联泰集群
LTHPC
专注提供计算力

山东大学集群用户使用培训

吉管良

LTHPC

2023-9-6

1.集群介绍

本集群包括2套存储系统，1个管理节点，6个CPU节点，8个GPU节点，2个NVLINK节点，节点之间使用100G 的infiniband网络互联。提供1440核心，48张A100（40G）,16张NVLINK A800的计算能力。

NVME存储：

存储容量160T，写速度20G/s,普通用户可用1T，（备注：程序计算优先使用存储/home/用户名，计算后数据移到/storage/用户名）

SATA存储：

存储容量756T,写速度6G/s,普通用户可用不限制，

用户可用分区

联泰集群

分区名	所含节点	详情	QOS权限
cpu	cnode0[1-6]	6个cpu节点， 每台节点96核心，768G内存	cpu96, cpu192, cpu288, cpu384, cpu480, cpu576
a100	gnode0[1-8]	8个a100节点， 每台96核心，2T内存,6张A100	a100g1， a100g2, a100g4, a100g6, a100g12, a100g18, a100g24, a100g30, a100g36, a100g42， a100g48
Nvlink	gnode[09-10]	2个a800节点， 每台64核心，1T内存,8张A800	nvlinkg1, nvlinkg2, nvlinkg4, nvlinkg8, nvlinkg16
debug	gnode08	1个a100节点， 每节点96核心，2T内存，6张A100	a100g1

集群目前有四个分区，分别为cpu, a100, nvlink, debug. 其中debug为调试分区。
用户默认拥有cpu96, a100g1, nvlinkg8, a100g2 等4个QOS,

Qos名称	资源限制	最大提交作业数	单作业最大运行时间
cpu96	每用户可用96cpu核心，	100	不限制
cpu192	每用户可用192cpu核心，	6	不限制
cpu288	每用户可用288cpu核心，	6	不限制
cpu384	每用户可用384cpu核心，	6	不限制
cpu480	每用户可用480cpu核心，	6	不限制
cpu576	每用户可用576cpu核心，	6	不限制

Qos名称	资源限制	最大提交作业数	单作业最大运行时间
a100g1	每用户可用64cpu核心，4张a100 每用户单个作业最大可用16cpu核心，1张a100， 每个用户可同时并发4个任务。	无	不限制
a100g2	每用户可用256cpu核心，16张a100 每用户单个作业最大可用32cpu核心，2张a100， 每个用户可同时并发8个任务。	无	不限制
a100g4	每用户可用64cpu核心，4张a100	6	不限制
a100g6	每用户可用96cpu核心，6张a100	6	不限制
a100g12	每用户可用192cpu核心，12张a100	6	不限制
a100g18	每用户可用288cpu核心，18张a100	6	不限制
a100g24	每用户可用384cpu核心，24个张a100	6	不限制
a100g30	每用户可用480cpu核心，30个张a100	6	不限制
a100g36	每用户可用576cpu核心，36个张a100	6	不限制
a100g42	每用户可用672cpu核心，42个张a100	6	不限制
a100g48	每用户可用768cpu核心，48个张a100	6	不限制

Qos名称	资源限制	最大提交作业数	单作业最大运行时间
nvlinkg1	每用户可用8cpu核心，1个nvlinka800	6	不限制
nvlinkg2	每用户可用16cpu核心，2个nvlinka800	6	不限制
nvlinkg4	每用户可用32cpu核心，4个nvlinka800	6	不限制
nvlinkg8	每用户可用64cpu核心，8个nvlinka800	6	不限制
nvlinkg16	每用户可用128cpu核心，16个nvlinka800	6	不限制

用户说明

- account：账户，一个账户可以含有多个用户。
- user：用户，多个用户可以共享一个账户。

```
###增加用户
sacctmgr  add Cluster cluster
sacctmgr  add  Account name=lthpc Cluster=cluster
sacctmgr  add user test0803 Account=lthpc Cluster=cluster

##所有用户增加
for i in `cat /root/user.txt`;do sacctmgr add user $i Account=lthpc Cluster=cluster;done
Sacctmgr list user
```

用户默认拥有Cpu96,a100g1,nvlinkg8等3个QOS

•GRES：Generic Resource，通用资源。
•TRES：Trackable RESources，可追踪资源。
•QOS：Quality of Service，服务质量，作业优先级。
•association：关联。可利用其实现，如用户的关联不在数据库中，这将阻止用户运行作业。该选项可以阻止用户访问无效账户。
•Partition：队列、分区。用于对计算节点、作业并行规模、作业时长、用户等进行分组管理，以合理分配资源

Cpu队列Qos设置

```
##设置cpu96
sacctmgr add qos cpu96
sacctmgr  modify qos cpu96 set GrpTRESPerUser=cpu=96
sacctmgr modify qos cpu96 set GrpSubmit=6
##设置cpu192
sacctmgr add qos cpu192
sacctmgr  modify qos cpu192 set GrpTRES=cpu=192
sacctmgr modify qos cpu192 set GrpSubmit=6
##设置cpu288
sacctmgr add qos cpu288
sacctmgr  modify qos cpu288 set GrpTRES=cpu=288
sacctmgr modify qos cpu288 set GrpSubmit=6
```

Cpu队列Qos设置

```
##设置cpu384
sacctmgr add qos cpu384
sacctmgr  modify qos cpu384 set GrpTRES=cpu=384
sacctmgr modify qos cpu384 set GrpSubmit=6
##设置cpu480
sacctmgr add qos cpu480
sacctmgr  modify qos cpu480 set GrpTRES=cpu=480
sacctmgr modify qos cpu480 set GrpSubmit=6
##设置cpu576
sacctmgr add qos cpu576
sacctmgr  modify qos cpu576 set GrpTRES=cpu=576
sacctmgr modify qos cpu576 set GrpSubmit=6
```

•**GrpTRES** =：在给定的任意时间，能从关联及其子项或QOS运行作业的TRES总数。如达到该限制，新作业将处于排队状态，仅当资源被从该组释放时才能开始运行。

•**GrpSubmitJobs** =：在给定的任意时间，能从关联及其子项QOS中提交到系统中的作业总数。如达到该限制，提交新作业将被拒绝，仅当之前作业结束时才能提交。

•**MaxTRESPerUser** = 每个用户能同时分配的最大TRES数。

•当给一个QOS赋予限制以用于队列QOS，请务必注意这些限制是在QOS层级，而不是对每个队列层级单独实行的。例如，某个QOS具有 **GrpTRES=cpu=20** 限制，且该QOS被赋予两个独立队列，用户将因该QOS被限制到20颗CPU而不是每个队列允许20颗CPU。

a100队列Qos设置

```
##设置a100g1
sacctmgr add qos a100g1
sacctmgr  modify qos a100g1 set MaxRESPerUser=cpu=64,gres/gpu:a100=4
sacctmgr modify  qos a100g1  set MaxTRESPerJob=cpu=16,gres/gpu:a100=1
##设置a100g2
sacctmgr add qos a100g2
sacctmgr  modify qos a100g2 set MaxTRESPerUser=cpu=64,gres/gpu:a100=4
Sacctmgr modify qos a100g2  set MaxTRESPerJob=cpu=32,gres/gpu:a100=2
##设置a100g4
sacctmgr add qos a100g4
sacctmgr  modify qos a100g4 set GrpTRES=cpu=64,gres/gpu:a100=4
sacctmgr modify qos a100g4 set GrpSubmit=6
##设置a100g6
sacctmgr add qos a100g6
sacctmgr  modify qos a100g6 set GrpTRES=cpu=96,gres/gpu:a100=6
sacctmgr modify qos a100g6 set GrpSubmit=6
```

a100队列Qos设置

```
##设置a100g12
sacctmgr add qos a10012
sacctmgr  modify qos a100g12 set GrpTRES=cpu=192,gres/gpu:a100=12
sacctmgr modify qos a100g12 set GrpSubmit=6
##设置a100g18
sacctmgr add qos a100g18
sacctmgr  modify qos a100g18 set GrpTRES=cpu=288,gres/gpu:a100=18
sacctmgr modify qos a100g18 set GrpSubmit=6
##设置a100g24
sacctmgr add qos a100g24
sacctmgr  modify qos a100g24 set GrpTRES=cpu=384,gres/gpu:a100=24
sacctmgr modify qos a100g24 set GrpSubmit=6
##设置a100g30
sacctmgr add qos a100g30
sacctmgr  modify qos a100g30 set GrpTRES=cpu=480,gres/gpu:a100=30
sacctmgr modify qos a100g30 set GrpSubmit=6
```

a100队列Qos设置

```
##设置a100g36
sacctmgr add qos a100g36
sacctmgr  modify qos a100g36 set GrpTRES=cpu=576,gres/gpu:3g.20gb=36
sacctmgr modify qos a100g36 set GrpSubmit=6
##设置a100g42
sacctmgr add qos a100g42
sacctmgr  modify qos a100g42 set GrpTRES=cpu=672,gres/gpu:3g.20gb=42
sacctmgr modify qos a100g42 set GrpSubmit=6
##设置a100g48
sacctmgr add qos a100g48
sacctmgr  modify qos a100g48 set GrpTRES=cpu=768,gres/gpu:3g.20gb=48
sacctmgr modify qos a100g48 set GrpSubmit=6
```

nvlink队列Qos设置

```
##设置nvlinkg1
sacctmgr add qos nvlinkg1
sacctmgr  modify qos nvlinkg1 set GrpTRES=cpu=8,gres/gpu:a800=1
sacctmgr modify qos nvlinkg1 set GrpSubmit=6
##设置nvlinkg2
sacctmgr add qos nvlinkg2
sacctmgr  modify qos nvlinkg2 set GrpTRES=cpu=16,gres/gpu:a800=2
sacctmgr modify qos nvlinkg2 set GrpSubmit=6
##设置nvlinkg4
sacctmgr add qos nvlinkg4
sacctmgr  modify qos nvlinkg4 set GrpTRES=cpu=32,gres/gpu:a800=4
sacctmgr modify qos nvlinkg4 set GrpSubmit=6
##设置nvlinkg6
sacctmgr add qos nvlinkg6
sacctmgr  modify qos nvlinkg6 set GrpTRES=cpu=64,gres/gpu:a800=8
sacctmgr modify qos nvlinkg6 set GrpSubmit=6
```

nvlink队列Qos设置

```
##设置nvlinkg16
sacctmgr add qos nvlinkg16
sacctmgr  modify qos nvlinkg16 set GrpTRES=cpu=128,gres/gpu:a800=16
sacctmgr modify qos nvlinkg16 set GrpSubmit=6
```

所有用户分配QOS

```
##设置a100g1
sacctmgr modify user test0803 set qos+=a100g1
##设置nvlinkg8
sacctmgr modify user test0803 set qos+=nvlinkg8
```

##设置

```
sacctmgr modify user bob set GrpTRES=cpu=1500,mem=200,gres/gpu=50
```

##不设置

```
sacctmgr modify user bob set GrpTRES=cpu=-1,mem=-1,gres/gpu=-1
```

官网地址：https://slurm.schedmd.com/resource_limits.html

SPECIFIC LIMITS OVER GRES

When a GRES has a type associated with it and a limit is applied over this specific type (e.g. *MaxTRESPerUser=gres/gpu:tesla=1*) if a user requests a generic gres, the type's limit will not be enforced. In this situation an additional lua job submit plugin to check the user request may become useful. For example, if one requests *--gres=gpu:2* having a limit set of *MaxTRESPerUser=gres/gpu:tesla=1*, the limit won't be enforced so it will still be possible to get two teslas.

Having this script and the limit in place will force the users to always specify a gpu with its type, thus enforcing the limits for each specific model.

```
function slurm_job_submit(job_desc, part_list, submit_uid
   if (job_desc.gres ~= nil)
   then
      for g in job_desc.gres:gmatch("[^,]+")
      do
         bad = string.match(g,'^gpu[:]*[0-9]*$')
         if (bad ~= nil)
         then
            slurm.log_info("User specified gpu GRES witho
            slurm.user_msg("You must always specify a typ
            return slurm.ERROR
         end
      end
   end
end
```

显示用户的qos


```
cluster      lthpc     yuting                    1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc    yuxueshi                   1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc zhangjinq+                    1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc    zhangrr                    1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc   zhangtian                   1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc zhangxiao+                    1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc zhangzhen+                    1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc zhaoxiaoyu                    1
                                                                a100g1,cpu96,nvlink+
cluster      lthpc      zhuxu                    1
                                                                a100g1,cpu96,nvlink+

root@master ~]#
```

查看Qos设置

联泰集群
LTHPC

##sacctmgr show qos -pn

```
100g12|0|00:00:00|||cluster|||1.000000|cpu=192,gres/gpu:a100=12||||6|||||||||||
|
100g18|0|00:00:00|||cluster|||1.000000|cpu=288,gres/gpu:a100=18||||6|||||||||||
|
100g24|0|00:00:00|||cluster|||1.000000|cpu=384,gres/gpu:a100=24||||6|||||||||||
|
100g30|0|00:00:00|||cluster|||1.000000|cpu=480,gres/gpu:a100=30||||6|||||||||||
|
100g36|0|00:00:00|||cluster|||1.000000|cpu=576,gres/gpu:a100=36||||6|||||||||||
|
100g42|0|00:00:00|||cluster|||1.000000|cpu=672,gres/gpu:a100=42||||6|||||||||||
|
100g48|0|00:00:00|||cluster|||1.000000|cpu=768,gres/gpu:a100=48||||6|||||||||||
|
vlinkg2|0|00:00:00|||cluster|||1.000000|cpu=16,gres/gpu:a800=2||||6|||||||||||

vlinkg4|0|00:00:00|||cluster|||1.000000|cpu=32,gres/gpu:a800=4||||6|||||||||||

vlinkg8|0|00:00:00|||cluster|||1.000000|cpu=64,gres/gpu:a800=8||||6|||||||||||

vlinkg16|0|00:00:00|||cluster|||1.000000|cpu=128,gres/gpu:a800=16||||6|||||||||
|||
putest1|0|00:00:00|||cluster|||1.000000||||||||||||||||||
```

2.修改用户密码

命令修改密码

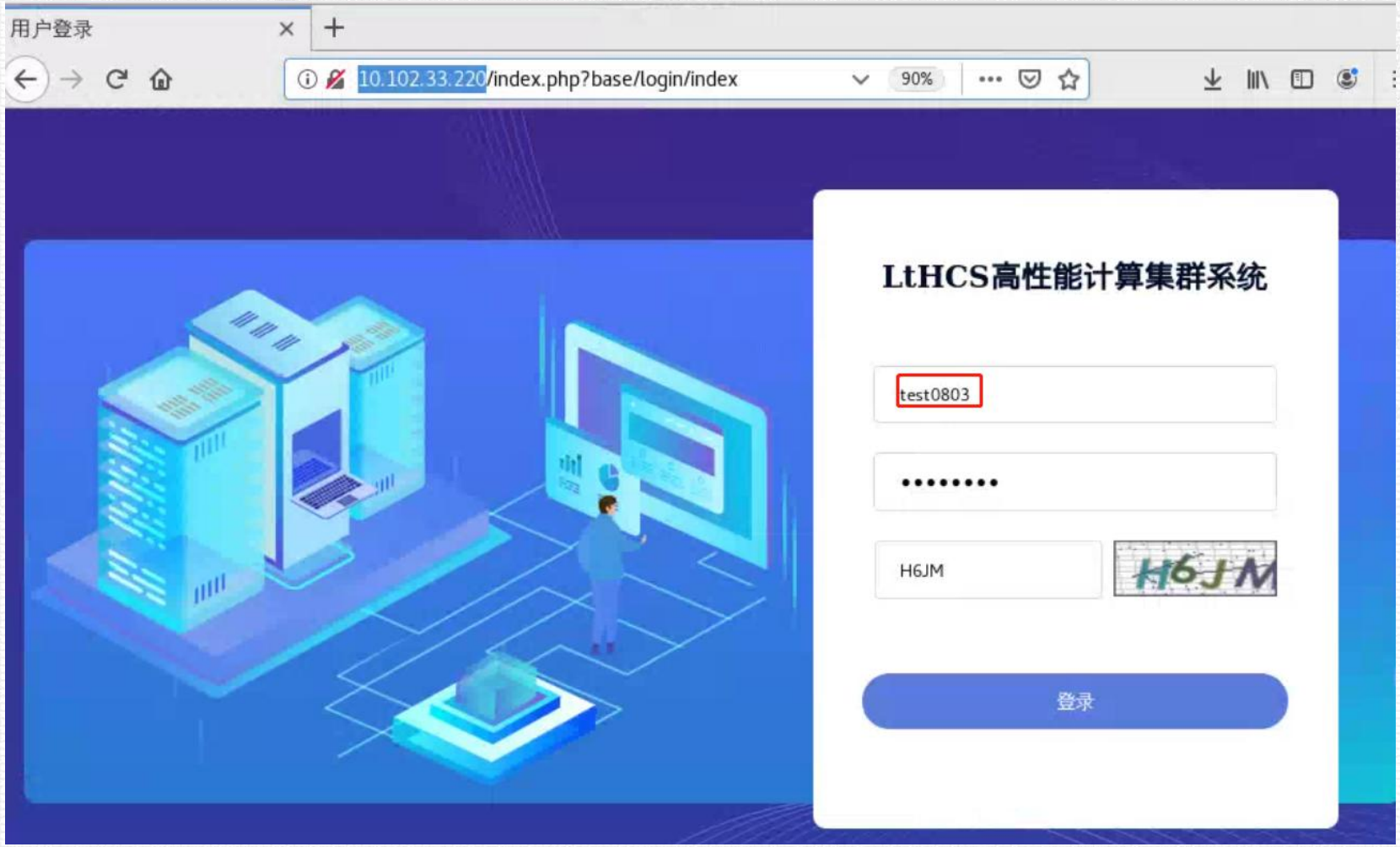
ldapuserpasswd.sh 新密码

```
(base) [test0803@master ~]$ ldapuserpasswd.sh  bbbb1111
Only root can do that.
test0803 password change ok!
```

界面修改密码

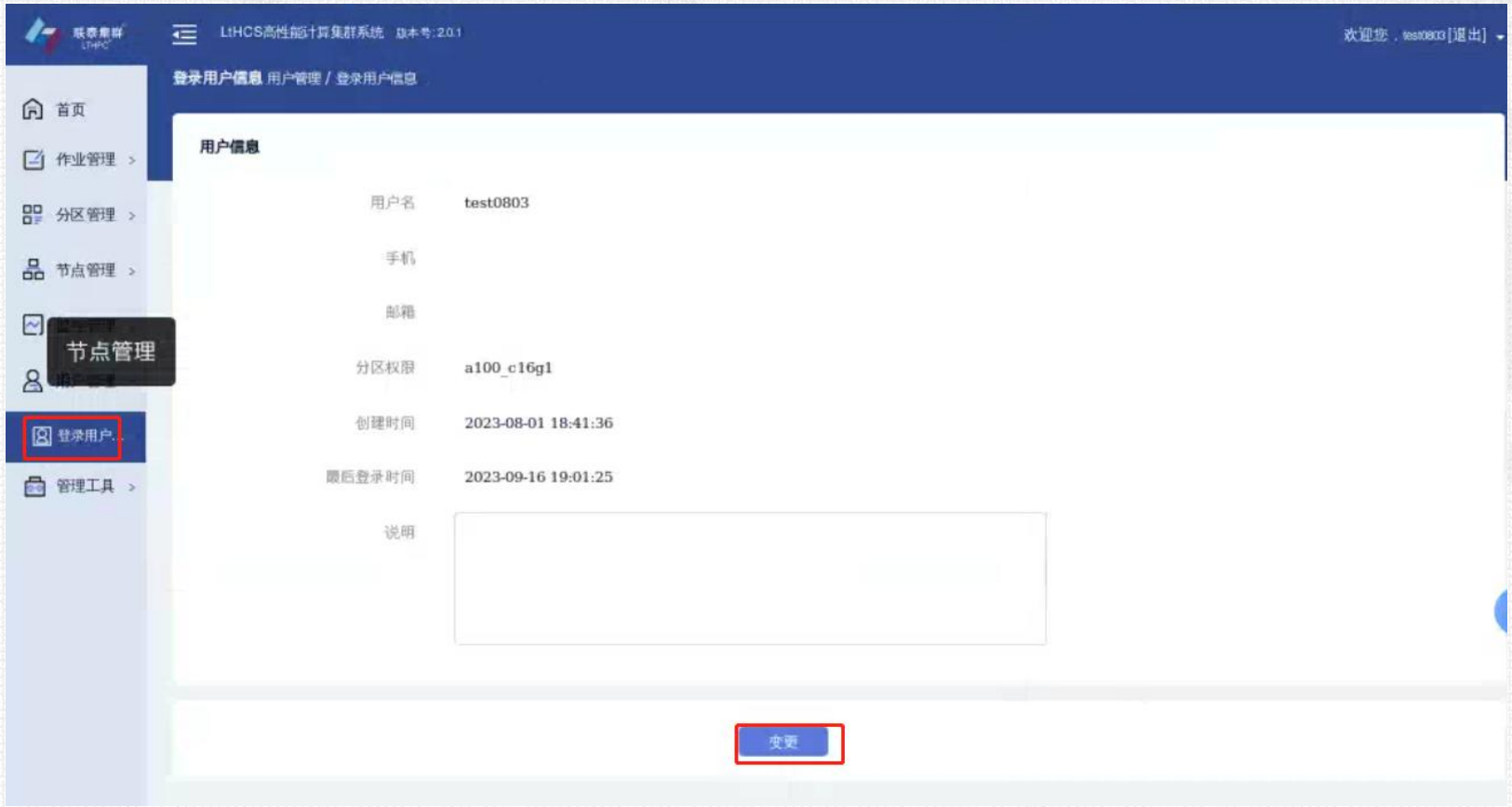
联泰集群
LTHPC

打开google游览器，输入:http://10.102.33.220,（如图）输入用户名和密码：



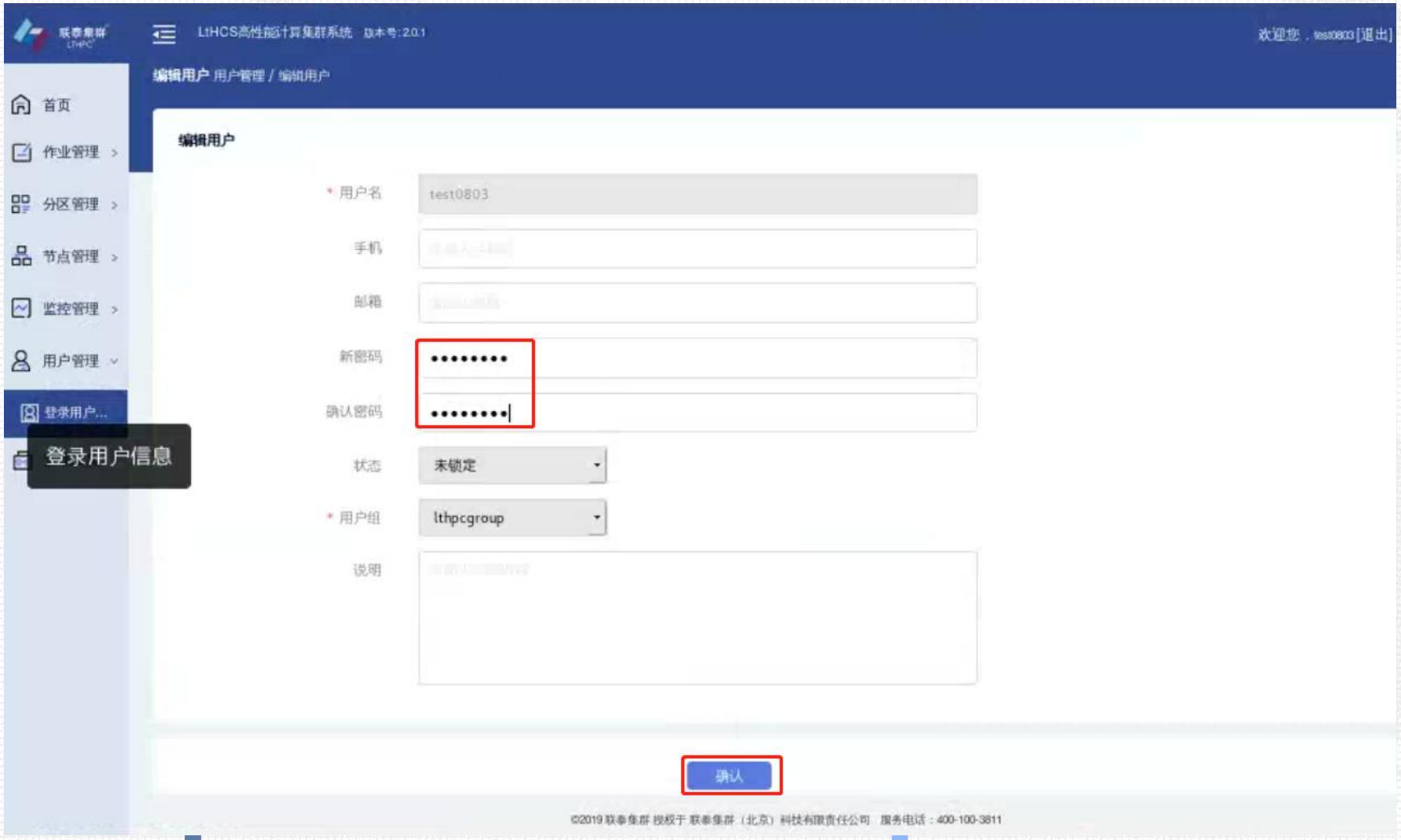
用户管理

选择登录用户，然后选择变更



输入新密码

新密码输入密码，然后选择“确认”，提示编辑成功。



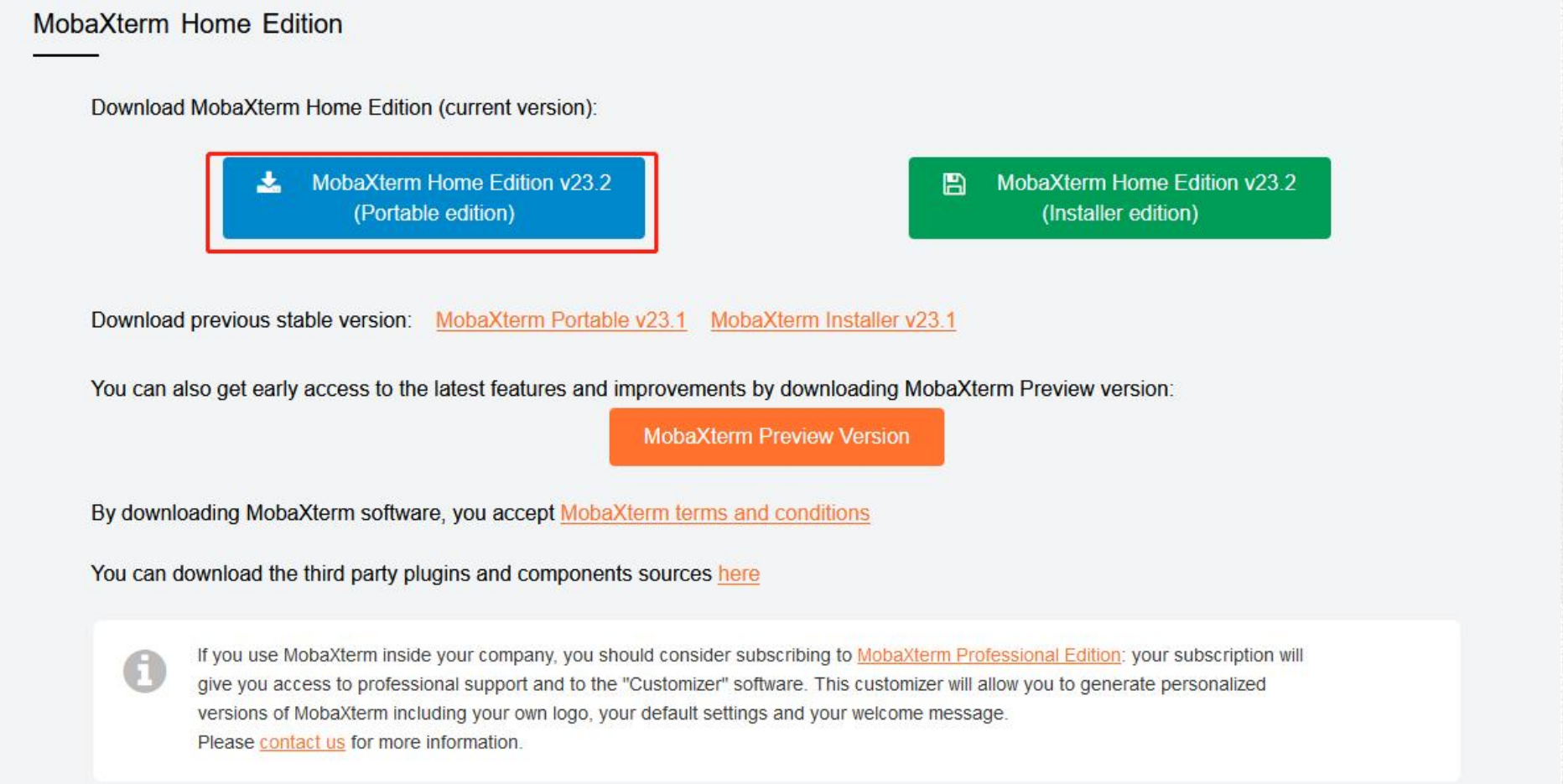
3.Ssh登录

外网地址及端口

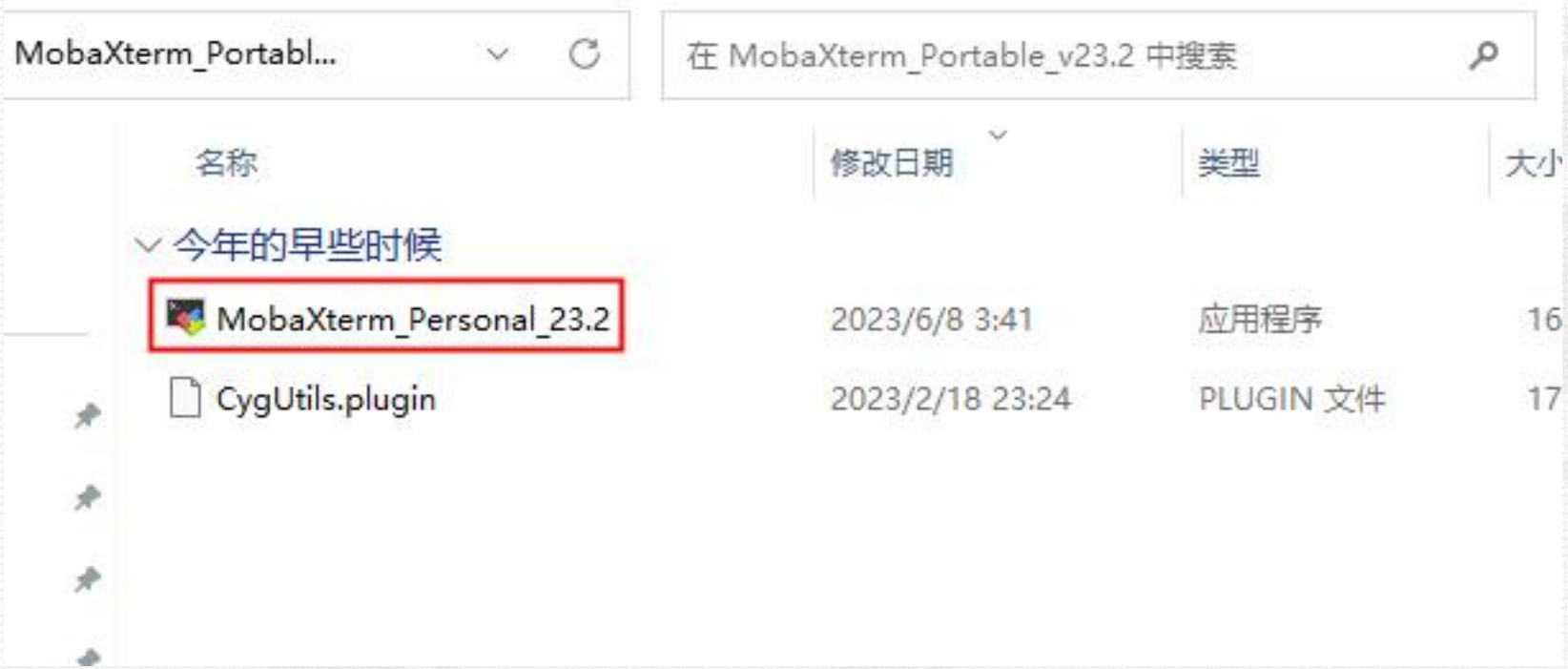
外网地址：10.102.33.220 ssh端口：10022

下载ssh工具--mobaxterm

下载地址：
https://mobaxterm.mobatek.net/download-home-edition.html
选左面便捷版

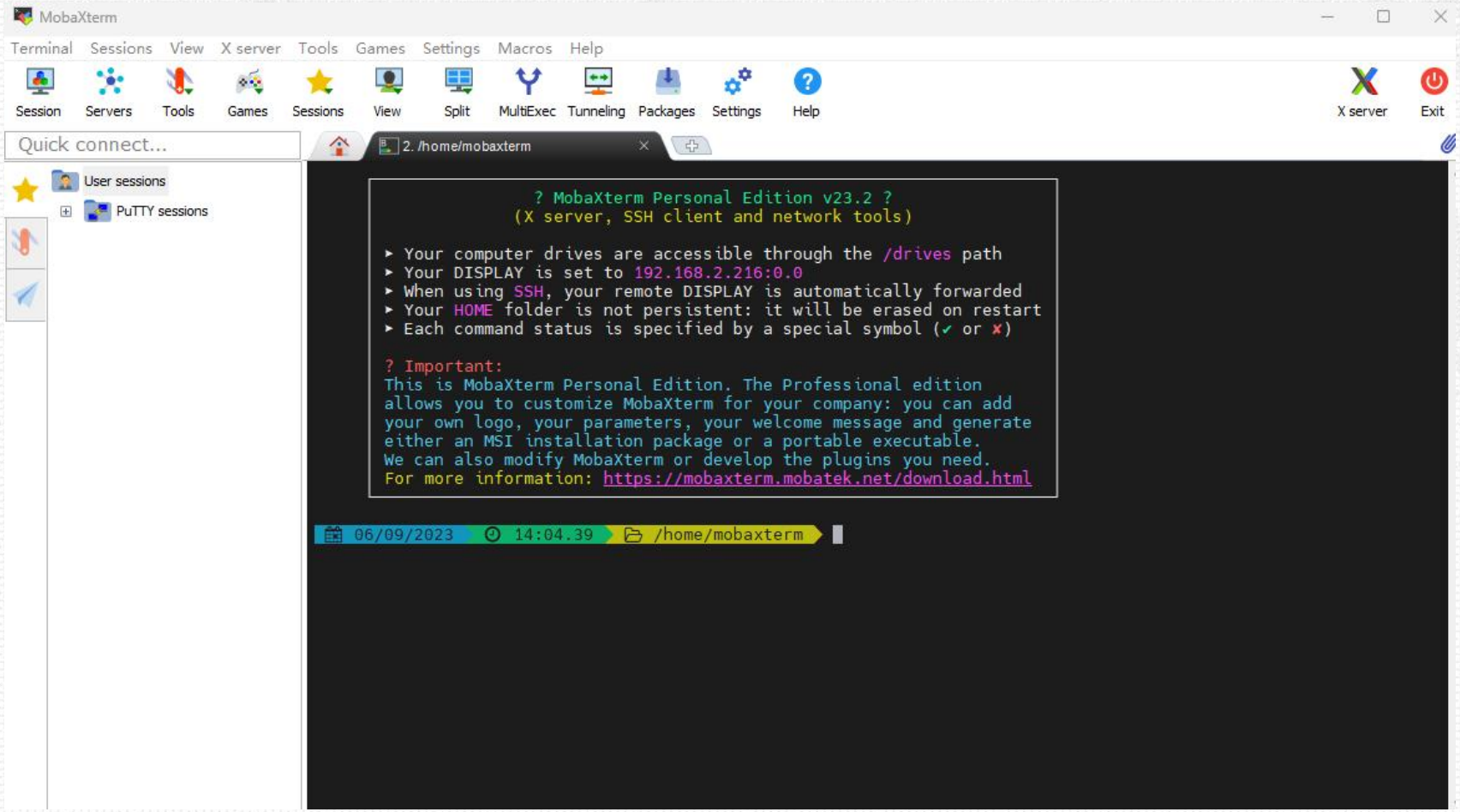


下载后的文件解压后，得到exe文件

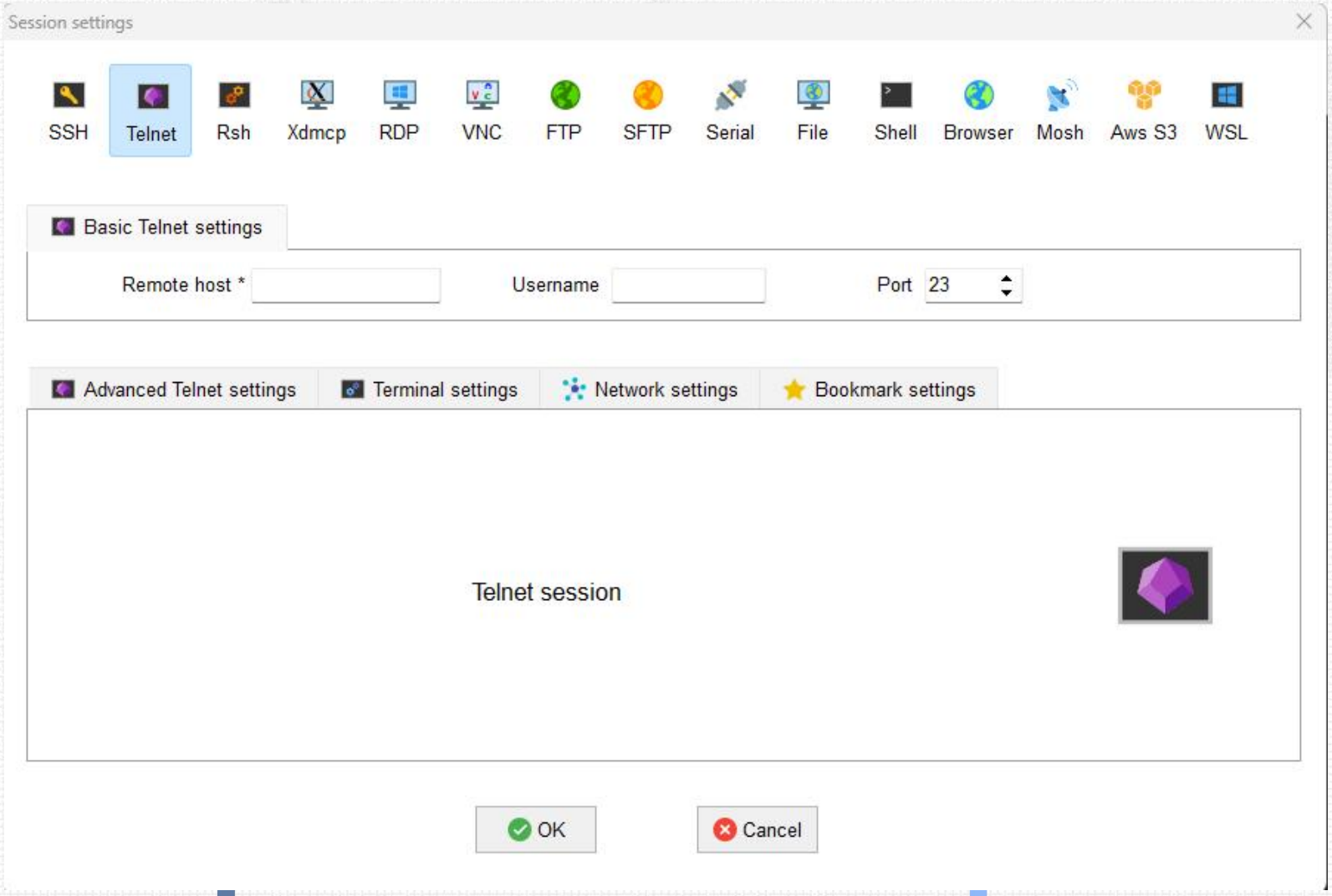


Mobaxterm使用

双击exe文件,打开如图:

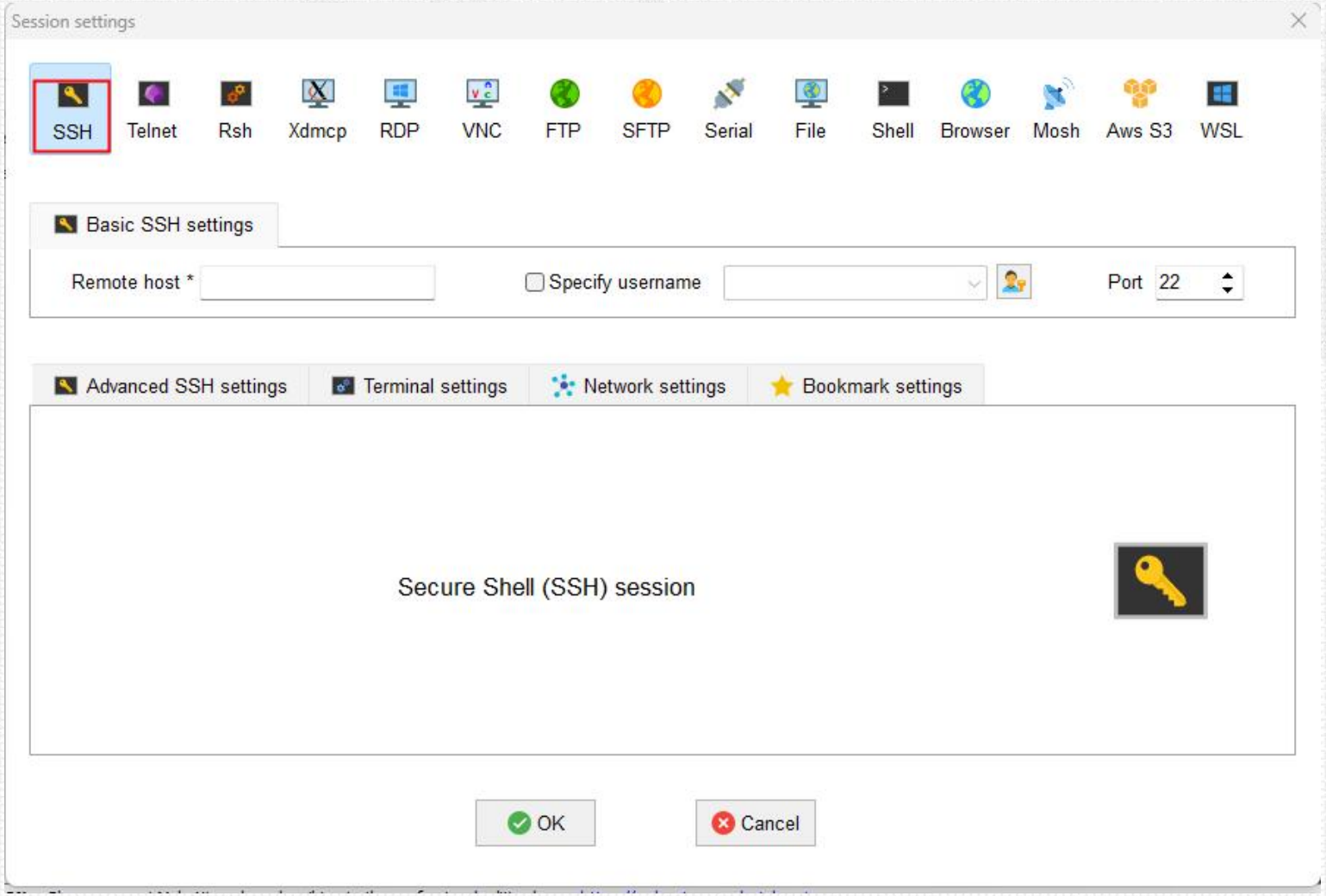


双击Session，打开如图：



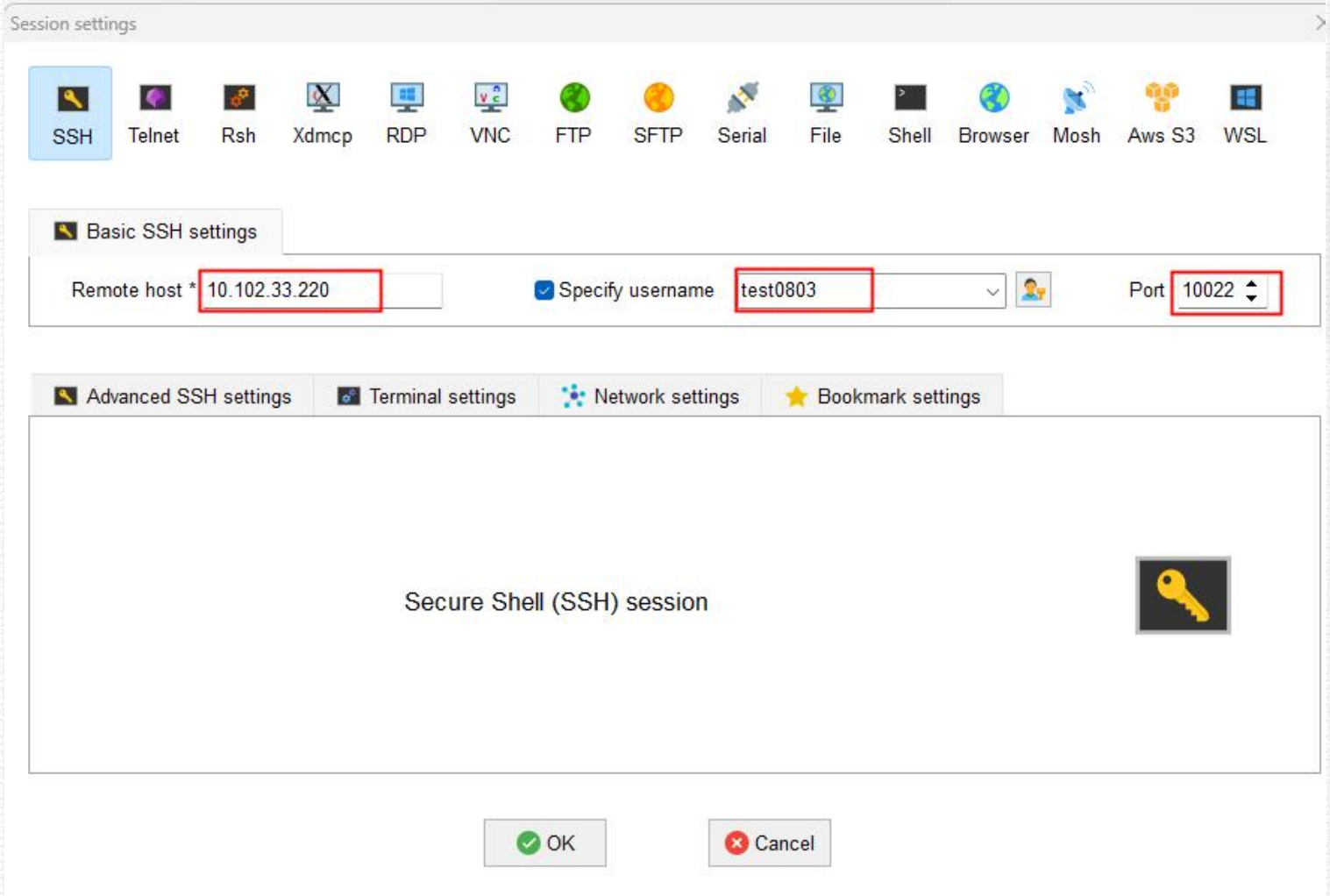
Mobaxterm使用

双击ssh，打开如图：

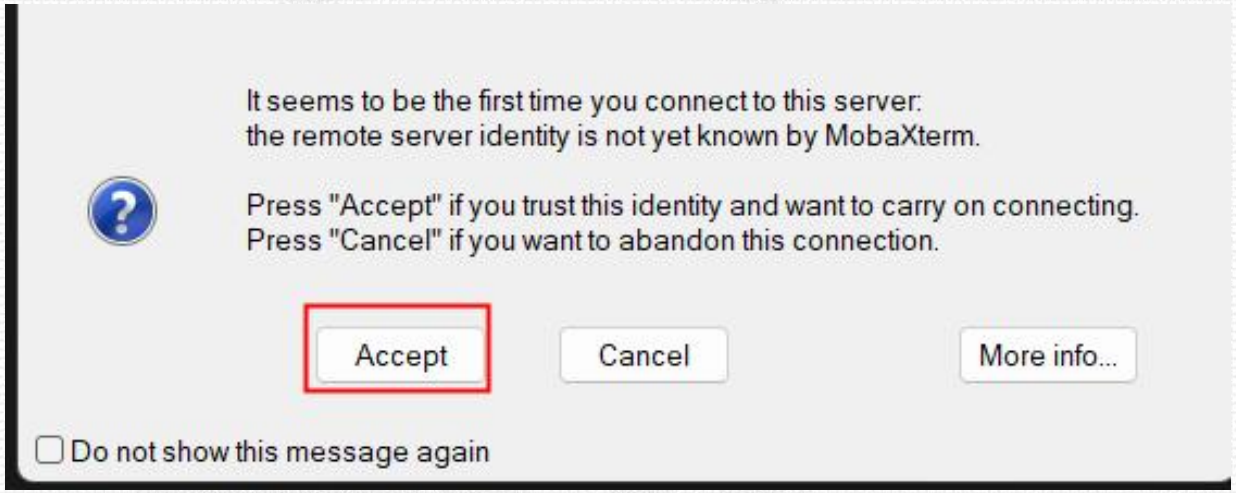


Mobaxterm使用

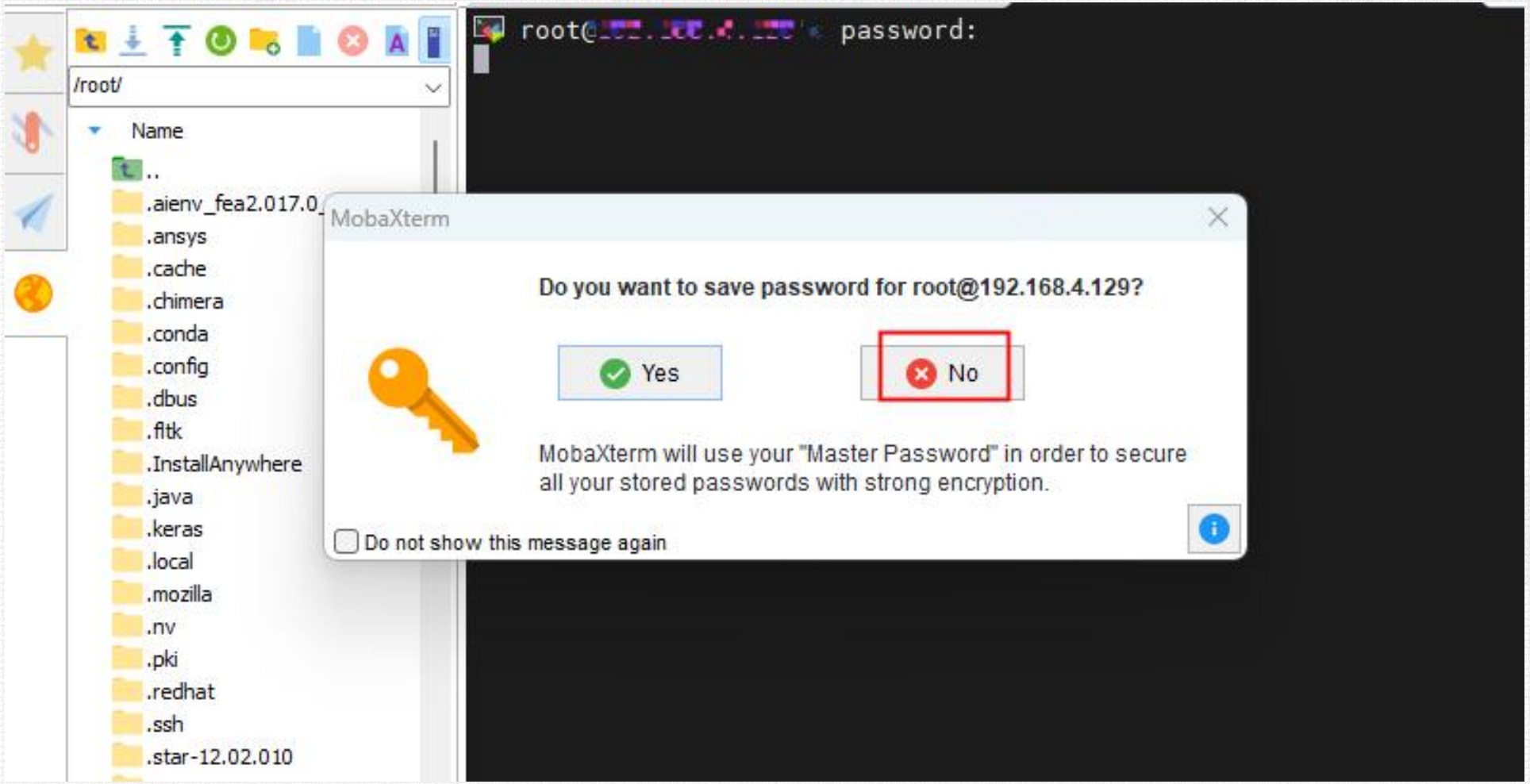
输入地址： 10.102.33.220 用户名：test0803 端口：10022



选择“ok”，第一次登录需要选择“Accept”

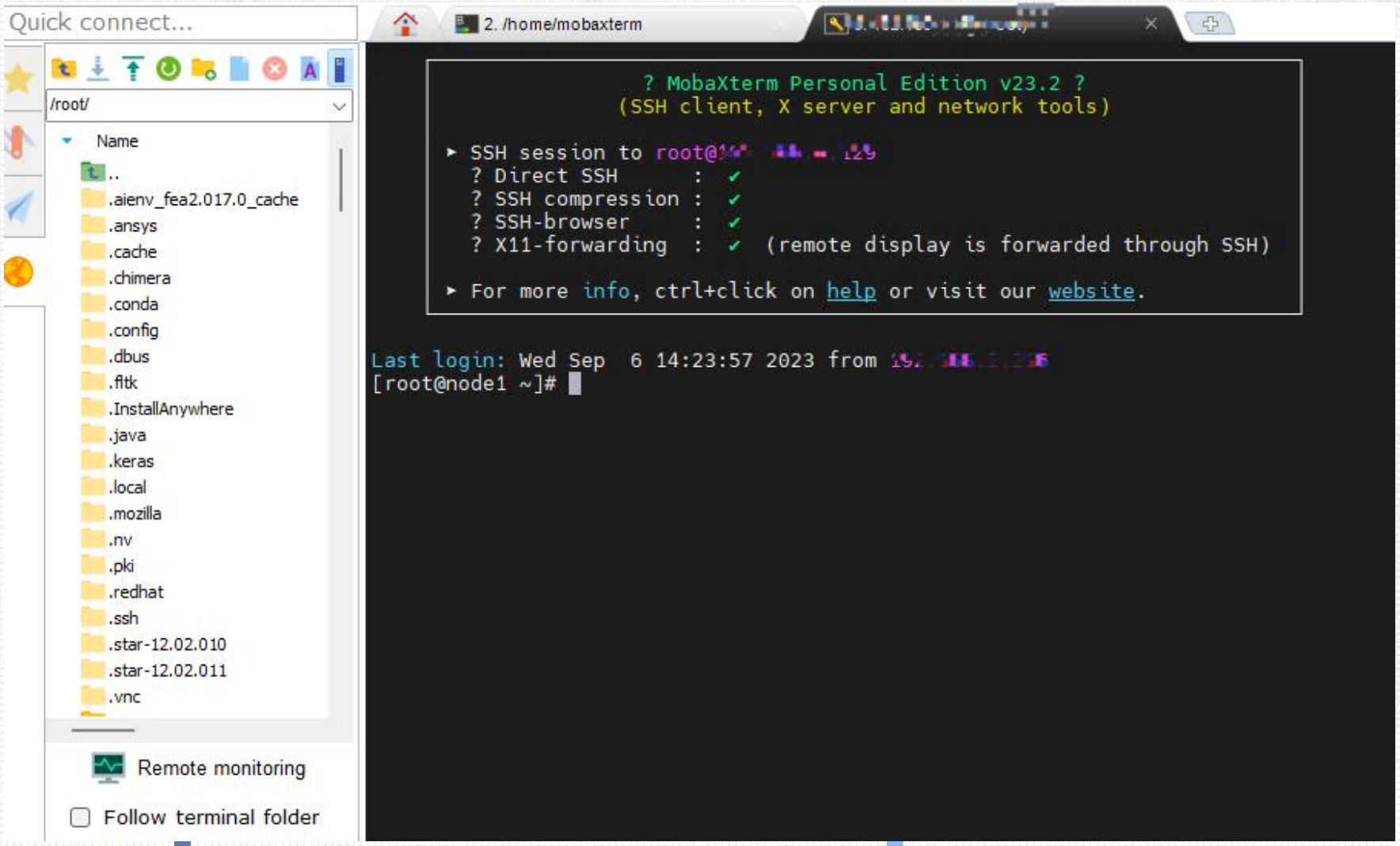


输入密码，可以选择“no”


root@ password:
/root/
Name
..
.aienv_fea2.017.0
.ansys
.cache
.chimera
.conda
.config
.dbus
.fltk
.InstallAnywhere
.java
.keras
.local
.mozilla
.nv
.pki
.redhat
.ssh
.star-12.02.010
MobaXterm
Do you want to save password for root@192.168.4.129?
Yes
No
MobaXterm will use your "Master Password" in order to secure all your stored passwords with strong encryption.
Do not show this message again

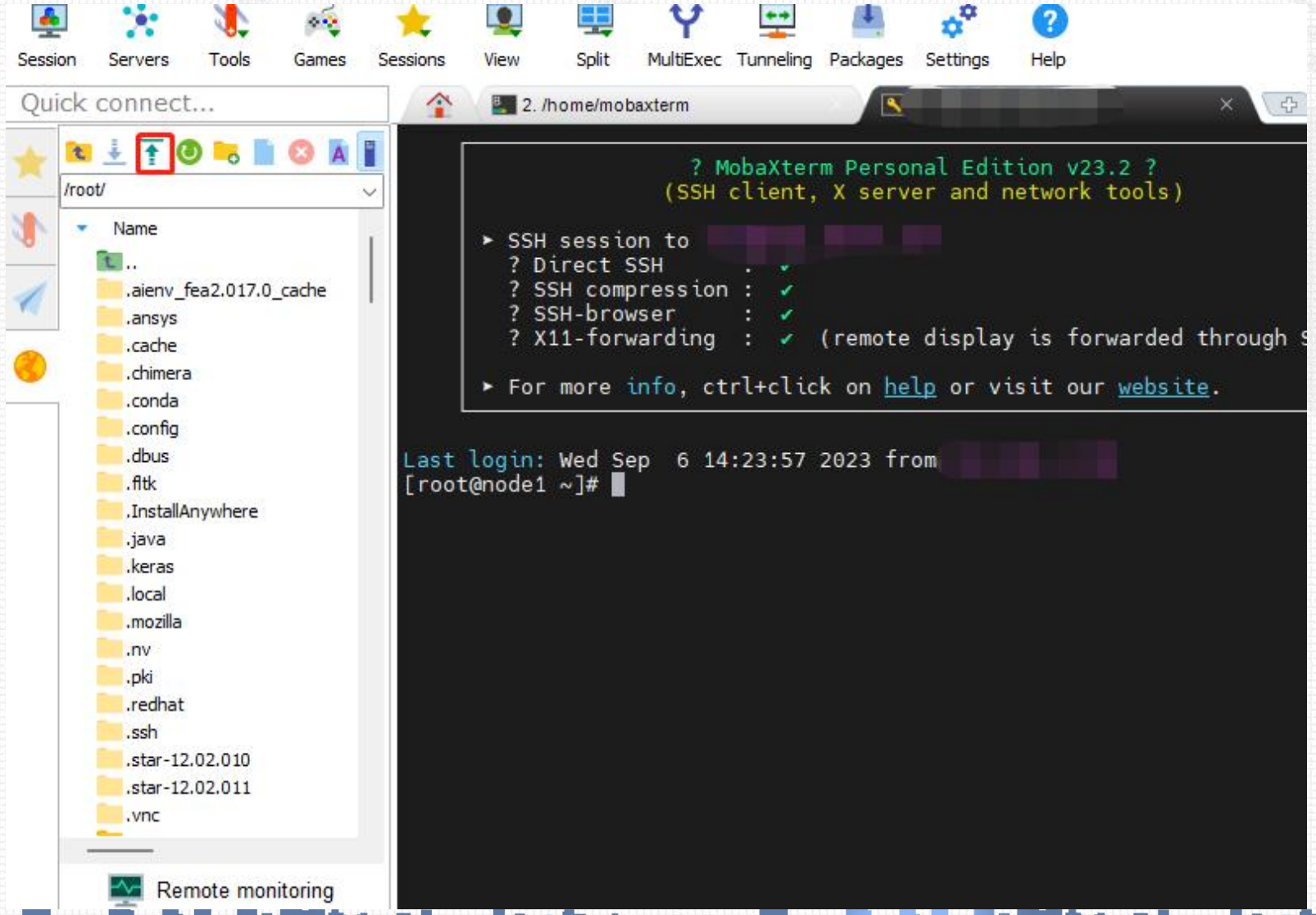
Mobaxterm使用

登录成功如图：



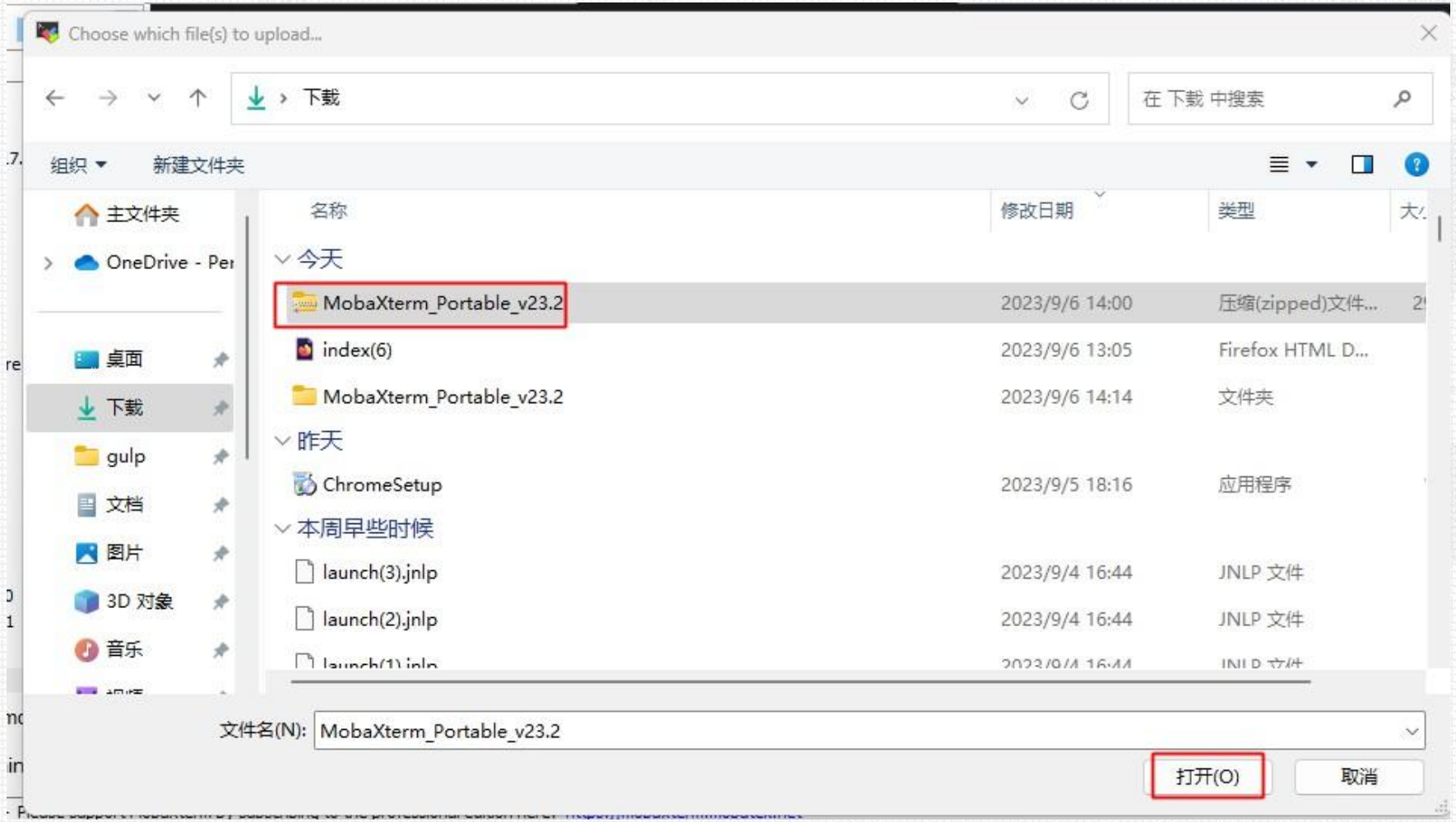
Mobaxterm上传文件

选择向上箭头，如图：



Mobaxterm上传文件

选择“上传文件”，选择“打开”，如图：



4.已安装软件

已安装软件

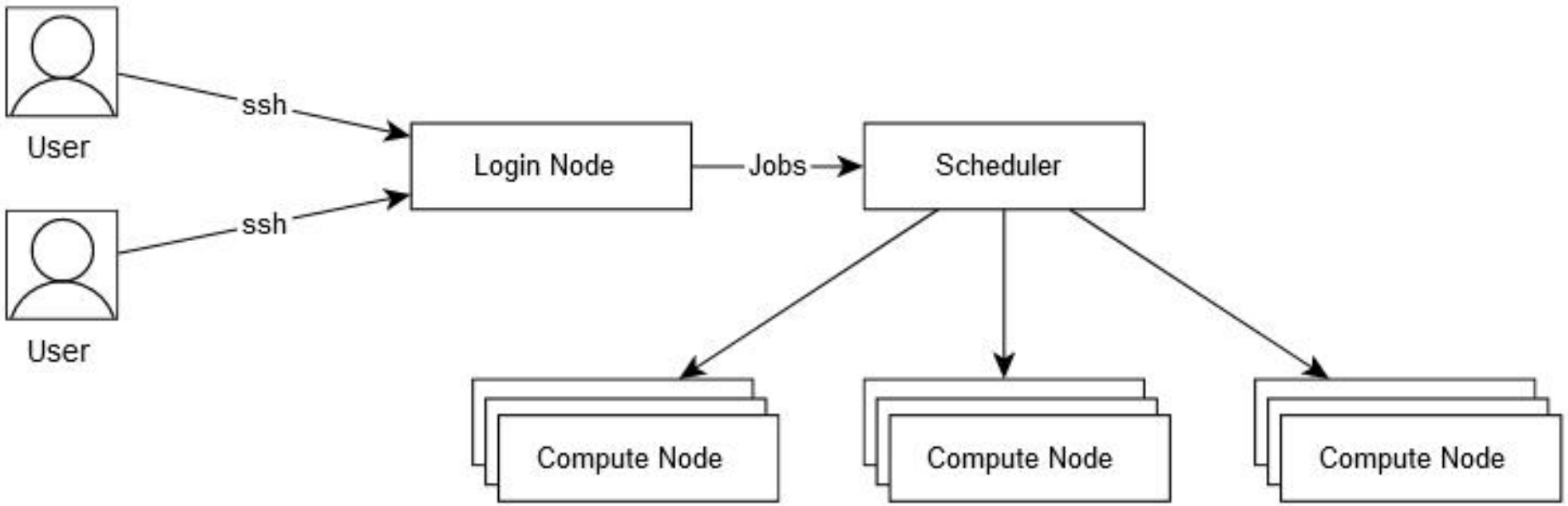
联泰集群 LTHPC

软件安装目录：/opt
已安装软件：ls /opt

```
anaconda3_2020      hmmer-3.3.2          pycharm-2020.3.3
Anaconda3-2023.03   ImageJ               pycharm-2022.1.3
ansys_inc           IMOD                 pycharm-community-2020.3.3
bamtools            imod-4.10            pymol
beagle-5.2          imod_4.11.24         python-2.7.18
blast-2.11.0        imod-4.11.3          python-3.7.10
blat35              impute_v2.3.2        python3.9.9
boost_1_74_0        intel                qe-6.6
bowtie2-2.4.2       IPMIView_2.21.0      R-4.0.3
bsoft               isl-0.15             relion3
bwa-0.7.17          jdk1.8.0_11          rh
ccpem-1.6.0         knem-1.1.4.90mlnx1   sage
CD-adapco           lammps               sage-8.7
chimerax-1.3        libint-1.1.5         sage-9.1
cmake3.18.4         libpng               sage-9.2
cnvator-0.3         libxc-5.1.2          sage-9.4
comsol55            libxsmm              sage-9.8
containerd          lingo18              sambamba-0.8
COPASI-4.29         maple2015            samtools-0-1.19
copasi-4.40.278     MATLAB               samtools-1.11
cufflinks-2.2.1     mellanox             samtools-1.3.1
Eigen               miniconda3           shapeit
emacs               minimap2             slurm-lthpc_2
EMAN2               modeller9.25         snpEff
eman2.91            mpc-1.0.3            STAR-2.7.8a
FastQC              mpfr-3.1.4           tiff
fftw3               nwchem-6.8.1         tinker-8.8.3
fftw3310            OpenBLAS-0.3.13      todesk
gatk-4.1.9.0        opencMISS            tophat
gcc-8.4.0           openmpi4             UCSF
gcc-9.3.0           openmpi401           VarScan-2.3.9
git237              openmpi415           velvet_1.2.10
gmap-2021           openmpi415_intel     xerces-c-3.2.3
gmapdb              openmpi.tar.gz       xtal
```

5.提交作业流程

提交作业流程


User
ssh
User
ssh
Login Node
Jobs
Scheduler
Compute Node
Compute Node
Compute Node

提交作业流程

运行作业的方式有两种：
一种是将计算过程写成脚本，通过sbatch指令提交到计算节点执行；
另一种是通过salloc申请到计算节点，再ssh连接到计算节点进行计算；
通过sinfo查看计算节点空闲状态；
可以通过squeue查看已经提交作业的排队情况；
通过scontrol show job 和sacct查询作业的相关信息；
通过scancel取消已经提交的作业。

6.3种队列脚本介绍

1.CPU脚本

```
vim cpu.sbatch
###脚本部分
#!/bin/bash
#SBATCH  -J test                                  ##作业名test
#SBATCH  --cpus-per-task=1                        ##每个任务需要的cpu核心数，默认为1
#SBATCH  --ntasks-per-node=96                     ##每节点任务数96
#SBATCH  -N 1                                     ##指定节点数为1
#SBATCH  -n 96                                    ##指定总任务数为96
#SBATCH  --qos=cpu96                              ##指定qos为cpu96
#SBATCH  -p cpu                                   ##指定队列名cpu

###程序部分
export
LD_LIBRARY_PATH=/opt/intel/compilers_amd_libraries_2019.4.243/linux/compiler/lib:/opt
/intel/compilers_and_libraries_2019.4.243/linux/compiler/lib/intel64_lin:$LD_LIBRARY_
PATH
MPIRUN=“/opt/openmpi415_intel/bin/mpirun”
cd $SLURM_SUBMIT_DIR                              ##SLURM_SUBMIT_DIR是脚本目录
$MPIRUN -np $SLURM_NPROCS ./xhpl                  ##SLURM_NPROCS是使用的任务数
```

```
vim a100.sbatch
###脚本部分
#!/bin/bash
#SBATCH  -J  yolov5s_1                    ##作业名yolov5s_1
#SBATCH  --gres=gpu:a100:1                ##指定使用gpu资源,a100数量1
#SBATCH  -N 1                             ##指定使用1个节点数
#SBATCH  -n 1                             ##指定总任务数1
#SBATCH  -o test.out                      ##指定输出文件
#SBATCH  -e test.error                    ##指定错误输出文件
#SBATCH  --qos=a100g1                     ##指定qos为a100g1
#SBATCH  -p a100                          ##指定队列a100
###程序部分
cd $SLURM_SUBMIT_DIR
/home/test0803/anaconda3/envs/pytorch2.0/bin/python   train.py  --batch 64  -
-data coco.yaml   --weights  yolov5s.pt --device 0
```

```
vim nvlink.sbatch
###脚本部分
#!/bin/bash
#SBATCH   -J   yolov5s_2                    ##作业名yolov5s_2
#SBATCH  --gres=gpu:a800:2                  ##指定使用gpu,卡名a800,数量为2
#SBATCH  -N 1                               ##指定使用1个节点数
#SBATCH  -n 2                               ##指定任务数2
#SBATCH  -o test.out                        ##指定输出文件
#SBATCH  -e test.error                      ##指定错误输出文件
#SBATCH  --qos=nvlinkg8                     ##指定qos为nvlinkg8
#SBATCH  -p nvlink                          ##指定队列nvlink
###程序部分
Cd $SLURM_SUBMIT_DIR
/home/test0803/anaconda3/envs/pytorch2.0/bin/python   train.py  --batch 64  -
-data coco.yaml   --weights  yolov5s.pt  -device 0，1
```

-n, --ntasks=<number> # sbatch并不会执行任务，当需要申请相应的资源来运行脚本，默认情况下一个任务一个核心，--cpus-per-task参数可以修改该默认值；

-c, --cpus-per-task=<ncpus> # 单进程CPU核数，默认是1.运行OpenMP等多线程程序时需设置，普通MPI程序不需要。

--ntasks-per-node=<ntasks> # 每个节点的任务数，--ntasks参数的优先级高于该参数，如果使用--ntasks这个参数，那么将会变为每个节点最多运行的任务数；

-N,--nodes=<number> #所需节点数

--gpus-per-task（默认没有设置）

GPUs required per task. Requires the job to specify a task count.

```
--gpus-per-task=[type:]<number>
       Specify the number of GPUs required for the job on each task to be spawned in the job's
       resource allocation. An optional GPU type specification can be supplied. For example
       "--gpus-per-task=volta:1". Multiple options can be requested in a comma separated list, for
       example: "--gpus-per-task=volta:3,kepler:1". See also the --gpus, --gpus-per-socket and
       --gpus-per-node options. This option requires an explicit task count, e.g. -n, --ntasks or
       "--gpus=X --gpus-per-task=Y" rather than an ambiguous range of nodes with -N, --nodes. This
       option will implicitly set --gpu-bind=per_task:<gpus_per_task>, but that can be overridden with
       an explicit --gpu-bind specification.
```

--gpus-per-task（默认不支持。已配置为（select/cons_tres））

Using the Consumable Trackable Resource Plugin: **select/cons_tres**

- The Consumable Trackable Resources (**cons_tres**) plugin provides all the same functionality provided by the Consumable Resources (**cons_res**) plugin. It also includes additional functionality specifically related to GPUs.
- Additional parameters available for the **cons_tres** plugin:
  - **DefCpuPerGPU**: Default number of CPUs allocated per GPU.
  - **DefMemPerGPU**: Default amount of memory allocated per GPU.
- Additional job submit options available for the **cons_tres** plugin:
  - **--cpus-per-gpu=**: Number of CPUs for every GPU.
  - **--gpus=**: Count of GPUs for entire job allocation.
  - **--gpu-bind=**: Bind task to specific GPU(s).
  - **--gpu-freq=**: Request specific GPU/memory frequencies.
  - **--gpus-per-node=**: Number of GPUs per node.
  - **--gpus-per-socket=**: Number of GPUs per socket.
  - **--gpus-per-task=**: Number of GPUs per task.
  - **--mem-per-gpu=**: Amount of memory for each GPU.

--gres

Generic resources required per node。

```
--gres=<list>
       Specifies a comma-delimited list of generic consumable resources.  The format of each entry  on
       the  list is "name[[:type]:count]".  The name is that of the consumable resource.  The count is
       the number of those resources with a default value of 1.  The count can have a suffix of "k" or
       "K"  (multiple  of  1024), "m" or "M" (multiple of 1024 x 1024), "g" or "G" (multiple of 1024 x
       1024 x 1024), "t" or "T" (multiple of 1024 x 1024 x 1024 x 1024), "p" or "P" (multiple of  1024
       x  1024  x  1024  x 1024 x 1024).  The specified resources will be allocated to the job on each
       node.  The available generic consumable resources is configurable by the system  administrator.
       A  list  of available generic consumable resources will be printed and the command will exit if
       the option argument is "help".  Examples of use include "--gres=gpu:2",  "--gres=gpu:kepler:2",
       and "--gres=help".
```

7.提交作业

sbatch test.sbatch，返回作业号，如图：

```
(base) [test0803@master test]$ sbatch test.sbatch
Submitted batch job 305
(base) [test0803@master test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               305       cpu     test test0803  R       0:01      1 cnode01
(base) [test0803@master test]$
```

8.查询作业

查询作业--squeue

squeue

```
(base) [test0803@master test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               305       cpu     test test0803  R       0:01      1 cnode01
(base) [test0803@master test]$
```

第一列:JOBID是作业号，作业号是唯一的。
第二列:PARTITION是作业运行使用的队列名。
第三列:NAME是作业名。
第四列:USER是用户名。
第五列:ST是作业状态，R表示正常运行，PD表示在排队，CG表示正在退出，S是管理员暂时挂起。
第六列:TIME是作业运行时间。
第七列:NODES是作业使用的节点数。
第八列:NODELIST(REASON)
对于运行作业（R状态）显示作业使用的节点列表；对于排队作业（PD状态），显示排队的原因。

节点原因

联泰集群
LTHPC

NODELIST(REASON):	
BeginTime:	未到用户所指定的任务开始时间
Dependency:	该作业所依赖的作业尚未完成
InvalidAccount:	用户的 SLURM 账号无效
InvalidQOS :	用户指定的 QoS 无效
ParitionTimeLimit:	用户申请的时间超过该分区时间上限
QOSMaxCpuPerUserLimit:	超过当前 QoS 用户最大 CPU 限制
QOSMaxGRESPerUser :	超过当前 QoS 用户最大 GRES(GPU) 限制
Priority:	存在一个或多个更高优先级的任务，该任务需要等待
ReqNodeNotAvail:	所申请的部分节点不可用
Resources :	暂无闲置资源，该任务需等待其他任务完成

squeue常用命令

```
squeu -j <job_id_list> # 显示指定作业号的作业信息，
squeue  -n <name_list> # 显示指定节点上的作业信息，
squeue  -t <state_list> # 显示指定状态的作业信息，
squeue -u <user_list> # 显示指定用户的作业信息，
squeue  -w <hostlist> # 显示指定节点上运行的作业，
```

查询作业scontrol show job

scontrol show job 305 ##查询作业的详细信息（提交时间，脚本路径等）

```
(base) [test0803@master test]$ scontrol show jobs 305
JobId=305 JobName=test
   UserId=test0803(1195) GroupId=test0803(1195) MCS_label=N/A
   Priority=4294901757 Nice=0 Account=lthpc QOS=cpu96
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:02:04 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2023-09-09T21:25:21 EligibleTime=2023-09-09T21:25:21
   AccrueTime=2023-09-09T21:25:21
   StartTime=2023-09-09T21:25:22 EndTime=Unknown Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-09-09T21:25:22 Scheduler=Main
   Partition=cpu AllocNode:Sid=master:87495
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=cnode01
   BatchHost=cnode01
   NumNodes=1 NumCPUs=96 NumTasks=96 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=96,node=1,billing=96
   Socks/Node=* NtasksPerN:B:S:C=96:0:*:* CoreSpec=*
   MinCPUsNode=96 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/test0803/test/test.sbatch
   WorkDir=/home/test0803/test
   StdErr=/home/test0803/test/slurm-305.out
   StdIn=/dev/null
   StdOut=/home/test0803/test/slurm-305.out
   Power=
```

9.删除作业

删除作业

```
scancel 56          #取消作业号为56的作业
scancel -n test     #取消作业名为test的作业
scancel -p cpu_c24 #取消提交到cpu_c24队列的作业
scancel -t PENDING #取消正在排队的作业
scancel -w gnode02 #取消运行在gnode02节点上的作业
```

10.资源查询

sinfo显示系统资源


```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
batch*       up   infinite      1 drain* gnode01
batch*       up   infinite     16   idle cnode[01-06],gnode[02-10],master
a100         up   infinite      1 drain* gnode01
a100         up   infinite      8   idle gnode[02-08],master
cpu          up   infinite      7   idle cnode[01-06],master
nvlink       up   infinite      3   idle gnode[09-10],master
(base) [test0803@master pytorch-mgpu-cifar10-master]$
```

第一列PARTITION是队列名。
第二列AVAIL是队列可用情况，如果显示up则是可用状态；如果是inact则是不可用状态。
第三列TIMELIMIT是作业运行时间限制，默认是infinite没有限制。
第四列NODES是节点数。
第五列STATE是节点状态，idle是空闲节点，alloc是已被占用节点，comp是正在释放资源的节点，其他状态的节点都不可用。
第六列NODELIST是节点列表。

sinfo -n gnode01
指定显示节点gnode01的使用情况。

11.提交96核心作业

查看96核心脚本

注意：
cpu对列:默认QOS是cpu96,支持最多96核心，如果想使用更多核心请联系管理人员

```
(base) [test0803@master test]$ cat test.sbatch
#!/bin/bash
#SBATCH -J test
#SBATCH -N 1
#STATCH --cpus-per-task=1
#SBATCH --ntasks-per-node=96
#SBATCH -n 96
#SBATCH -p cpu
#SBATCH --qos=cpu96
export LD_LIBRARY_PATH=/opt/intel/compilers_and_libraries_2019.4.243/linux/compiler/lib:/opt/intel/compilers_and
_libraries_2019.4.243/linux/compiler/lib/intel64_lin:$LD_LIBRARY_PATH
MPIRUN=/home/test0803/openmpi415_intel/bin/mpirun
cd $SLURM_SUBMIT_DIR
$MPIRUN -np $SLURM_NPROCS ./xhpl
(base) [test0803@master test]$
```

提交作业

执行命令：sbatch test.sbatch，返回作业号305，如图：

```
(base) [test0803@master test]$ sbatch test.sbatch
Submitted batch job 305
(base) [test0803@master test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               305       cpu     test test0803  R       0:01      1 cnode01
(base) [test0803@master test]$
```

查询作业

联泰集群
LTHPC

```
(base) [test0803@master test]$ sbatch test.sbatch
Submitted batch job 305
(base) [test0803@master test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               305       cpu     test test0803  R       0:01      1 cnode01
(base) [test0803@master test]$
```

查看提交二个96核任务


```
(base) [test0803@master test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               288       cpu     test test0803 PD       0:00      1 (QOSGrpCpuLimit)
               287       cpu     test test0803  R       0:03      1 cnode01
(base) [test0803@master test]$
```

作业排队，显示资源超出限制。

12.提交192核心MPI作业

查看用户的qos


```
sacctmgr show assoc |grep test0803
```

```
sacctmgr(base) [test0803@master ~]$ sacctmgr show assoc |grep test0803
   cluster       lthpc   test0803                         1
                                                                                a100g1,cpu96,nvlink+

(base) [test0803@master ~]$
```

发现QOS没有CPU192

设置权限

用户需要申请联系管理员申请权限
命令如下：

```
sacctmgr show assoc |grep test0803
Sacctmgr modify user test0803 set qos+=cpu192
```

记得重启服务

```
systemctl restart slurmctld
systemctl restart slurmdbd
systemctl restart slurmd
```

修改脚本

```
vim cpu.sbatch
###脚本部分
#!/bin/bash
#SBATCH  -J   test                          ##作业名test
#SBATCH  --ntasks-per-node=96               ##每节点任务数96
#SBATCH  --cpus-per-task=1                  ##每进程的CPU数1，默认为1
#SBATCH  -N 2                               ##指定使用节点数为2
#SBATCH  -n 192                             ##指定总任务数为192
#SBATCH  --qos=cpu192                       ##指定QOS为cpu192
#SBATCH  -p cpu                             ##指定队列名cpu

###程序部分
export
LD_LIBRARY_PATH=/opt/intel/compilers_amd_libraries_2019.4.243/linux/compiler/lib:/opt
/intel/compilers_and_libraries_2019.4.243/linux/compiler/lib/intel64_lin:$LD_LIBRARY_
PATH
MPIRUN=“/opt/openmpi415_intel/bin/mpirun”
cd $SLURM_SUBMIT_DIR                        ##SLURM_SUBMIT_DIR是脚本目录
$MPIRUN -np $SLURM_NPROCS ./xhpl            ##SLURM_NPROCS是使用的任务数
```

查看脚本


```
(base) [test0803@master test]$ cat test.sbatch
#!/bin/bash
#SBATCH -J test
#SBATCH -N 2
#SBATCH --ntasks-per-node=96
#SBATCH -n 192
#SBATCH -p cpu
#SBATCH --qos=cpu192
export LD_LIBRARY_PATH=/opt/intel/compilers_and_libraries_2019.4.243/linux/compiler/lib:/opt/intel/compilers_and
_libraries_2019.4.243/linux/compiler/lib/intel64_lin:$LD_LIBRARY_PATH
MPIRUN=/home/test0803/openmpi415_intel/bin/mpirun
cd $SLURM_SUBMIT_DIR
$MPIRUN -np $SLURM_NPROCS ./xhpl
```

备注：--cpus-per-task不设置默认为1

提交作业

sbatch test.sbatch

```
(base) [test0803@master test]$ vim test.sbatch
(base) [test0803@master test]$ sbatch test.sbatch
Submitted batch job 289
```

查看作业

squeue

```
(base) [test0803@master test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               289       cpu     test test0803  R       0:28      2 cnode[01-02]
```

作业显示两个节点正在运行，作业正常

13.提交576核MPI作业

查看用户qos

sacctmgr show assoc |grep test0803

```
root@master ~]# sacctmgr show assoc |grep test0803
  cluster      lthpc   test0803                    1
                                                                                cpu192,cpu96
```

显示qos不支持cpu576

查看作业脚本

```
(base) [test0803@master test]$ cat test.sbatch
#!/bin/bash
#SBATCH -J test
#SBATCH -N 6
#SBATCH --ntasks-per-node=96
#SBATCH -n 576
#SBATCH -p cpu
#SBATCH --qos=cpu576
export LD_LIBRARY_PATH=/opt/intel/compilers_and_libraries_2019.4.243/linux/compiler/lib:/opt/intel/compilers_and
_libraries_2019.4.243/linux/compiler/lib/intel64_lin:$LD_LIBRARY_PATH
MPIRUN=/home/test0803/openmpi415_intel/bin/mpirun
cd $SLURM_SUBMIT_DIR
$MPIRUN -np $SLURM_NPROCS ./xhpl
```

作业脚本显示qos为cpu576

提交作业

```
(base) [test0803@master test]$ vim test.sbatch
(base) [test0803@master test]$ sbatch test.sbatch
sbatch: error: Batch job submission failed: Invalid qos specification
```

会提示无效的QOS

14. 设置cpus-per-task不为1提交作业

查看作业脚本

查看作业脚本，显示每任务24核心，任务数4个

```
[root@master intelxhpl]# cat intel.sbatch
#!/bin/bash
#SBATCH -J intelxhpl
#SBATCH -N 1
#STATCH --cpus-per-task=24
#SBATCH --ntasks-per-node=96
#SBATCH -n 4
#SBATCH -p cpu
#SBATCH --qos=cpu96
export LD_LIBRARY_PATH=/opt/intel/compilers_and_libraries_2019.4.243/linux/compiler/lib:/opt/intel/compilers_and
_libraries_2019.4.243/linux/compiler/lib/intel64_lin:$LD_LIBRARY_PATH
source /opt/intel/compilers_and_libraries_2019.4.243/linux/bin/compilervars.sh intel64
cd $SLURM_SUBMIT_DIR
./runme_intel64_static
[root@master intelxhpl]#
```

提交作业

sbatch intel.slurm

```
(base) [test0803@master intelxhpl]$ sbatch intel.sbatch
Submitted batch job 313
(base) [test0803@master intelxhpl]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               313       cpu intelxhp test0803  R       0:00      1 cnode01
```

scontrol show jobs 313

```
(base) [test0803@master intelxhpl]$ scontrol show job 313
JobId=313 JobName=intelxhpl
   UserId=test0803(1195) GroupId=test0803(1195) MCS_label=N/A
   Priority=4294901749 Nice=0 Account=lthpc QOS=cpu96
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:01:20 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2023-09-10T08:52:54 EligibleTime=2023-09-10T08:52:54
   AccrueTime=2023-09-10T08:52:54
   StartTime=2023-09-10T08:52:55 EndTime=Unknown Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-09-10T08:52:55 Scheduler=Main
   Partition=cpu AllocNode:Sid=master:87495
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=cnode01
   BatchHost=cnode01
   NumNodes=1 NumCPUs=96 NumTasks=96 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=96,node=1,billing=96
   Socks/Node=* NtasksPerN:B:S:C=96:0:*:* CoreSpec=*
   MinCPUsNode=96 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/test0803/intelxhpl/intel.sbatch
   WorkDir=/home/test0803/intelxhpl
   StdErr=/home/test0803/intelxhpl/slurm-313.out
   StdIn=/dev/null
   StdOut=/home/test0803/intelxhpl/slurm-313.out
   Power=
```

节点查看

节点查看可以看到4个进程，每个进程2400（即24核心，每核心100%）

```
%Cpu(s): 98.1 us,  1.9 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 21121251+total, 16717018+free, 12087184+used, 31955142+buff/cache
KiB Swap: 33554428 total, 33554428 free,        0 used. 19890609+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
56656 test0803  20   0   13.4g  13.1g   4472 R  2393  0.6  10:04.85 xhpl_intel64_st
56657 test0803  20   0   13.3g  13.1g   4472 R  2393  0.6  10:04.92 xhpl_intel64_st
56654 test0803  20   0   13.4g  13.2g   4532 R  2392  0.7  10:04.46 xhpl_intel64_st
56655 test0803  20   0   13.4g  13.2g   4628 R  2391  0.7  10:04.89 xhpl_intel64_st
182055 gdm      20   0    9.9g 139308  59392 S   1.6  0.0  27:47.81 gnome-shell
57034 root      20   0  165476   3516   1648 R   1.0  0.0   0:00.17 top
56642 root      20   0  278224   4084   2560 S   0.3  0.0   0:00.12 slurmstepd
242359 zabbix   20   0   27812   7680   7460 S   0.3  0.0  61:00.07 zabbix_agentd
```

查看程序的结果，显示4个任务，结果正常。

```
--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be            1.110223e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC23C2C4       100400   384     2     2             140.70              4.79544e+03
HPL_pdgesv() start time Sun Sep 10 08:55:01 2023

HPL_pdgesv() end time   Sun Sep 10 08:57:22 2023

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0042113 ...... PASSED
================================================================================

Finished      1 tests with the following results:
              1 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------

End of Tests.
================================================================================
```

15.提交1张A100脚本

半张a100脚本

备注：A100分区默认qos为a100g1

```
vim a100.sbatch
###脚本部分
#!/bin/bash
#SBATCH  -J  yolov5s_1                    ##作业名yolov5s_1
#SBATCH  --gres=gpu:a100:1                ##指定使用gpu资源,a100数量1
#SBATCH  -N 1                             ##指定使用1个节点数
#SBATCH  -n 1                             ##指定总任务数1
#SBATCH  -o test.out                      ##指定输出文件
#SBATCH  -e test.error                    ##指定错误输出文件
#SBATCH  --qos=a100g1                     ##指定qos为a100g1
#SBATCH  -p a100                          ##指定队列a100
###程序部分
cd $SLURM_SUBMIT_DIR
/home/test0803/anaconda3/envs/pytorch2.0/bin/python   train.py  --batch 64  -
-data coco.yaml   --weights  yolov5s.pt --device 0
```

查看作业脚本

联泰集群
LTHPC

```
(base) [test0803@master yolov5-1]$ cat yolov5.slurm
#!/bin/bash
#SBATCH -J yolov5_1
#SBATCH --gres=gpu:a100:1
#SBATCH -N 1
#SBATCH -n 1
##SBATCH -o test.out
##SBATCH -e test.error
#SBATCH -p a100
#SBATCH --qos=a100g1
export OMPI_MCA_btl_openib_allow_ib=1
echo #CUDA_VISIBLE_DEVICES
cd $SLURM_SUBMIT_DIR
/home/test0803/anaconda3/envs/pytorch2.0/bin/python   train.py --batch 64 --data
 coco.yaml  --weights yolov5s.pt --device  $CUDA_VISIBLE_DEVICES
(base) [test0803@master yolov5-1]$
```

提交作业


```
(base) [test0803@master yolov5-1]$ vim yolov5.slurm
(base) [test0803@master yolov5-1]$ sbatch yolov5.slurm
Submitted batch job 315
(base) [test0803@master yolov5-1]$
```

查询作业

```
(base) [test0803@master yolov5-1]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               315      a100 yolov5_1 test0803  R       0:58      1 gnode02
(base) [test0803@master yolov5-1]$
```

查询作业

scontrol show jobs 414

```
  AccrueTime=2023-09-11T20:06:32
  StartTime=2023-09-11T20:06:33 EndTime=Unknown Deadline=N/A
  SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-09-11T20:06:33 Scheduler
Main
  Partition=a100 AllocNode:Sid=master:52261
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=gnode03
  BatchHost=gnode03
  NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
  TRES=cpu=1,node=1,billing=1,gres/gpu:a100=1
  Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
  MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
  Features=(null) DelayBoot=00:00:00
  OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
  Command=/home/test0803/yolov5-1/yolov5.slurm
  WorkDir=/home/test0803/yolov5-1
  StdErr=/home/test0803/yolov5-1/slurm-414.out
  StdIn=/dev/null
  StdOut=/home/test0803/yolov5-1/slurm-414.out
  Power=
  TresPerNode=gres:gpu:a100:1
```

执行nvidia-smi

```
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA A100-PCIE-40GB          On  | 00000000:17:00.0 Off |                    0 |
| N/A   31C    P0              39W / 250W |   1179MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-PCIE-40GB          On  | 00000000:25:00.0 Off |                    0 |
| N/A   28C    P0              32W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   2  NVIDIA A100-PCIE-40GB          On  | 00000000:99:00.0 Off |                    0 |
| N/A   28C    P0              31W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   3  NVIDIA A100-PCIE-40GB          On  | 00000000:AD:00.0 Off |                    0 |
| N/A   28C    P0              32W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   4  NVIDIA A100-PCIE-40GB          On  | 00000000:C5:00.0 Off |                    0 |
| N/A   28C    P0              30W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   5  NVIDIA A100-PCIE-40GB          On  | 00000000:D9:00.0 Off |                    0 |
| N/A   28C    P0              33W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A      6509      C   ...naconda3/envs/pytorch2.0/bin/python     1166MiB |
+---------------------------------------------------------------------------------------+
```

16. 提交两张A100作业

提交两个半张a100脚本

备注：A100分区默认qos为a100g1，申请使用a100g2

```
vim a100.sbatch
###脚本部分
#!/bin/bash
#SBATCH  -J  test                          ##作业名test
#SBATCH  --gres=gpu:a100:2                 ##指定使用gpu资源,a100数量2
#SBATCH  -N 1                              ##指定使用1个节点数
#SBATCH  -n 2                              ##指定总任务数2
#SBATCH  -o test.out                       ##指定输出文件
#SBATCH  -e test.error                     ##指定错误输出文件
#SBATCH  --qos=a100g2                      ##指定qos为a100g2
#SBATCH  -p a100                           ##指定队列a100
###程序部分
cd $SLURM_SUBMIT_DIR
echo $CUDA_VISIBLE_DEVICE
/home/test0803/anaconda3/envs/pytorch2.0/bin/python    train_cifar10.py
```

提交作业

联泰集群
LTHPC

```
(base) [test0803@gnode03 pytorch-mgpu-cifar10-master]$ vim pytorch2.slurm
(base) [test0803@gnode03 pytorch-mgpu-cifar10-master]$ sbatch pytorch2.slurm
Submitted batch job 431
```

查看作业使用资源

联泰集群
LTHPC

显示2个任务，2张卡

```
JobId=431 JobName=test
   UserId=test0803(1195) GroupId=test0803(1195) MCS_label=N/A
   Priority=4294901672 Nice=0 Account=lthpc QOS=a100g2
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:04:52 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2023-09-11T20:48:27 EligibleTime=2023-09-11T20:48:27
   AccrueTime=2023-09-11T20:48:27
   StartTime=2023-09-11T20:48:27 EndTime=Unknown Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-09-11T20:48:27 Scheduler=Main
   Partition=a100 AllocNode:Sid=gnode03:8509
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=gnode03
   BatchHost=gnode03
   NumNodes=1 NumCPUs=2 NumTasks=2 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=2,node=1,billing=2,gres/gpu:a100=2
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/test0803/pytorch-mgpu-cifar10-master/pytorch2.slurm
   WorkDir=/home/test0803/pytorch-mgpu-cifar10-master
   StdErr=/home/test0803/pytorch-mgpu-cifar10-master/test.error
   StdIn=/dev/null
   StdOut=/home/test0803/pytorch-mgpu-cifar10-master/test.out
   Power=
   TresPerNode=gres:gpu:a100:2
```

节点查看


```
|=========================================+======================+======================|
|   0  NVIDIA A100-PCIE-40GB          On  | 00000000:17:00.0 Off |                    0 |
| N/A   33C    P0              62W / 250W |    737MiB / 40960MiB |      4%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-PCIE-40GB          On  | 00000000:25:00.0 Off |                    0 |
| N/A   30C    P0              54W / 250W |    635MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   2  NVIDIA A100-PCIE-40GB          On  | 00000000:99:00.0 Off |                    0 |
| N/A   29C    P0              51W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   3  NVIDIA A100-PCIE-40GB          On  | 00000000:AD:00.0 Off |                    0 |
| N/A   29C    P0              39W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   4  NVIDIA A100-PCIE-40GB          On  | 00000000:C5:00.0 Off |                    0 |
| N/A   30C    P0              51W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   5  NVIDIA A100-PCIE-40GB          On  | 00000000:D9:00.0 Off |                    0 |
| N/A   29C    P0              56W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A     47825      C   ...naconda3/envs/pytorch2.0/bin/python      734MiB |
|    1   N/A  N/A     47825      C   ...naconda3/envs/pytorch2.0/bin/python      622MiB |
+-----------------------------------------------------------------------------------------+
```

17. 使用—gpus-per-task提交作业

提交两张a100脚本

备注：A100分区默认qos为a100g1，申请使用a100g2

```
vim a100.sbatch
###脚本部分
#!/bin/bash
#SBATCH  -J  cifar10                          ##作业名cifar10
#SBATCH  -N 1                                 ##指定使用1个节点数
#SBATCH  -n 1                                 ##指定总任务数1
#SBATCH  -o test.out                          ##指定输出文件
#SBATCH  -e test.error                        ##指定错误输出文件
#SBATCH  --qos=a100g2                         ##指定qos为a100g2
#SBATCH  -p a100                              ##指定队列a100
#SBATCH  --gpus-per-task=100:2                ##指定每个任务的GPU为2，格式：a100:2
###程序部分
cd $SLURM_SUBMIT_DIR
/home/test0803/anaconda3/envs/pytorch2.0/bin/python   train_cifar10.py
```

查看作业脚本


```
(base) [test0803@gnode03 pytorch-mgpu-cifar10-master]$ cat pytorch.slurm
#!/bin/bash
#SBATCH -J pytorch
##SBATCH --gres=gpu:a100:1
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --qos=a100g2
#SBATCH -p a100
#SBATCH --gpus-per-task=a100:2
export OMPI_MCA_btl_openib_allow_ib=1
echo $CUDA_VISIBLE_DEVICES
cd /home/test0803/pytorch-mgpu-cifar10-master
/home/test0803/anaconda3/envs/pytorch2.0/bin/python  train_cifar10.py
```

提交作业脚本

联泰集群
LTHPC

```
(base) [test0803@gnode03 pytorch-mgpu-cifar10-master]$ sbatch pytorch.slurm
Submitted batch job 428
(base) [test0803@gnode03 pytorch-mgpu-cifar10-master]$ squeue
```

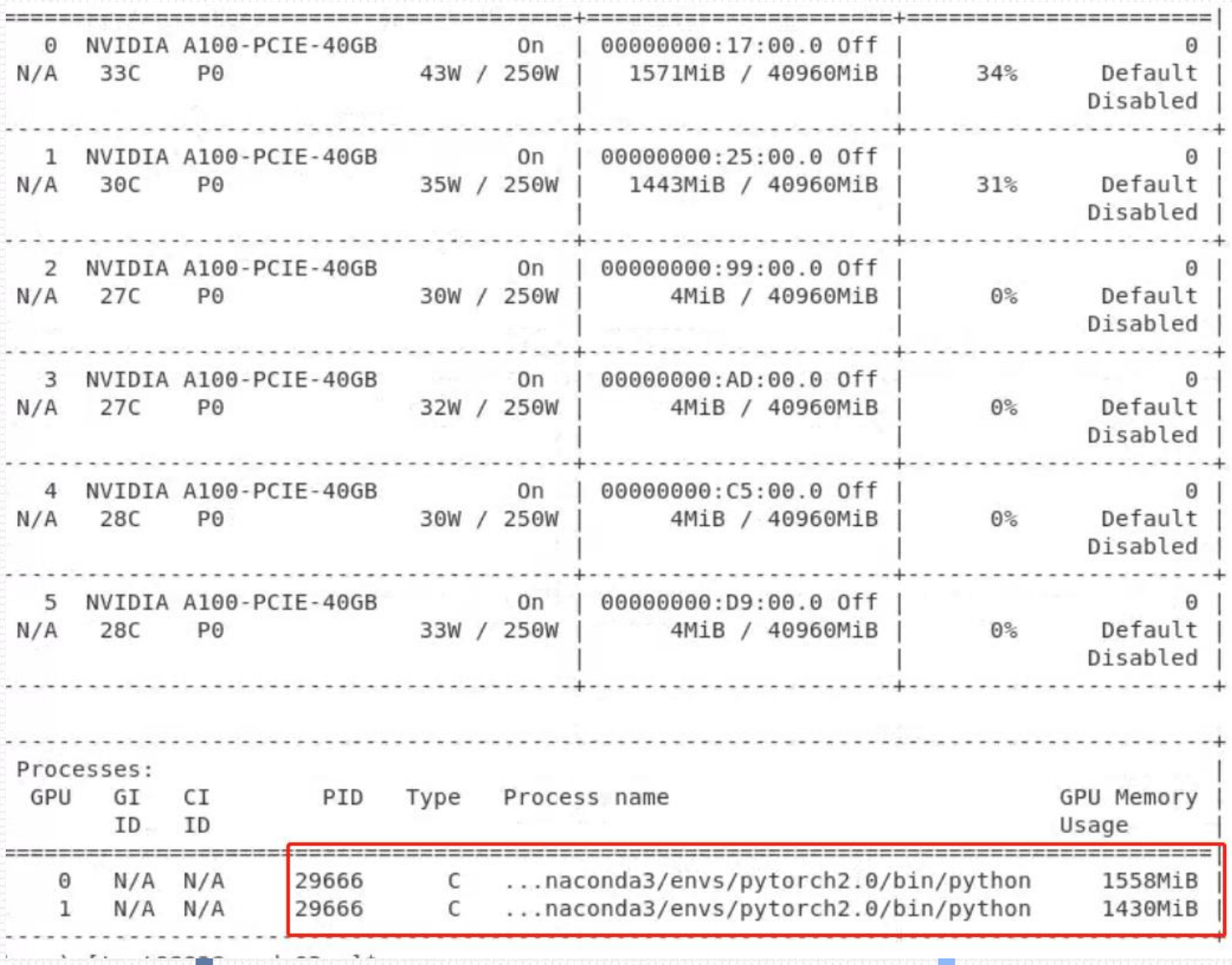
查看作业使用情况

联泰集群 LTHPC

scontrol show jobs 428　(显示1任务，2张GPU卡)

```
base) [test0803@gnode03 pytorch-mgpu-cifar10-master]$ scontrol show jobs 428
obId=428 JobName=pytorch
  UserId=test0803(1195) GroupId=test0803(1195) MCS_label=N/A
  Priority=4294901675 Nice=0 Account=lthpc QOS=a100g2
  JobState=RUNNING Reason=None Dependency=(null)
  Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
  RunTime=00:03:36 TimeLimit=UNLIMITED TimeMin=N/A
  SubmitTime=2023-09-11T20:40:40 EligibleTime=2023-09-11T20:40:40
  AccrueTime=2023-09-11T20:40:40
  StartTime=2023-09-11T20:40:40 EndTime=Unknown Deadline=N/A
  SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-09-11T20:40:40 Scheduler=Main
  Partition=a100 AllocNode:Sid=gnode03:8509
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=gnode03
  BatchHost=gnode03
  NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
  TRES=cpu=1,node=1,billing=1,gres/gpu:a100=2
  Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
  MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
  Features=(null) DelayBoot=00:00:00
  OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
  Command=/home/test0803/pytorch-mgpu-cifar10-master/pytorch.slurm
  WorkDir=/home/test0803/pytorch-mgpu-cifar10-master
  StdErr=/home/test0803/pytorch-mgpu-cifar10-master/slurm-428.out
  StdIn=/dev/null
  StdOut=/home/test0803/pytorch-mgpu-cifar10-master/slurm-428.out
  Power=
  TresPerTask=gres:gpu:a100:2
```

节点查看结果



```
|=========================================+======================+======================|
|   0  NVIDIA A100-PCIE-40GB           On | 00000000:17:00.0 Off |                    0 |
| N/A   33C    P0              43W / 250W |   1571MiB / 40960MiB |     34%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-PCIE-40GB           On | 00000000:25:00.0 Off |                    0 |
| N/A   30C    P0              35W / 250W |   1443MiB / 40960MiB |     31%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   2  NVIDIA A100-PCIE-40GB           On | 00000000:99:00.0 Off |                    0 |
| N/A   27C    P0              30W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   3  NVIDIA A100-PCIE-40GB           On | 00000000:AD:00.0 Off |                    0 |
| N/A   27C    P0              32W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   4  NVIDIA A100-PCIE-40GB           On | 00000000:C5:00.0 Off |                    0 |
| N/A   28C    P0              30W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   5  NVIDIA A100-PCIE-40GB           On | 00000000:D9:00.0 Off |                    0 |
| N/A   28C    P0              33W / 250W |      4MiB / 40960MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A     29666      C   ...naconda3/envs/pytorch2.0/bin/python     1558MiB |
|    1   N/A  N/A     29666      C   ...naconda3/envs/pytorch2.0/bin/python     1430MiB |
+---------------------------------------------------------------------------------------+
```

18. 提交1张nvlink作业

查看作业脚本

设置—gres=gpu:a800:1

```
root@master pytorch-mgpu-cifar10-master]# cat pytorch.slurm
!/bin/bash
SBATCH -J pytorch
SBATCH --gres=gpu:a800:1
SBATCH -N 1
SBATCH -n 2
SBATCH --qos=nvlinkg8
SBATCH -p nvlink
xport OMPI_MCA_btl_openib_allow_ib=1
cho $CUDA_VISIBLE_DEVICES
vidia-smi
d /home/test0803/pytorch-mgpu-cifar10-master
home/test0803/anaconda3/envs/pytorch2.0/bin/python  train_cifar10.py
```

提交作业并查看

sbatch pytorch.slurm

```
(base) [test0803@master pytorch-mgpu-cifar10-master]$ sbatch pytorch.slurm
Submitted batch job 324
(base) [test0803@master pytorch-mgpu-cifar10-master]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               324    nvlink  pytorch test0803  R       0:02      1 gnode09
```

查看作业

联泰集群 LTHPC

scontrol show jobs 324

```
(base) [test0803@master pytorch-mgpu-cifar10-master]$ scontrol show jobs 324
JobId=324 JobName=pytorch
   UserId=test0803(1195) GroupId=test0803(1195) MCS_label=N/A
   Priority=4294901738 Nice=0 Account=lthpc QOS=nvlinkg8
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:01:01 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2023-09-10T10:27:47 EligibleTime=2023-09-10T10:27:47
   AccrueTime=2023-09-10T10:27:47
   StartTime=2023-09-10T10:27:47 EndTime=Unknown Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-09-10T10:27:47 Scheduler=Main
   Partition=nvlink AllocNode:Sid=master:87495
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=gnode09
   BatchHost=gnode09
   NumNodes=1 NumCPUs=2 NumTasks=2 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=2,node=1,billing=2,gres/gpu:a800=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/test0803/pytorch-mgpu-cifar10-master/pytorch.slurm
   WorkDir=/home/test0803/pytorch-mgpu-cifar10-master
   StdErr=/home/test0803/pytorch-mgpu-cifar10-master/slurm-324.out
   StdIn=/dev/null
   StdOut=/home/test0803/pytorch-mgpu-cifar10-master/slurm-324.out
   Power=
   TresPerNode=gres:gpu:a800:1
```

节点查看


```
un Sep 10 10:30:29 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.86.10              Driver Version: 535.86.10    CUDA Version: 12.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A800-SXM4-80GB           On | 00000000:10:00.0 Off |                    0 |
| N/A   57C    P0             271W / 400W |   1999MiB / 81920MiB |     82%      Default |
|                               |                      |                  Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A800-SXM4-80GB           On | 00000000:16:00.0 Off |                    0 |
| N/A   30C    P0              54W / 400W |      2MiB / 81920MiB |      0%      Default |
|                               |                      |                  Disabled |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA A800-SXM4-80GB           On | 00000000:49:00.0 Off |                    0 |
| N/A   30C    P0              55W / 400W |      2MiB / 81920MiB |      0%      Default |
```

19. 提交4张nvlink作业

查看作业脚本

设置—gres=gpu:a800:4

```
(base) [test0803@master pytorch-mgpu-cifar10-master]$ cat pytorch.slurm
#!/bin/bash
#SBATCH -J pytorch
#SBATCH --gres=gpu:a800:4
#SBATCH -N 1
#SBATCH -n 4
#SBATCH --qos=nvlinkg8
#SBATCH -p nvlink
export OMPI_MCA_btl_openib_allow_ib=1
echo $CUDA_VISIBLE_DEVICES
nvidia-smi
cd /home/test0803/pytorch-mgpu-cifar10-master
/home/test0803/anaconda3/envs/pytorch2.0/bin/python  train_cifar10.py
(base) [test0803@master pytorch-mgpu-cifar10-master]$
```

提交作业并查看

sbatch pytorch.slurm

```
sbatch: error: for instance: #!/bin/sh
(base) [test0803@master pytorch-mgpu-cifar10-master]$ sbatch pytorch.slurm
Submitted batch job 326
(base) [test0803@master pytorch-mgpu-cifar10-master]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               325    nvlink  pytorch test0803  R       4:42      1 gnode10
               326    nvlink  pytorch test0803  R       0:02      1 gnode09
(base) [test0803@master pytorch-mgpu-cifar10-master]$
```

查看作业

scontrol show jobs 326

```
(base) [test0803@master pytorch-mgpu-cifar10-master]$ scontrol  show jobs 326
JobId=326 JobName=pytorch
   UserId=test0803(1195) GroupId=test0803(1195) MCS_label=N/A
   Priority=4294901736 Nice=0 Account=lthpc QOS=nvlinkg8
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:01:27 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2023-09-10T10:36:40 EligibleTime=2023-09-10T10:36:40
   AccrueTime=2023-09-10T10:36:40
   StartTime=2023-09-10T10:36:40 EndTime=Unknown Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-09-10T10:36:40 Scheduler=Main
   Partition=nvlink AllocNode:Sid=master:87495
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=gnode09
   BatchHost=gnode09
   NumNodes=1 NumCPUs=4 NumTasks=4 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=4,node=1,billing=4,gres/gpu:a800=4
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/test0803/pytorch-mgpu-cifar10-master/pytorch.slurm
   WorkDir=/home/test0803/pytorch-mgpu-cifar10-master
   StdErr=/home/test0803/pytorch-mgpu-cifar10-master/slurm-326.out
   StdIn=/dev/null
   StdOut=/home/test0803/pytorch-mgpu-cifar10-master/slurm-326.out
   Power=
   TresPerNode=gres:gpu:a800:4
```

节点查看

联泰集群 LTHPC

```
(base) [test0803@gnode09 ~]$ nvidia-smi
Sun Sep 10 10:39:36 2023
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.86.10              Driver Version: 535.86.10    CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA A800-SXM4-80GB           On | 00000000:10:00.0 Off |                    0 |
| N/A   33C    P0              91W / 400W |   1625MiB / 81920MiB |     28%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   1  NVIDIA A800-SXM4-80GB           On | 00000000:16:00.0 Off |                    0 |
| N/A   31C    P0              78W / 400W |   1611MiB / 81920MiB |     20%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   2  NVIDIA A800-SXM4-80GB           On | 00000000:49:00.0 Off |                    0 |
| N/A   32C    P0              91W / 400W |   1619MiB / 81920MiB |     24%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   3  NVIDIA A800-SXM4-80GB           On | 00000000:4D:00.0 Off |                    0 |
| N/A   31C    P0              79W / 400W |   1515MiB / 81920MiB |     23%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
```

tail –f slurm-326.out

```
[==============>.....................................................]  Step: 20ms | Tot: 2s68ms | Loss: 0.396 |
[==============>.....................................................]  Step: 19ms | Tot: 2s87ms | Loss: 0.398 |
[==============>.....................................................]  Step: 20ms | Tot: 2s108ms | Loss: 0.398 |
[==============>.....................................................]  Step: 19ms | Tot: 2s127ms | Loss: 0.399 |
[==============>.....................................................]  Step: 20ms | Tot: 2s148ms | Loss: 0.400 |
[==============>.....................................................]  Step: 20ms | Tot: 2s168ms | Loss: 0.399 |
[==============>.....................................................]  Step: 19ms | Tot: 2s188ms | Loss: 0.399 |
[===============>.....................................................]  Step: 31ms | Tot: 2s219ms | Loss: 0.399 |
[===============>.....................................................]  Step: 30ms | Tot: 2s250ms | Loss: 0.399 |
[===============>.....................................................]  Step: 29ms | Tot: 2s279ms | Loss: 0.399 |
[===============>.....................................................]  Step: 26ms | Tot: 2s306ms | Loss: 0.398 |
[===============>.....................................................]  Step: 27ms | Tot: 2s333ms | Loss: 0.399 |
[===============>.....................................................]  Step: 36ms | Tot: 2s369ms | Loss: 0.398 |
[================>.....................................................]  Step: 23ms | Tot: 2s393ms | Loss: 0.398 |
[================>.....................................................]  Step: 27ms | Tot: 2s420ms | Loss: 0.399 |
[================>.....................................................]  Step: 33ms | Tot: 2s453ms | Loss: 0.399 |
[================>.....................................................]  Step: 31ms | Tot: 2s485ms | Loss: 0.399 |
[================>.....................................................]  Step: 27ms | Tot: 2s513ms | Loss: 0.399 |
[================>.....................................................]  Step: 28ms | Tot: 2s541ms | Loss: 0.400 |
[=================>.....................................................]  Step: 26ms | Tot: 2s568ms | Loss: 0.400 |
[=================>.....................................................]  Step: 27ms | Tot: 2s595ms | Loss: 0.402 |
[=================>.....................................................]  Step: 23ms | Tot: 2s619ms | Loss: 0.402 |
[=================>.....................................................]  Step: 19ms | Tot: 2s639ms | Loss: 0.402 |
[=================>.....................................................]  Step: 19ms | Tot: 2s658ms | Loss: 0.402 |
[=================>.....................................................]  Step: 19ms | Tot: 2s678ms | Loss: 0.402 |
[==================>.....................................................]  Step: 19ms | Tot: 2s698ms | Loss: 0.401 |
[==================>.....................................................]  Step: 20ms | Tot: 2s718ms | Loss: 0.401 |
[==================>.....................................................]  Step: 19ms | Tot: 2s737ms | Loss: 0.401 |
[==================>.....................................................]  Step: 19ms | Tot: 2s757ms | Loss: 0.400 |
[==================>.....................................................]  Step: 19ms | Tot: 2s776ms | Loss: 0.400 |
[==================>.....................................................]  Step: 32ms | Tot: 2s809ms | Loss: 0.400 |
[==================>.....................................................]  Step: 30ms | Tot: 2s839ms | Loss: 0.400 |
[==================>.....................................................]  Step: 29ms | Tot: 2s869ms | Loss: 0.400 |
[==================>.....................................................]  Step: 35ms | Tot: 2s904ms | Loss: 0.400 |
124/391 04% (13714/15872)
```

20. 深度学习框架安装及使用

salloc –p debug –qos=a100g1

```
[test0803@master ~]$ salloc -p debug --qos=a100g1
salloc: Granted job allocation 757
[test0803@master ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               532      a100  cryo_em subaoqua  R 4-01:48:21      1 gnode02
               731      a100    ppisp liujunta  R    3:11:28      1 gnode01
               729       cpu interact  xiangcx  R    5:24:56      1 cnode01
               748       cpu interact yuxueshi  R    1:04:46      1 master
               756       cpu      PW1     lidi  R       1:57      1 cnode01
               757     debug interact test0803  R       0:03      1 gnode08
               732    nvlink ppispa80 liujunta  R    3:10:53      1 gnode09
               749    nvlink      GNN subaoqua  R      44:40      1 gnode09
[test0803@master ~]$
```

ssh gnode08

```
[test0803@master ~]$ ssh gnode08
The authenticity of host 'gnode08 (192.168.4.138)' can't be established.
ECDSA key fingerprint is SHA256:AGNLcY19BNOpNVs0qA4vXyZ8VJU7qS2I4qR0YGFlwX0.
ECDSA key fingerprint is MD5:78:05:cc:0f:76:f7:e0:6c:5c:cf:ca:33:8a:70:12:97.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gnode08,192.168.4.138' (ECDSA) to the list of known
hosts.
Last login: Sun Sep 17 11:33:32 2023
```

加载conda环境


```
module avail
module load anaconda3/anaconda3_23
which conda
```

```
(base) [test0803@gnode08 ~]$ module avail

------------------------- /usr/share/Modules/modulefiles -------------------------
dot         module-git  module-info modules     null        use.own

------------------------------ /opt/modulefiles/ -------------------------------
anaconda3/anaconda3_23
(base) [test0803@gnode08 ~]$ module load anaconda3/anaconda3_23
(base) [test0803@gnode08 ~]$ which conda
/opt/anaconda3 2023/bin/conda
```

#初始化环境(第一次需要执行)
conda init
#加载环境
source ~/.bashrc
#创建虚拟环境
conda create –n pytorch1.12 python=3.10

```
(base) [test0803@gnode08 ~]$ conda create -n pytorch1.12 python=3.10
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 23.1.0
  latest version: 23.7.4

Please update conda by running
```

查看虚拟环境

conda env list

```
(base) [test0803@gnode08 ~]$ conda env list
# conda environments:
#
pytorch1.12              /home/test0803/.conda/envs/pytorch1.12
                         /home/test0803/anaconda3
                         /home/test0803/anaconda3/envs/pytorch1.9
                         /home/test0803/anaconda3/envs/pytorch2.0
base                  *  /opt/anaconda3_2023

(base) [test0803@gnode08 ~]$
```

激活虚拟环境

conda activate pytorch1.12

```
(base) [test0803@gnode08 ~]$ conda activate pytorch1.12
(pytorch1.12) [test0803@gnode08 ~]$
```

pip install torch==1.13.0+cu117 torchvision==0.14.0+cu117 torchaudio==0.13.0 –extra-index-url https://download.pytorch.org/whl/cu117

```
(pytorch1.12) [test0803@gnode08 ~]$ pip install torch==1.13.0+cu117 torchvision=
=0.14.0+cu117 torchaudio==0.13.0 --extra-index-url https://download.pytorch.org/
whl/cu117
Looking in indexes: https://pypi.org/simple, https://download.pytorch.org/whl/cu
117
Collecting torch==1.13.0+cu117
  Downloading https://download.pytorch.org/whl/cu117/torch-1.13.0%2Bcu117-cp310-
cp310-linux_x86_64.whl (1806.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 GB 1.2 MB/s eta 0:00:00
Collecting torchvision==0.14.0+cu117
  Downloading https://download.pytorch.org/whl/cu117/torchvision-0.14.0%2Bcu117-
cp310-cp310-linux_x86_64.whl (24.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.3/24.3 MB 3.0 MB/s eta 0:00:00
Collecting torchaudio==0.13.0
  Downloading https://download.pytorch.org/whl/cu117/torchaudio-0.13.0%2Bcu117-c
```

查看pytorch版本

联泰集群 LTHPC

pip list

```
AmberUtils          21.0
certifi             2023.7.22
charset-normalizer  3.2.0
idna                3.4
MMPBSA.py           16.0
mpi4py              3.0.3
numpy               1.26.0
packmol-memgen      1.1.0rc0
ParmEd              3.4.1
pdb4amber           20.1
Pillow              10.0.1
pip                 23.2.1
pyMSMT              20.0
pytraj              2.0.6
requests            2.31.0
sander              16.0
setuptools          68.0.0
torch               1.13.0+cu117
torchaudio          0.13.0+cu117
torchvision         0.14.0+cu117
typing_extensions   4.7.1
urllib3             2.0.4
wheel               0.38.4
```

修改算例提交脚本

#修改分区为debug
#修改脚本里python路径为虚拟环境的路径

```
[test0803@master yolov5-1]$ cat yolov5_1.slurm
#!/bin/bash
#SBATCH -J yolov5_1
#SBATCH --gres=gpu:a100:1
#SBATCH -N 1
#SBATCH -n 1
##SBATCH -o test.out
##SBATCH -e test.error
#SBATCH -p debug
#SBATCH --qos=a100g1
export OMPI_MCA_btl_openib_allow_ib=1
echo #CUDA_VISIBLE_DEVICES
cd $SLURM_SUBMIT_DIR
/home/test0803/.conda/envs/pytorch1.12/bin/python   train.py --batch 64 --data coco.yaml  --weights yolov5s.pt -
-device  $CUDA_VISIBLE_DEVICES
```

提交算例脚本

sbatch yolov5_1.sbatch

```
[test0803@master yolov5-1]$ vi  yolov5_1.slurm
[test0803@master yolov5-1]$ sbatch yolov5_1.slurm
Submitted batch job 750
```

查看作业状态

联泰集群
LTHPC

squeue

```
[test0803@master yolov5-1]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               532      a100  cryo_em subaoqua  R 4-01:19:16      1 gnode02
               731      a100    ppisp liujunta  R    2:42:23      1 gnode01
               729       cpu interact  xiangcx  R    4:55:51      1 cnode01
               748       cpu interact yuxueshi  R      35:41      1 master
               750     debug yolov5_1 test0803  R       0:11      1 gnode08
               732    nvlink ppispa80 liujunta  R    2:41:48      1 gnode09
               749    nvlink      GNN subaoqua  R      15:35      1 gnode09
```

tail –f slurm-750.out

```
Image sizes 640 train, 640 val
Using 8 dataloader workers
Logging results to runs/train/exp35
Starting training for 100 epochs...

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       0/99      13.7G    0.04068    0.06463    0.01607        796        640:
       0/99      13.7G    0.04068    0.06463    0.01607        796        640:
       0/99      13.7G    0.04266    0.06613    0.01611        841        640:
       0/99      13.7G    0.04266    0.06613    0.01611        841        640:
       0/99      13.7G     0.0425    0.06411      0.016        701        640:
       0/99      13.7G     0.0425    0.06411      0.016        701        640:
       0/99      13.7G    0.04207    0.06269    0.01583        744        640:
       0/99      13.7G    0.04207    0.06269    0.01583        744        640:
       0/99      13.7G    0.04217    0.06317    0.01584        836        640:
       0/99      13.7G    0.04217    0.06317    0.01584        836        640:
       0/99      13.7G    0.04227      0.063    0.01566        921        640:
       0/99      13.7G    0.04227      0.063    0.01566        921        640:
       0/99      13.7G     0.0426    0.06375    0.01564        910        640:
       0/99      13.7G     0.0426    0.06375    0.01564        910        640:
       0/99      13.7G    0.04278    0.06399    0.01576        850        640:
       0/99      13.7G    0.04278    0.06399    0.01576        850        640:
       0/99      13.7G    0.04263     0.0636    0.01597        806        640:
```

21. 节点状态异常

squeue

```
[root@master yolov5-1]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               430      a100 interact liujunta  R      25:33      1 gnode03
               404    nvlink e2e_cros wangwenk CG       0:00      1 gnode09
[root@master yolov5-1]#
```

```
scontrol update NodeName=gnode09  State=down Reason=hung_proc
scontrol  update NodeName=gnode09  State=resume
```

22. Gcc使用

Gcc9.3/gcc8.4使用

系统默认版本是4.8.5,使用高版本的GCC，请使用如下命令加载

```
##查看gcc版本
module avail

##加载gcc版本
module load  gcc/gcc9.3.0

 ##查看已加载的GCC版本
module  list

##卸载GCC版本
module unload gcc/gcc9.3.0
```

联泰集群
LTHPC
专注提供计算力

THANKS

END