

目录

一. 基本环境.....	1
1. 集群硬件环境.....	1
2. 集群网络环境.....	2
二. 应用部署.....	2
1. 应用软件部署.....	2
(1) 如要使用 <code>pyrhon-2.7.18</code> 只需执行如下命令.....	2
(2) 如要使用 <code>intel-2018</code> 编译 执行如下命令.....	2
2. 计算队列规划.....	2
三. 远程登录.....	2
1. 登录.....	2
2. 数据传输.....	3
3. 软件编译.....	4
1. 示例安装 <code>boost</code>	4
四. 作业提交.....	6
1. PBS 基本组件.....	6
2. PBS 命令提交作业.....	6
3. PBS 基本脚本格式.....	10
4. 查看用户组配额.....	11
5. jupyter 使用.....	12
6. VNC 使用.....	12
① 在 VNC 客户端输入 <code>111.186.40.49:2</code> 、.....	13
② 会有弹出框, 选择 <code>Continue</code>	13
③ 会有弹出框, 选择 <code>Continue</code>	13
④ 再一次弹出框, 输入免密.....	14
⑤ 登录完成.....	14
五. Linux 常用命令说明.....	15
1. 帮助命令.....	15
(1) <code>man</code> 获取相关的命令帮助信息.....	15
(2) <code>Info</code> 获取相关命令的详细使用方法.....	15
2. 操作命令.....	15

一. 基本环境

1. 集群硬件环境

集群共含 53 个节点、2 个千兆交换机、1 个万兆交换机、1 个 IB 交换机设备（交换机都是占据 1U 的空间），具体包括：

2 个管理/登录节点（`login01`、`login02`），每个占据 2U 的空间，内存 192GB，操作系统 `CentOS7.5`

36 个 4 路计算节点（gr01、gr02、gr03...gr36），每个占据 2U 的空间，内存 394GB，操作系统 CentOS7.5,每个节点包含 72 核，384G 内存

1 个 8 路计算节点也叫胖节点（fat01），占据 4U 的空间，操作系统 CentOS7.5，包含 192 核，6144G 内存。

14 个存储节点（inspur01-inspur14），每个占据 4U 的空间

2. 集群网络环境

ipmi: 硬件管理网络-千兆 ip 网段（192.168.11.1）

系统网络: 计算节点采用千兆网络，存储节点采用万兆网络 ip 网段（192.168.10.1）

计算网络: IB 网网段（10.0.0.1）

二. 应用部属

1. 应用软件部署

应用软件安装路径 /opt/

应用软件安装包路径 /opt/sourcecode/

普通用户若要安装自己的应用，可以直接安装在自己用户的家目录下/home/用户名

目前集群中已经安装的软件，用户如果需要设置其环境变量，在安装软件 /opt/module/modulefiles/软件名/版本号文件修改

(1) 如要使用 python-2.7.18 只需执行如下命令

```
module load python/python-2.7.18
```

(2) 如要使用 intel-2018 编译 执行如下命令

```
module load compiler/intel-2018
```

2. 计算队列规划

目前集群队列: normal、small、fat 和 all 队列,

normal 队列: gr01-gr32（总核心数 2304），默认任务提交队列，walltime（72 小时）;

small 队列: gr33-gr36（总核心数 288）;

fat 队列: fat01（总核心数 192）;

all 队列: gr01-gr36(总核心数 2592)。

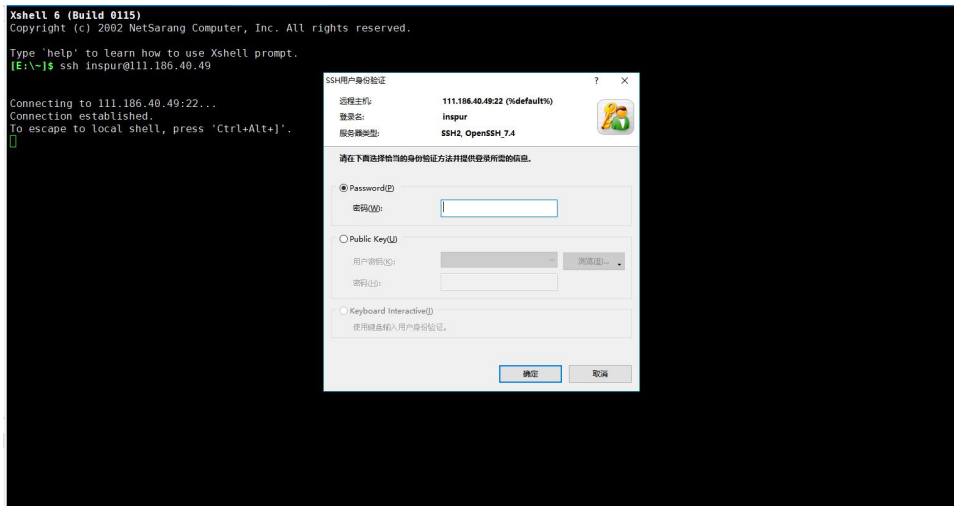
三. 远程登录

1. **登录**使用命令 `ssh 用户名@ip 地址`

```
Xshell 6 (Build 0115)
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[E:\~]$ ssh inspur@111.186.40.49
```

输入密码



登入集群

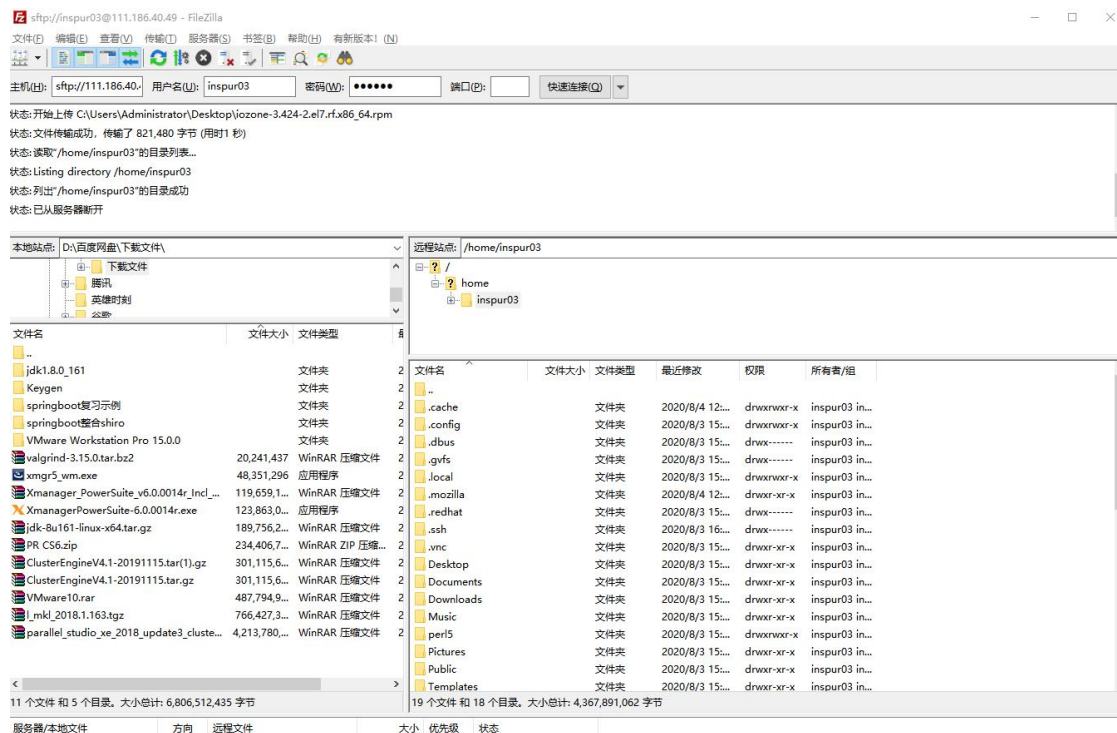
```
Last login: Wed Aug 12 13:16:06 2020 from 210.22.98.27
=====
Welcome to use Gravity cluster.
This cluster contains:
1. Thirty-six 4-cpu compute nodes(gr01-gr36), one node has 72 cpu cores and 384G memory
2. one 8-cpu compute node(fat01) with 192 cpu cores and 6144G memory
-----
A job scheduling system(pbs/torque),which configured with three queues:
1. normal(gr01-gr32)[default queue]
2. small(gr33-gr36)
3. fat(fat01)
the sample submit script for torque:
#PBS -N gr01-linpack
#PBS -l nodes=gr01:ppn=72
#PBS -q normal
#PBS -V
#PBS -S /bin/bash
#PBS -W group_list=project2
### Set intel environment###
module load mpi/intel-2018
cd $PBS_O_WORKDIR
NP=`cat $PBS_NODEFILE | wc -l`
cat $PBS_NODEFILE > /tmp/nodefile.$$
mpirun -machinefile /tmp/nodefile.$$ -n $NP EXEC
rm -rf /tmp/nodefile.$$
rm -rf /tmp/nodes.$$
notes: the group_list argument is your project name
-----
module can be used to manage the software in this cluster. Usage as follow:
1. module avail # list all software
2. module load modulefiles # load modulefiles for special software
3. module unload modulefiles # unload modulefiles
4. module list # list modulefiles loaded now
5. module switch modulefiles1 modulefiles2 # switch from modulefiles1 to modulefiles2
-----
enjoy it!
[inspur@login02 ~]$
```

2. 数据传输

安装 Filezilla 软件后，可以双击打开该软件，按照下图进行设置，完成后单

击快速链接或者回车可以登录 ftp 站点。

(1) 数据集 IP 用户名 密码和端口号



(2) 直接把需要传输的数据拉到具体目录中就行

3. 软件编译

用户登录到集群后，可以将自己需要运行的软件进行编译安装

普通用户（inspur）可以在自己家目录底下创建变量文件 /home/inspur/.modulerc 内容如下：

```
#%Module1.0#####
prepend-path MODULEPATH /home/inspur/module/modulefile
```

1. 示例安装 boost

(1) 解压，编译，安装

解压 boost_1_73_0.tar.gz

>> cd /home/inspur/sourcecode

```
[inspur@login02 ~]$ cd sourcecode/
[inspur@login02 sourcecode]$ ls
boost_1_73_0.tar.gz  hdf5-1.12.0.tar
```

>> tar -xvf boost_1_73_0.tar.gz (解压)

```
[inspur@login01 sourcecode]$ ls
boost_1_73_0.tar.gz hdf5-1.12.0.tar
[inspur@login01 sourcecode]$ tar -xvf boost_1_73_0.tar.gz
```

>> cd boost_1_73_0 (进入目录)

```
[inspur@login01 sourcecode]$ ls
boost_1_73_0 boost_1_73_0.tar.gz hdf5-1.12.0.tar
[inspur@login01 sourcecode]$ cd boost_1_73_0/
[inspur@login01 boost_1_73_0]$ ls
boost boost-build.jam boostcpp.jam boost.png bootstrap.sh index.htm INSTALL Libs more rst.css tools
boost-build.jam boost.css bootstrap.bat doc index.html Jamroot LICENSE_1_0.txt README.md status
```

>> ./bootstrap.sh --prefix=/home/inspur/Downloads/boost_1_73_0 (编辑 boost, 选择安装路径)

```
[inspur@login01 boost_1_73_0]$ ./bootstrap.sh --prefix=/home/inspur/Downloads/boost_1_73_0
Building Boost.Build engine with toolset gcc... tools/build/src/engine/b2
Detecting Python version... 2.7
Detecting Python root... /usr
Unicode/ICU support for Boost.Regex?... /usr
Generating Boost.Build configuration in project-config.jam for gcc...

Bootstrapping is done. To build, run:

./b2
```

>> ./b2 install

```
[inspur@login01 boost_1_73_0]$ ./b2 install
```

(2) 设置软件环境变量

首先到/home/inspur/module/modulefile/目录下创建文件夹 boost

```
[inspur@login01 modulefile]$ ls
boost
```

再到 boost 文件下创建文件 1_73_0,可以复制其他的修改内容

```
##Module1.0#####
##
## modules modulefile
##
## modulefiles/modules. Generated from modules.in by configure.
##
proc ModulesHelp { } {
    global version prefix

    puts stderr "\tmodules - loads the modules software & application environment"
    puts stderr "\n\tThis adds $prefix/* to several of the"
    puts stderr "\tenvironment variables."
    puts stderr "\n\tVersion $version\n"
}

module-whatism "boost 1_73_0"

set version 1_73_0
set prefix /home/inspur/Downloads/boost_1_73_0

set root $prefix

prepend-path PATH $root/bin
prepend-path MANPATH $root/man
prepend-path LIBRARY_PATH $root/lib
prepend-path LD_LIBRARY_PATH $root/lib
prepend-path LD_RUN_PATH $root/lib
prepend-path CPATH $root/include

~
~
```

19, 34-46 All

文件参数意义

```
$ module avail      # 列出所有可用的变量
$ module load python/python-2.6.6      # 加载 python-2.6.6
$ module unload python/python-2.6.6    # 卸载 python-2.6.6
$ module switch python python/python-2.6.6  # 将 python 转换为 2.6.6 版本
```

说明: modulefile 一些基本的变量

```
1. append-path variable value      # 后加一些环境变量
2. prepend-path variable value     # 前置一些环境变量
3. prereq modulefile              # 需要一些 modulefile 文件
4. conflict modulefile            # 与一些变量冲突
5. set-alias alias-name alias-string  # 设置别名
```

说明 2: module 一些基本的参数

```
1. help modulefile      # modulefile 的帮助文档
2. add modulefile       # 增加一个 modulefile
3. load modulefile      # 加载一个 modulefile
4. unload modulefile    # 卸载一个 modulefile
5. switch modulefile1 modulefile2  # 将 modulefile1 转化为 modulefile2, 如果 modulefile1 没有指定, 默认是当前已经加载的与 modulefile2 具有相同的目录文件的模块
6. display modulefile   # 演示一个或多个 modulefile 文件的信息, display 子命令将展示出 modulefile 加载后环境变化后的结果
7. list                # 列出目前已经加载的模块
8. avail               # 列出系统提供的所有 module 模块
```

四. 作业提交

1. PBS 基本组件

pbs command:用于提交、监视、修改和删除作业

pbs_server: 服务器, 提供基本的批处理服务, 例如接收/创建一个批处理作业, 管理维护作业队列, 管理输出结果等

pbs_mom:执行器, 是一个守护进程, 从 pbs server 处接收作业后放入其执行队列中等待执行

maui:调度器, 检查作业并分配资源, 对用户提交的作业进行调度

2. PBS 命令提交作业

(1) qsub ./pbs 脚本 提交一个新的作业

```

[inspur@login02 gr19]$ qsub ./linpack.pbs

```

(2) qstat 参数 查看作业状态 job ID 作业编号, Name 作业名, user 提交作业用户, Time use 运行时间 Queue 提交队列 S 代表状态 R 代表运行, Q 代表排队, C 代表运算完毕或者退出

```

[inspur@login02 ~]$ qstat
Job ID          Name          User          Time Use S Queue
-----
244.login01     Gadget        lizz          2253:28: R normal
258.login01     pcigale       kunxu         2050:13: R fat
262.login01     Gizmo         zli           3018:01: R normal
272.login01     Phantom       zli           3602:36: R small
347.login01     athena++      zli           1507:41: R normal
370.login01     file_create   kunxu         00:57:11 R fat

```

(3) qstat -a 查看作业 比着直接使用 qstat 更详细

```

[inspur@login02 gr19]$ qstat -a
login01:
Job ID          Username      Queue    Jobname      SessID  NDS   TSK   Req'd      Req'd      Elap
Memory          Time      S      Time
-----
576.login01     inspur        normal   inspur-test-3 316030   2    144    --    7200:00:0 R    00:04:28

```

(4) qdel 作业编号 取消指定的作业 canceljob 也可以取消作业

```

[inspur@login02 gr19]$ qstat
Job ID          Name          User          Time Use S Queue
-----
573.login01     testcopy      inspur        00:00:00 C normal
[inspur@login02 gr19]$ qdel 573.login01

```

(5) checkjob 作业编号 显示作业状态、资源需求、环境、限制、信任、历史、已分配资源和资源利用等

```

[inspur@login02 gr19]$ checkjob 575.login01

checking job 575

State: Running
Creds: user:inspur group:project2 class:normal qos:DEFAULT
WallTime: 00:00:33 of INFINITY
SubmitTime: Fri Aug 14 12:52:04
  (Time Queued Total: 00:00:04 Eligible: 00:00:04)

StartTime: Fri Aug 14 12:52:08
Total Tasks: 144

Req[0] TaskCount: 144 Partition: DEFAULT
Network: [NONE] Memory >= 0 Disk >= 0 Swap >= 0
Opsys: [NONE] Arch: [NONE] Features: [normalPAR]
Allocated Nodes:
[gr32:72][gr31:72]

IWD: [NONE] Executable: [NONE]
Bypass: 0 StartCount: 1
PartitionMask: [ALL]
Flags:          RESTARTABLE

Reservation '575' (-00:00:33 -> INFINITY Duration: INFINITY)
PE: 144.00 StartPriority: 1

```

(6) nqs2pbs: 将 nqs 作业脚本转换为 pbs 作业脚本

(7) pbsnodes: 显示节点信息

```
[inspur@login02 gr19]$ pbsnodes
gr01
  state = free
  power_state = Running
  np = 72
  properties = normalPAR
  ntype = cluster
  status = rectime=1597388234,macaddr=b4:05:5d:2f:e8:b2,cpuClock=Fixed,varattr=,jobs=,state=free,size=488678924kb:427568504kb,netload=9255171548238,gres=,loadave=0.00,ncpus=72,physmem=394631944kb,availmem=423238892kb,totmem=427399948kb,idletime=158156,nusers=0,nsessions=0,uname=Linux gr01 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64,opsys=Linux
  mom_service_port = 15002
  mom_manager_port = 15003

gr02
  state = free
  power_state = Running
  np = 72
  properties = normalPAR
  ntype = cluster
  status = rectime=1597388276,macaddr=b4:05:5d:37:0f:92,cpuClock=Fixed,varattr=,jobs=,state=free,size=488683628kb:427568504kb,netload=8531974338283,gres=,loadave=0.00,ncpus=72,physmem=394631992kb,availmem=423388164kb,totmem=427399988kb,idletime=678982,nusers=0,nsessions=0,uname=Linux gr02 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64,opsys=Linux
  mom_service_port = 15002
  mom_manager_port = 15003

gr03
  state = free
  power_state = Running
  np = 72
  properties = normalPAR
  ntype = cluster
  status = rectime=1597388271,macaddr=b4:05:5d:37:11:90,cpuClock=Fixed,varattr=,jobs=,state=free,size=488552828kb:427568504kb,netload=683261988537,gres=,loadave=0.00,ncpus=72,physmem=394631992kb,availmem=423388164kb,totmem=427399988kb,idletime=678982,nusers=0,nsessions=0,uname=Linux gr03 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64,opsys=Linux
  mom_service_port = 15002
  mom_manager_port = 15003
```

(8) pbsnodes -l free 列出空闲的节点

```
[inspur@login02 gr19]$ pbsnodes -l free
gr01 free
gr02 free
gr03 free
gr04 free
gr05 free
gr06 free
gr07 free
gr08 free
gr09 free
gr10 free
gr11 free
gr12 free
gr13 free
gr14 free
gr15 free
gr16 free
gr17 free
gr18 free
gr19 free
gr20 free
gr21 free
gr22 free
gr23 free
gr24 free
gr25 free
gr26 free
gr27 free
```

(9) printjob: 显示指定脚本中的作业信息

```
[inspur@login02 gr19]$ printjob linpack.pbs
Short read of 171 bytes, file linpack.pbs
linpack.pbs contains an old version of the ji_qs structure.
  expecting version 0x00020302, read 0x53425023
  Skipping prt_job_struct()
  pbs_server may be able to upgrade job automatically
```


(10) showq 命令 可以查看作业编号, 用户名, 运行状态, remaining 剩余时间, infinity 无限, starttime 开始时间

```
[inspur@login02 ~]$ showq
```

ACTIVE JOBS-----							
JOBNAME	USERNAME	STATE	PROC	REMAINING			STARTTIME
262	zli	Running	144	19:04:18	Fri Aug 7	16:39:00	
272	zli	Running	72	20:45:52	Fri Aug 7	18:20:34	
371	zli	Running	256	2:23:04:38	Sun Aug 9	20:39:20	
244	lizz	Running	400	INFINITY	Fri Aug 7	12:02:32	
258	kunxu	Running	150	INFINITY	Fri Aug 7	16:02:09	
375	kunxu	Running	72	INFINITY	Sun Aug 9	21:01:42	
376	kunxu	Running	72	INFINITY	Sun Aug 9	21:02:26	
378	kunxu	Running	72	INFINITY	Sun Aug 9	21:03:43	
379	kunxu	Running	72	INFINITY	Sun Aug 9	21:04:05	
380	kunxu	Running	72	INFINITY	Sun Aug 9	21:04:38	
381	kunxu	Running	72	INFINITY	Sun Aug 9	21:05:00	
382	kunxu	Running	72	INFINITY	Sun Aug 9	21:05:22	
383	kunxu	Running	72	INFINITY	Sun Aug 9	21:05:33	
385	kunxu	Running	72	INFINITY	Sun Aug 9	21:07:12	
386	kunxu	Running	72	INFINITY	Sun Aug 9	21:07:34	
387	kunxu	Running	72	INFINITY	Sun Aug 9	21:07:45	

(11) qhold: 挂起一个作业 被挂起的作业在用 qstat 命令查询时显示的状态标志为 H

(12) qmove: 将一个作业从一个队列移到另一个队列中

(13) qnodes: pbsnodes 的别名, 显示节点信息

```
[inspur@login02 gr19]$ qnodes
gr01
  state = free
  power_state = Running
  np = 72
  properties = normalPAR
  ntype = cluster
  status = rectime=1597381674,macaddr=b4:05:5d:2f:e8:b2,cpuclock=Fixed,varattr=,jobs=,state=free,size=408670872kb:427568
504kb,netload=10105835260464,gres=,loadave=0.00,ncpus=72,physmem=394631944kb,availmem=423220556kb,totmem=427399940kb,idletim
e=159596,nusers=0,nsessions=0,uname=Linux gr01 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64,opsys=linu
x
  mom_service_port = 15002
  mom_manager_port = 15003
gr02
  state = free
  power_state = Running
  np = 72
  properties = normalPAR
  ntype = cluster
  status = rectime=1597381671,macaddr=b4:05:5d:37:0f:92,cpuclock=Fixed,varattr=,jobs=,state=free,size=408603584kb:427568
504kb,netload=8531985789329,gres=,loadave=0.00,ncpus=72,physmem=394631992kb,availmem=423303968kb,totmem=427399988kb,idletim
e=672377,nusers=0,nsessions=0,uname=Linux gr02 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64,opsys=linux
```

(14) qorder: 交换两个作业的排队顺序

(15) qrls: 将被挂起的作业送入准备运行的队列中

(16) qselect: 显示扶额和条件的作业的作业号

```
[inspur@login02 gr19]$ qselect -s R
577.login01
578.login01
```

(17) showbf: 显示有特殊资源需求的资源的可用性

(18) showstart: 显示空闲作业的估计开始时间

(19) tracejob: 追踪作业信息

(20) pstat 命令 查看计算节点使用情况 pmem是总物理内存, mem是物理内存加上 32G 缓存, resi 是已用内存

```
[root@login02 ~]# pestat
```

node	state	load	pmem	ncpu	mem	resi	usrs	tasks	jobids/users
gr01	free	0.00	385382	72	417382	4815		0/0	0
gr02	free	0.00	385382	72	417382	5027		0/0	0
gr03	free	0.00	385382	72	417382	5083		0/0	0
gr04	free	0.00	385382	72	417382	4829		0/0	0
gr05	free	0.00	385382	72	417382	5006		0/0	0
gr06	free	0.00	385382	72	417382	5541		0/0	0
gr07	free	0.00	385382	72	417382	5012		0/0	0
gr08	free	0.00	385382	72	417382	5004		0/0	0
gr09	free	0.00	385382	72	417382	5022		0/0	0
gr10	free	0.00	385382	72	417382	5068		0/0	0
gr11	free	0.00	385382	72	417382	5090		0/0	0
gr12	free	0.00	385382	72	417382	5068		0/0	0
gr13	free	0.00	385382	72	417382	5065		0/0	0
gr14	free	0.00	385382	72	417382	5046		0/0	0
gr15	free	0.00	385382	72	417382	5124		0/0	0
gr16	free	0.00	385382	72	417382	4790		0/0	0
gr17	free	0.00	385382	72	417382	4841		0/0	0
gr18	free	0.00	385382	72	417382	5054		0/0	0
gr19	free	0.00	385382	72	417382	6038		0/0	0
gr20	free	0.00	385382	72	417382	4901		0/0	0
gr21	free	0.00	385382	72	417382	4143		0/0	0
gr22	free	0.00	385382	72	417382	4998		0/0	0
gr23	free	0.00	385382	72	417382	5036		0/0	0

3. PBS 基本脚本格式

脚本包含三部分：

资源声明：即规定所需要的节点数，核数，作业名，所要递交的队列

环境变量：即运行作业时，需要的各个节点的基本属性，比如某些软件的路径等

可执行程序：即需要通过 MPI 来运行的并程序

如下例子说明

脚本声明部分：

```
#PBS -N vasp                \\设定应用程序名字
#PBS -l nodes=2:ppn=12      \\启动 2 个节点每个节点 12 个核心
#PBS -l walltime=999:00:00  \\申请 999 小时的工作，不满足将无法继续进行计算
#PBS -q normal              \\申请 normal 队列
#PBS -W group_name          \\以项目组名义申请资源
```

环境变量部分：

```
### intel ###              \\intel 环境变量生效
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
```

可执行程序部分：

```
cd $PBS_O_WORKDIR
NP=`wc -l < $PBS_NODEFILE`
mpirun -genv I_MPI_DEVICE rdma -machinefile $PBS_NODEFILE -np $NP $PBS_O_WORKDIR/vasp
\\应用软件参数
```

举例

```
[inspur02@login02 run999]$ cat job.s
#PBS -N test15
#PBS -l nodes=1:ppn=4
#PBS -l mem=16gb
#PBS -l walltime=48:00:00
#PBS -q normal
#PBS -S /bin/bash

### Set intel environment###

#source /opt/intel/compilers_and_libraries/linux/bin/compilervars.sh intel64
#source /opt/intel/impi/2018.1.163/bin64/mpivars.sh
module load mpi/intel-2018

# job start

cd /home/inspur02/run999

# run galaxy
mpirun -np 4 ./galaxy_mpi << eof
999 # run number
0 # time of .dmp file
eof
```

#PBS -N test15 \\设定应用程序名字

#PBS -l nodes=1:ppn=4 \\启动 1 个节点每个节点 4 个核心

#PBS -l walltime=48:00:00 \\申请 48 小时的工作，不满足将无法继续进行计算

#PBS -l mem=16gb \\申请 16gb 的内存

#PBS -q normal \\申请 normal 队列

###Set intel environment###设置环境变量

Module load mpi/intel-2018 加载环境变量

#job start 工作开始

cd /home/inspur02/run999

备注:一般的有问题的作业会在 5-30 秒内变成状态 c，然后从 qstat 中消失，建议提交作业后检查一下 qstat，若作业状态为 c 或消失，需要查看作业 e 和 o 文件，根据日志信息分析报错原因 文件名最后的数字为作业 id

4. 查看用户组配额

命令:icfsquota 用户名

```
[inspur@login02 gr19]$ icfsquota inspur
max_size=3T,used=1.1TB
[inspur@login02 gr19]$ icfsquota yuyu22
max_size=30T,used=18.4TB
[inspur@login02 gr19]$
```

5. jupyter 使用

命令: `jupyter-lab` 不建议使用 `root` 用户运行, 可以使用 `--allow-root` 绕过, 普通用户直接使用命令 `jupyter-lab` 就行

```
[root@login01 ~]# jupyter-lab --allow-root
[I 13:17:41.644 LabApp] JupyterLab extension loaded from /opt/python-2.7.18/lib/python2.7/site-packages/jupyterlab
[I 13:17:41.644 LabApp] JupyterLab application directory is /opt/python-2.7.18/share/jupyter/lab
[W 13:17:41.647 LabApp] JupyterLab server extension not enabled, manually loading...
[I 13:17:41.647 LabApp] JupyterLab extension loaded from /opt/python-2.7.18/lib/python2.7/site-packages/jupyterlab
[I 13:17:41.647 LabApp] JupyterLab application directory is /opt/python-2.7.18/share/jupyter/lab
[I 13:17:41.652 LabApp] Serving notebooks from local directory: /root
[I 13:17:41.652 LabApp] The Jupyter Notebook is running at:
[I 13:17:41.652 LabApp] http://localhost:8888/?token=517e756c2cfacf2357a06597b2a21e50ea3895ed05af7321
[I 13:17:41.652 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:17:41.655 LabApp]

To access the notebook, open this file in a browser:
file:///root/.local/share/jupyter/runtime/nbserver-86382-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=517e756c2cfacf2357a06597b2a21e50ea3895ed05af7321
START /bin/kfmclient openURL "/root/.local/share/jupyter/runtime/nbserver-86382-open.html" text/html
kdeinit4: Shutting down running client.
klauncher: Exiting on signal 15
```

6. VNC 使用

(1) 开启 VNC 服务, 正常情况下输入 `vncserver` 命令会默认添加端口号。如果想要设置自己的端口可以在后面加上:端口号.例如: `vncserver :3` 设置端口号为:3。

```
[inspur03@login02 ~]$ vncserver

You will require a password to access your desktops.

Password: █
```

(2) 设置密码, 第一次输入密码, 在输入确认密码

```
[inspur03@login02 ~]$ vncserver

You will require a password to access your desktops.

Password:
Verify:
```

(3) 选择是否仅设置查看权限的密码如果选择 `y` 会再次输入密码, `n` 会显是创建完成 (注意只有第一次创建的时候会让输入密码, 其余都会采用第一次创建时的密码)

```
Would you like to enter a view-only password (y/n)? n
```

(4) 配置完成截图


```
[inspur03@login02 ~]$ vncserver

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n
A view-only password is not used

New 'login02:2 (inspur03)' desktop is login02:2

Creating default startup script /home/inspur03/.vnc/xstartup
Creating default config /home/inspur03/.vnc/config
Starting applications specified in /home/inspur03/.vnc/xstartup
Log file is /home/inspur03/.vnc/login02:2.log
```

(5) 查看 VNC 端口号输入命令 `vncserver -list` 查看该用户创建的 VNC 端口

```
[inspur03@login02 ~]$ vncserver -list

TigerVNC server sessions:

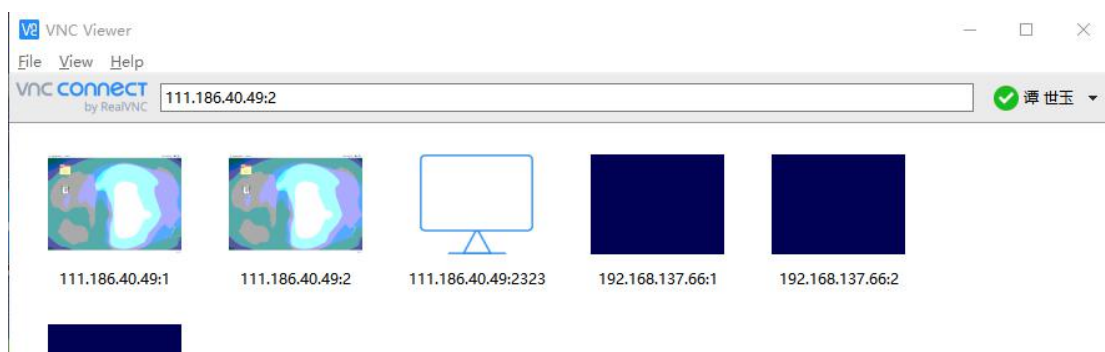
X DISPLAY #      PROCESS ID
:2           103919
```

(6) 其他命令

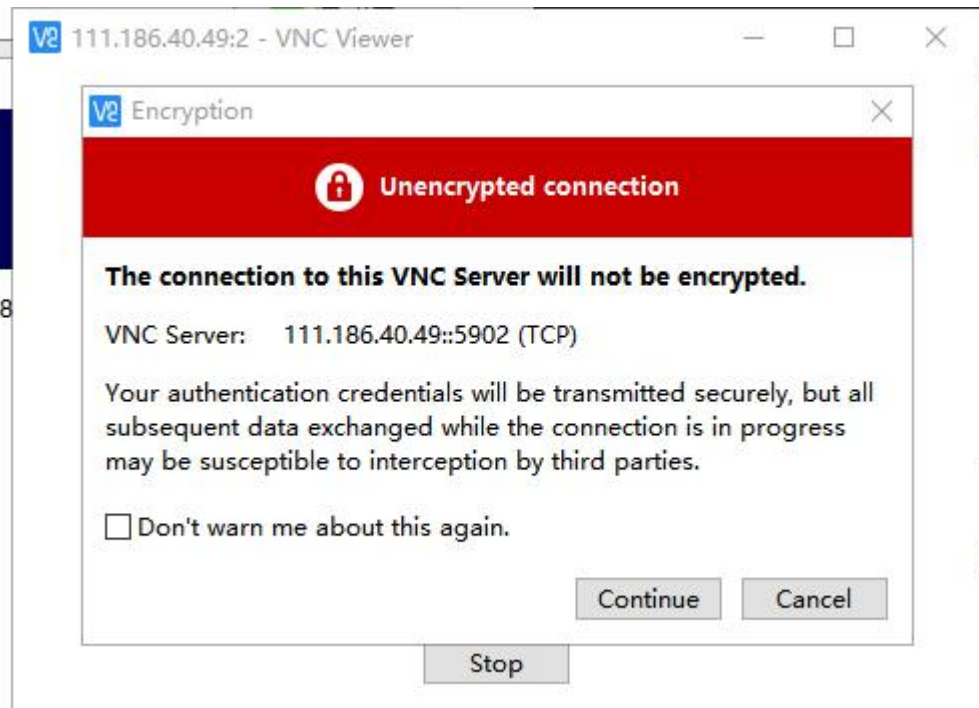
结束 vnc 功能, `vncserver -kill :2 vnc` 更改密码 `vncpasswd`, 按提示输入两次密码即可
每次图形登陆, 最好用 `vncserver -list` 查看是否已经开启 vnc 功能, 避免占用多个 vnc 号

(7) 使用方法

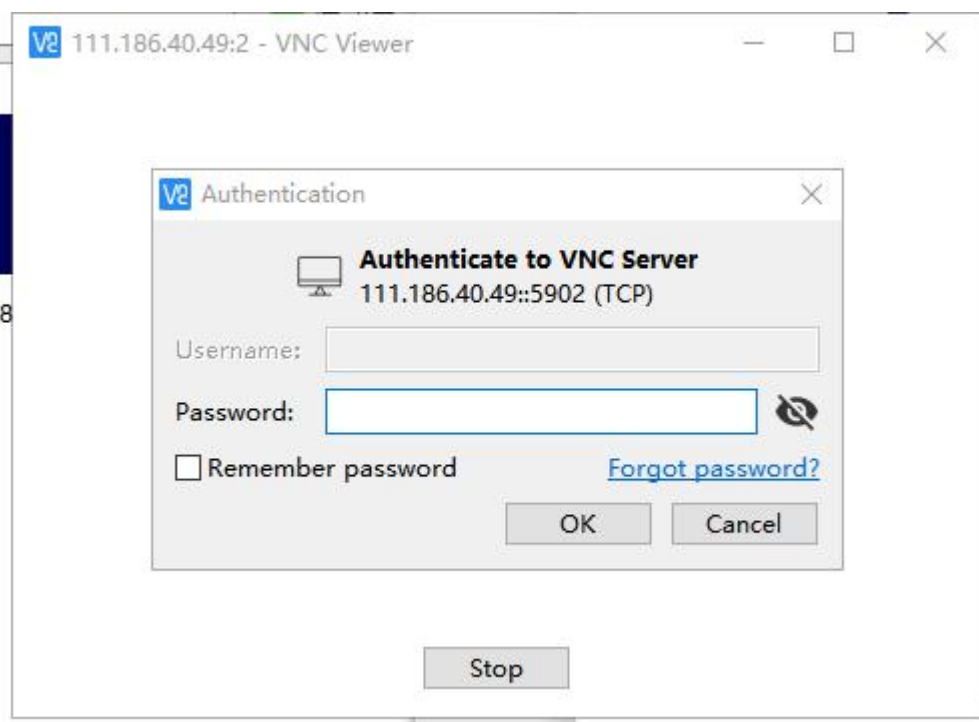
- ① 在 VNC 客户端输入 111.186.40.49:2、
- ② 会有弹出框, 选择 Continue



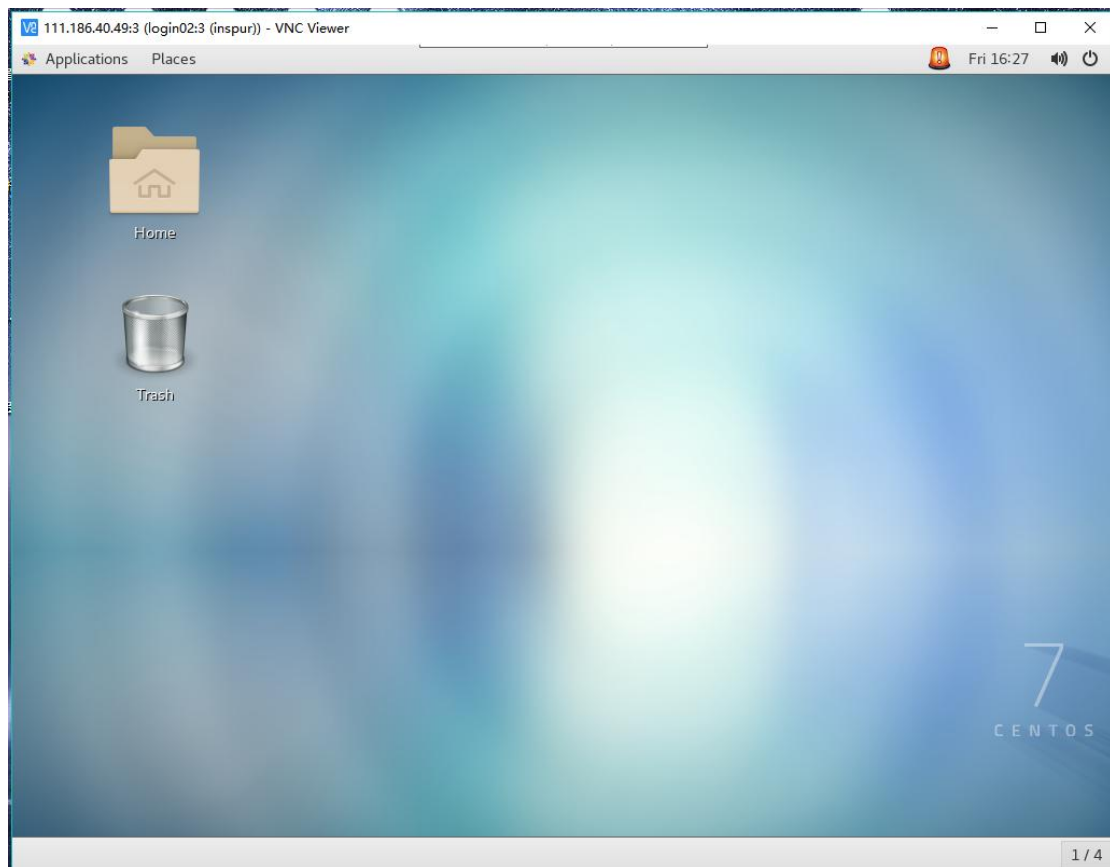
- ③ 会有弹出框, 选择 Continue



④ 再一次弹出框，输入免密



⑤ 登录完成



五. Linux 常用命令说明

1. 帮助命令

- (1) man 获取相关的命令帮助信息

例如:man cp 可以获取关于 cp 命令的使用信息

- (2) Info 获取相关命令的详细使用方法

例如:info info 可以获取如何使用 info 的详细信息

```
[inspur@login02 python]$ info info
file: info.info, Node: Top, Next: Getting Started, Up: (dir)

Info: An Introduction
*****

The GNU Project distributes most of its on-line manuals in the "Info
format", which you read using an "Info reader". You are probably using
an Info reader to read this now.
```

2. 操作命令

- (1) vim 修改文件内容

- (2) `cat` 查看文件内容
- (3) `cp` 将文件拷贝到另一个文件
- (4) `diff` 比较两个文本文件，列出不同之处
- (5) `mv` 文件或目录的移动或更名
- (6) `rm` 删除文件/目录
- (7) `touch` 创建文件
- (8) `mkdir` 创建文件夹
- (9) `tar` 解压文件
- (10) `cd` 切换目录
- (11) `pwd` 显示当前工作目录
- (12) `ssh` (主机名) 登录到其他节