

ln -s ~/linuxbook/examples/* /home/faculty/linuxbook/examples

名称：chmod

语法：chmod [option] mode file-list

说明：改变或设置参数 file-list 中的读、写或执行权限

实例：添加文件 job 的可执行权限

chmod +x job

语法：chmod [option] [files]

说明：备份文件。可用来建立备份文件，或还原备份文件。

实例 1：备份 test 目录下的文件，并命名为 test.tar.gz，可执行命令：

tar -zcvf test.tar.gz test

实例 2：解压缩相关的 test.tar.gz 文件，可执行命令：

tar -zxvf test.tar.gz

其他

名称：echo

语法：echo $variable

说明：显示变量 variable 的值。

实例 1：显示当前用户路径 PATH 的值

echo $PATH

名称：ps

语法：$ps [options]

说明：用于查看当前系统中的活跃进程

实例 1：显示当前所有进程

ps -aux

名称：kill

语法：$kill [-signal] pid

说明：终止指定进程

实例 1：终止 1511 号进程

kill 1511

附录 2：Vi 使用

简要使用流程

使用 "vi [选项] [文件 ..]" 命令打开要编辑的文件

使用 "方向键" 浏览文件

按下 "i" 进入编辑模式

编辑

按 "Esc" 键退出编辑模式

输入 ":w" 回车保存，再输入":q" 回车退出。或者直接输入 ":wq" 回车，代表保存并退出

两种操作模式

编辑模式：对文本进行编辑处理

命令模式: 接收按键指令执行操作，如复制、粘贴、搜索、替换、保存、另存为等

编辑模式

i: 进入编辑模式

a: 进入编辑模式，将光标向后移动一位o:

进入编辑模式，在光标处插入一个空行

r: 按下 r 键，再按任意字符键，将光标所在处的字符替换成新输入的字符

Esc: 退出编辑模式

命令模式

移动光标

↑或 k：把当前光标向上移动一行，保持光标的列位置。

↓或 j：把当前光标向下移动一行，保持光标的列位置。

→或 l：把当前光标向右移动一个字符。

←或 h：把当前光标向左移动一个字符。

$：把当前光标移动到该行行末。

^：把当前光标移动到该行行首。

w：把当前光标移动到该行的下一个字的首字符上。

b：把当前光标移动到该行的上一个字的首字符上。

e：把当前光标移动到该行的该字的末尾字符上。

^F：向前滚动一整屏正文。

^D：向下滚动半个屏正文。

^B：向后滚动一整屏正文。

^U：向上滚动半个屏正文。

搜索与替换

/word: 从光标处开始，向后搜索文本中出现 word 的字符串

?word: 从光标处开始，向前搜索文本中出现 word 的字符串

:1,$s/word1/word2/g: 在第 1 行与最后一行之间搜索 word1，并将其替换为 word2

:n1,n2s/word1/word2/g: 在第 n1 行与第 n2 行之间搜索 word1，并将其替换为 word2

删除 (剪切)、复制与粘贴

x,X: x 为向后删除一个字符，X 为向前删除一个字符

dd: 删除光标所在行

yy: 复制光标所在行的内容

nyy: 复制光标到第 1 行的所有内容

y1G: 复制光标到第 1 行的所有内容

yG: 复制光标到最后一行的所有内容

p,P: p 为将复制或剪切的内容粘贴在光标下一行，P 为粘贴在上一行u:

撤消上一操作

管理命令

:w:保存

:w!: 强制保存

:q:退出 vi 编辑器

:q!:强制退出

:w [文件名]:另存为..

:r[文件名]:读取另一个文档的内容，内容追加到光标所在行之后

:set nu:显示正文的行号。

:set nonu:取消行号。

:![命令]：暂时离开 vi 编辑器，并在 shell 中执行命令