联系电话：021-64032612
联系邮箱：hpc@ion.ac.cn
联系地址：岳阳路319号8号楼605

CEBSIT GPU集群使用培训

脑科学数据与计算中心

2021年9月24日

一 使用集群计算资源的方式

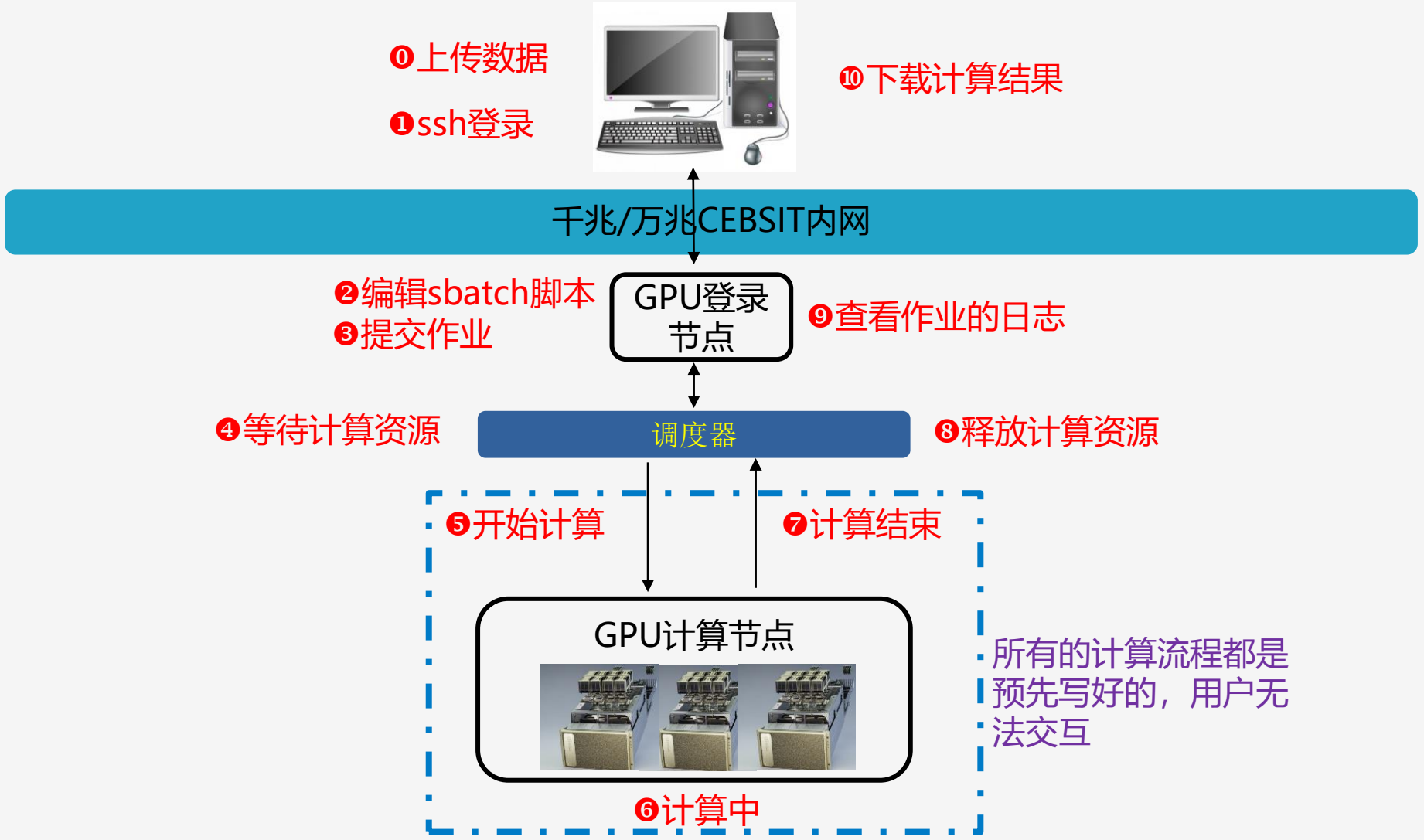
二 集群硬件和软件栈介绍

三 GPU集群容器工具介绍与容器示例

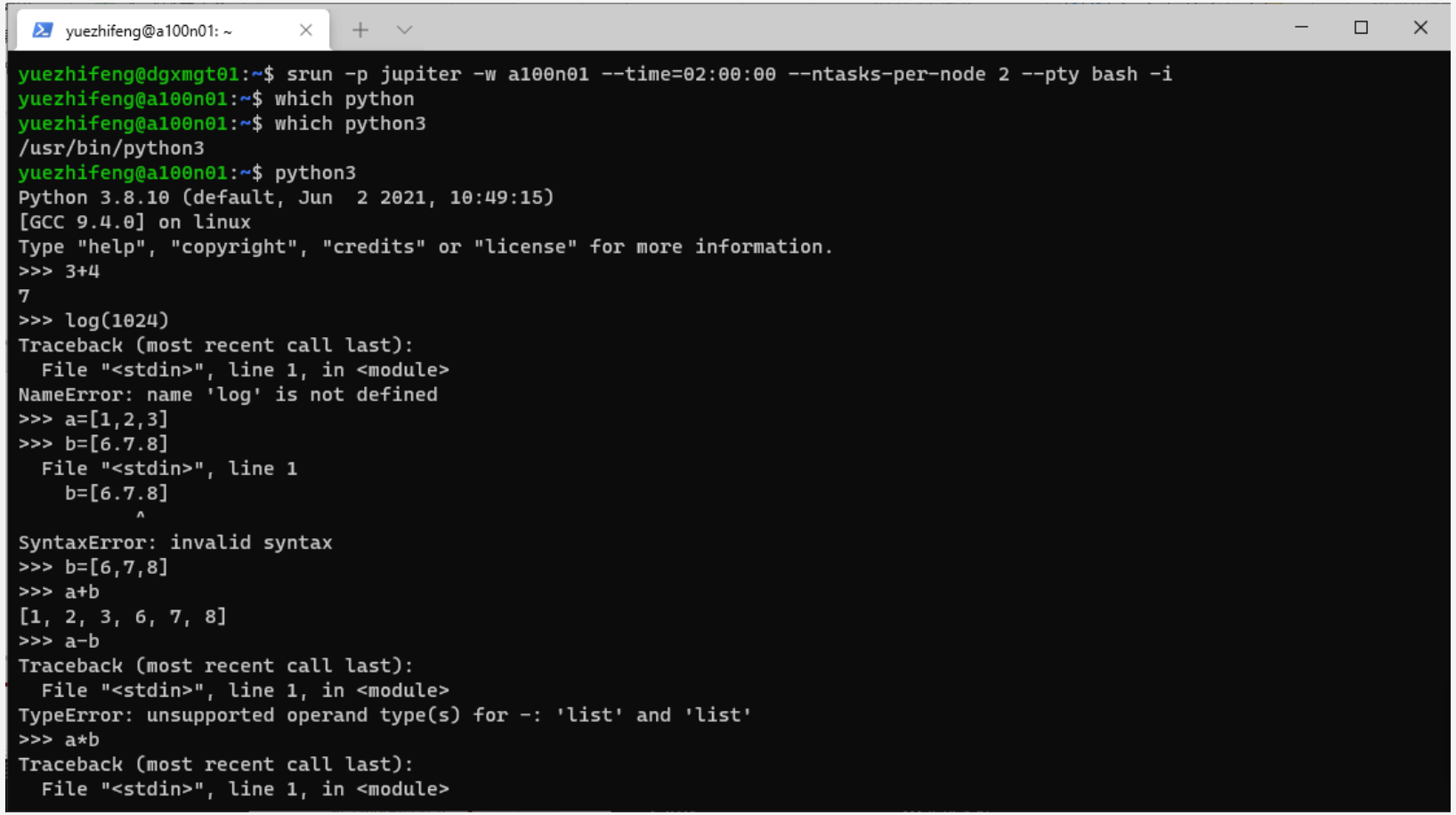
四 交互式使用计算资源的详细步骤

五 集群使用注意事项和账户申请流程

最一般的使用集群计算的流程



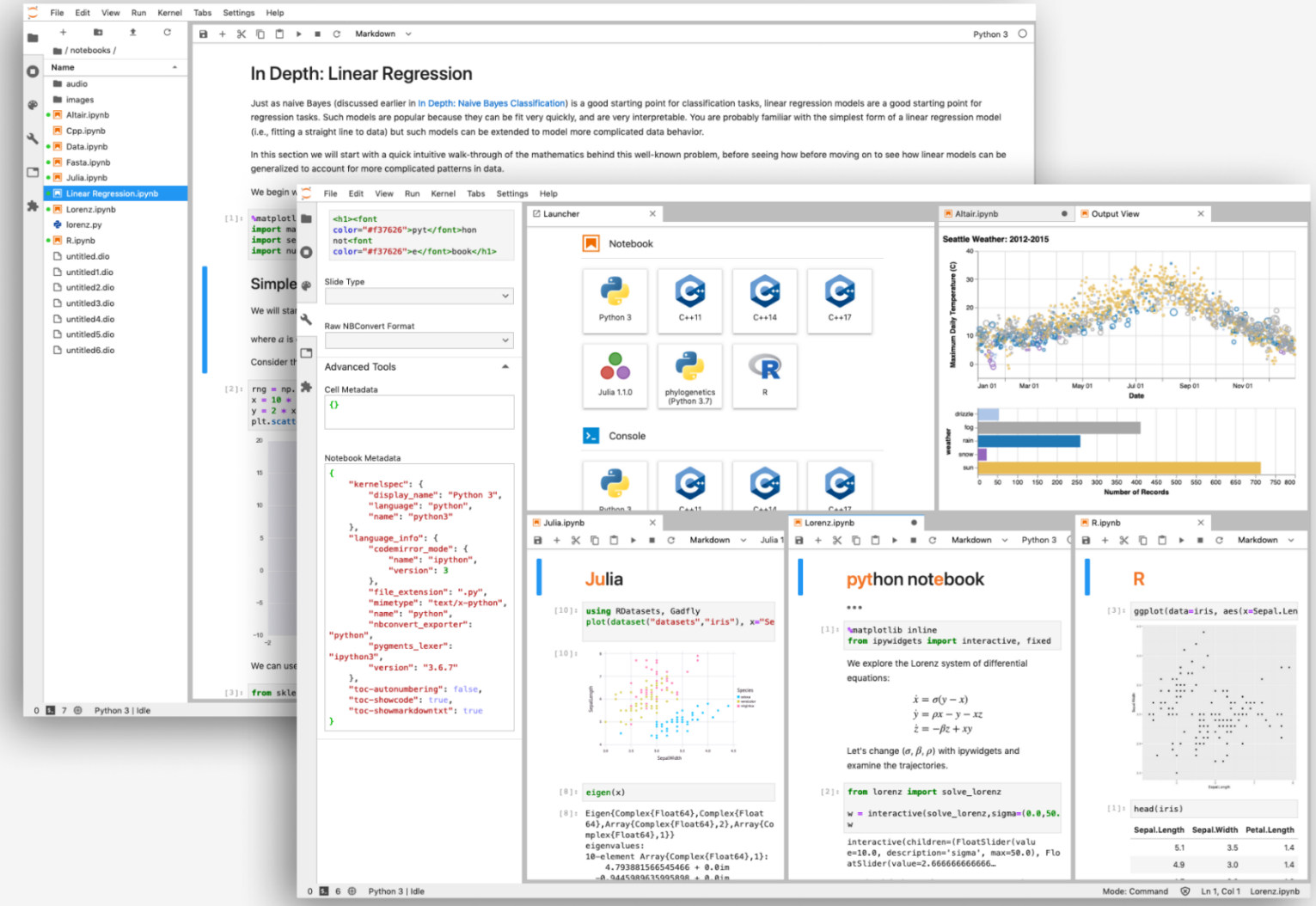
交互式使用计算节点的方式：1. 命令行模式



为了充分使用计算机提供的计算资源，早期很多计算机会连接若干终端控制台，这些终端控制台从硬件上构造很简单，只包括键盘和显示器，**不执行计算的任务，只简单的把用户的输入发送到主计算机去处理，然后再把计算结果返回给用户**。从软件使用上看，只提供给用户一个使用命令行的字符界面，用于接收用户输入和反馈计算结果。

像Windows下的命令行状态，Linux、Unix下的字符终端程序，这些就称为虚拟控制台。

交互式使用计算节点的方式：2. Jupyter



Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。——[Jupyter Notebook官方介绍]

简而言之，Jupyter Notebook是以网页的形式打开，可以在网页页面中**直接编写代码**和**运行代码**，代码的**运行结果**也会直接在代码块下显示的程序。如在编程过程中需要编写说明文档，可在同一个页面中直接编写，便于作及时的说明和解释。

一 使用集群计算资源的方式

二 集群硬件和软件栈介绍

三 GPU集群容器工具介绍与容器示例

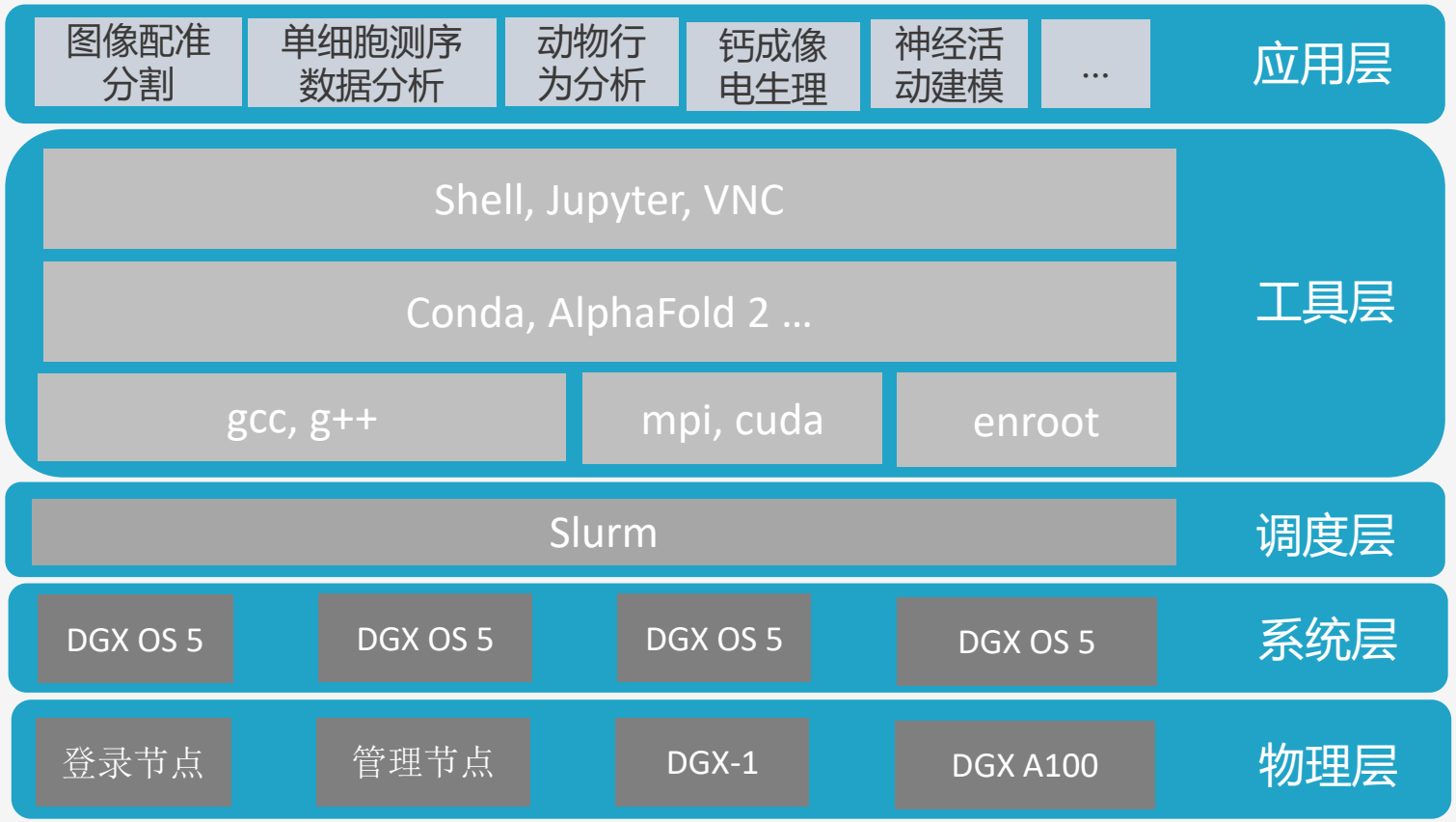
四 交互式使用计算资源的详细步骤

五 集群使用注意事项和账户申请流程

CEBSIT GPU 集群硬件总览

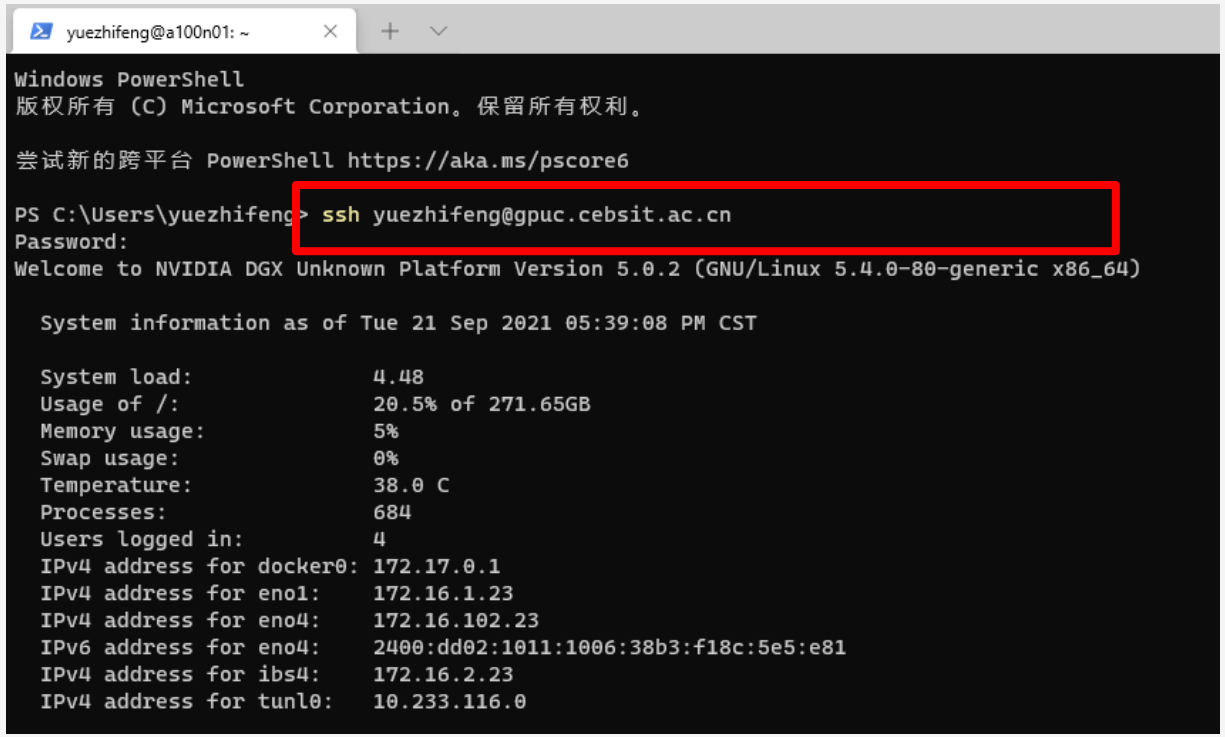
队列	节点型号	节点数	单节点资源
mars	DGX-1	1	40核； 512GB内存； 8xV100 (32GB)； 7TB SSD；
jupiter	DGX A100	4	128核； 1TB内存； 8xA100 (40GB)； 14TB SSD；

CEBSIT GPU 软件栈简介



CEBSIT GPU 集群软件栈

命令行SSH登陆集群：以系统自带PowerShell为例

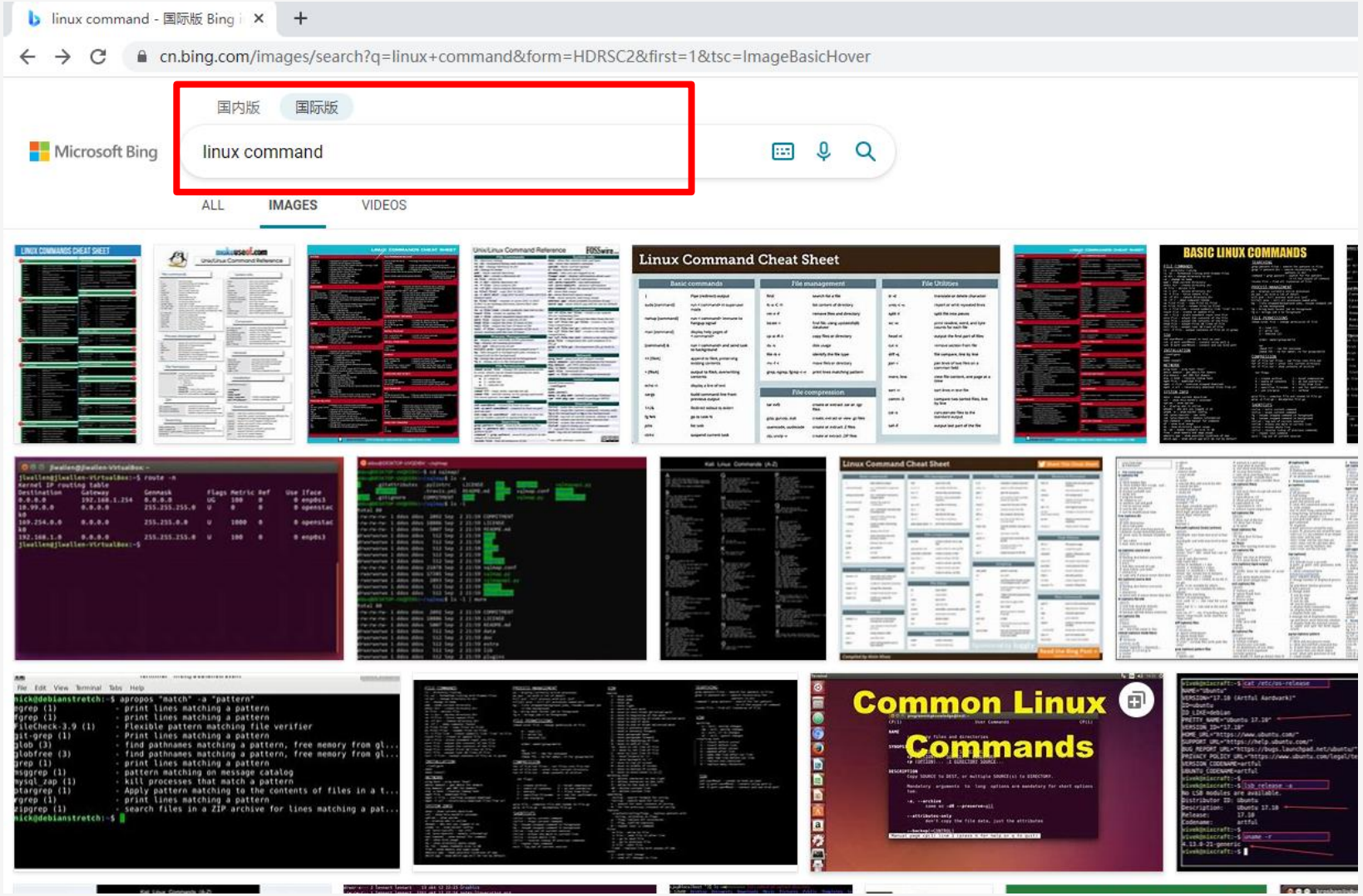


ssh yourname@gpuc.cebsit.ac.cn

Linux基础命令

1. ls : 列出当前或指定目录下的文件或目录
2. passwd: 更改用户登录密码
3. logout: 退出登录
4. pwd: 显示当前目录
5. cd : 进入指定目录
6. chmod :更改文件读、写或执行权限
7. rm: 删除文件或目录
8. cp: 拷贝文件或目录
9. find: 在指定目录下查找文件
10. mv : 文件更名或移动
11. vi :文本编辑器
12. top : 查看系统长时间运行的主要进程
13. ps -ef:查看系统进程。
14. kill : 杀掉一个指定进程号的进程或向系统发送一个信号。
15. man : 给出指定命令的详细描述。
16. date :显示或设置系统时间。
17. rcp,ftp, sftp,scp: 远程文件拷贝。

最快的学习方式是搜索“linux command”



ssh登录后就可以指挥slurm了

CEBSIT GPU 集群软件栈

Slurm常用命令

命令	功能
sacct	查看历史作业信息
salloc	分配资源
sbatch	提交批处理作业
scancel	取消作业
scontrol	系统与作业控制
sinfo	查看节点与分区状态
squeue	查看队列状态
srun	执行作业

sinfo

- 学习sinfo最好的方法就是看help

- https://slurm.schedmd.com/sinfo.html
- http://hpc.pku.edu.cn/_book/guide/slurm/sinfo.html
- https://docs.hpc.sjtu.edu.cn/job/slurm.html#sinfo

```
yuezhifeng@dgxmgt01:~$ sinfo --help
Usage: sinfo [OPTIONS]
  -a, --all                  show all partitions (including hidden and those
                             not accessible)
  -d, --dead                 show only non-responding nodes
  -e, --exact                group nodes only on exact match of configuration
      --federation           Report federated information if a member of one
  -h, --noheader             no headers on output
  --hide                     do not show hidden or non-accessible partitions
  -i, --iterate=seconds      specify an iteration period
      --local                show only local cluster in a federation.
                             Overrides --federation.
  -l, --long                 long output - displays more information
  -M, --clusters=names       clusters to issue commands to. Implies --local.
                             NOTE: SlurmDBD must be up.
  -n, --nodes=NODES          report on specific node(s)
  --noconvert                don't convert units from their original type
                             (e.g. 2048M won't be converted to 2G).
  -N, --Node                 Node-centric format
  -o, --format=format        format specification
  -O, --Format=format        long format specification
  -p, --partition=PARTITION  report on specific partition
  -r, --responding           report only responding nodes
  -R, --list-reasons         list reason nodes are down or drained
  -s, --summarize            report state summary only
  -S, --sort=fields          comma separated list of fields to sort on
  -t, --states=node_state    specify the what states of nodes to view
  -T, --reservation          show only reservation information
  -v, --verbose              verbosity level
  -V, --version              output version information and exit

Help options:
  --help                     show this help message
  --usage                    display brief usage message
```

sinfo

- (A/I/O/T) - allocated/idle/other/total

```
yuezhifeng@dgxmgt01:~$ sinfo -N -o "%10P%10N%15C%10O%10e%20G"
PARTITION NODELIST  CPUS(A/I/O/T)  CPU_LOAD  FREE_MEM  GRES
jupiter*  a100n01   0/256/0/256    1.34      805967    gpu:8
jupiter*  a100n02   0/256/0/256    3.80      922057    gpu:8
jupiter*  a100n03   0/256/0/256    2.91      928446    gpu:8
jupiter*  a100n04   128/128/0/256  7.94      503113    gpu:8
mars      v100n01   40/0/0/40      3.16      5134      gpu:8(S:0-1)
```

```
yuezhifeng@dgxmgt01:~$ sinfo -O nodehost,cpusstate,freemem,gres,gresused
HOSTNAMES           CPUS(A/I/O/T)       FREE_MEM            GRES                GRES_USED
a100n04             128/128/0/256       503113              gpu:8               gpu:(null):4(IDX:0-3
a100n01             0/256/0/256         805967              gpu:8               gpu:(null):0(IDX:N/A
a100n02             0/256/0/256         922057              gpu:8               gpu:(null):0(IDX:N/A
a100n03             0/256/0/256         928446              gpu:8               gpu:(null):0(IDX:N/A
v100n01             40/0/0/40           5134                gpu:8(S:0-1)        gpu:(null):8(IDX:0-7
```

GPU卡的数目

已使用GPU卡的数目

squeue

- 学习squeue最好的方法就是看help

- https://slurm.schedmd.com/squeue.html
- http://hpc.pku.edu.cn/_book/guide/slurm/squeue.html
- https://docs.hpc.sjtu.edu.cn/job/slurm.html#squeue

```
yuezhifeng@dgxmgt01:~$ squeue --help
Usage: squeue [OPTIONS]
  -A, --account=account(s)        comma separated list of accounts
                                  to view, default is all accounts
  -a, --all                       display jobs in hidden partitions
      --array-unique              display one unique pending job array
                                  element per line
      --federation                Report federated information if a member
                                  of one
  -h, --noheader                  no headers on output
      --hide                      do not display jobs in hidden partitions
  -i, --iterate=seconds           specify an interation period
  -j, --job=job(s)                comma separated list of jobs IDs
                                  to view, default is all
      --local                     Report information only about jobs on the
                                  local cluster. Overrides --federation.
  -l, --long                      long report
  -L, --licenses=(license names)  comma separated list of license names to view
  -M, --clusters=cluster_name     cluster to issue commands to.  Default is
                                  current cluster.  cluster with no name will
                                  reset to default. Implies --local.
  -n, --name=job_name(s)          comma separated list of job names to view
      --noconvert                 don't convert units from their original type
                                  (e.g. 2048M won't be converted to 2G).
  -o, --format=format             format specification
  -O, --Format=format             format specification
  -p, --partition=partition(s)    comma separated list of partitions
                                  to view, default is all partitions
  -q, --qos=qos(s)                comma separated list of qos's
                                  to view, default is all qos's
  -R, --reservation=name          reservation to view, default is all
  -r, --array                     display one job array element per line
      --sibling                   Report information about all sibling jobs
                                  on a federated cluster. Implies --federation.
  -s, --step=step(s)              comma separated list of job steps
                                  to view, default is all
  -S, --sort=fields               comma separated list of fields to sort on
      --start                     print expected start times of pending jobs
  -t, --states=states             comma separated list of states to view,
                                  default is pending and running,
                                  '--states=all' reports all states
  -u, --user=user_name(s)         comma separated list of users to view
      --name=job_name(s)          comma separated list of job names to view
  -v, --verbose                   verbosity level
  -V, --version                   output version information and exit
  -w, --nodelist=hostlist         list of nodes to view, default is
                                  all nodes

Help options:
  --help                          show this help message
  --usage                         display a brief summary of squeue options
```

sbatch

- 学习sbatch最好的方法就是看help

- https://slurm.schedmd.com/sbatch.html
- http://hpc.pku.edu.cn/_book/guide/slurm/sbatch.html
- https://docs.hpc.sjtu.edu.cn/job/slurm.html#sbatch

```
yuezhifeng@dgxmgt01:~$ cat run_nvidia-smi
#!/bin/bash

nvidia-smi
yuezhifeng@dgxmgt01:~$ sbatch -p jupiter -w a100n01 --time=02:00:00 --gres=gpu:1 ./run_nvidia-smi
Submitted batch job 2678
yuezhifeng@dgxmgt01:~$ cat slurm-2678.out
Tue Sep 21 18:06:20 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.142.00   Driver Version: 450.142.00   CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| 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  A100-SXM4-40GB      On   | 00000000:07:00.0 Off |                    0 |
| N/A   32C    P0    54W / 400W |      0MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Slurm任务标准输出结果

salloc

- 学习salloc最好的方法就是看help

- https://slurm.schedmd.com/salloc.html
- http://hpc.pku.edu.cn/_book/guide/slurm/salloc.html
- https://docs.hpc.sjtu.edu.cn/job/slurm.html#srun-salloc

```
yuezhifeng@dgxmgt01:~$ salloc -N 1 -n 1 --gres=gpu:2
salloc: Granted job allocation 2679
salloc: Waiting for resource configuration
salloc: Nodes a100n04 are ready for job
yuezhifeng@dgxmgt01:~$ ssh a100n04
Warning: Permanently added 'a100n04,172.16.1.215' (ECDSA) to the list of known hosts.
Welcome to NVIDIA DGX Server Version 5.0.5 (GNU/Linux 5.4.0-80-generic x86_64)

  System information as of Tue 21 Sep 2021 06:08:39 PM CST

  System load:                15.56
  Usage of /:                 8.5% of 1.72TB
  Memory usage:               38%
  Swap usage:                 0%
  Processes:                  2611
  Users logged in:            1
  IPv4 address for docker0:   172.17.0.1
  IPv4 address for enp226s0:  172.16.1.215
  IPv4 address for enp97s0f0: 172.16.102.215
  IPv6 address for enp97s0f0: 2400:dd02:1011:1006:e1f4:78ed:43e0:2ae0
  IPv4 address for ibp225s0f0: 172.16.2.215
  IPv4 address for tunl0:     10.233.67.0

The system has 0 critical alerts and 8 warnings. Use 'sudo nvsm show alerts' for more details.

yuezhifeng@a100n04:~$ nvidia-smi
Tue Sep 21 18:08:49 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.142.00   Driver Version: 450.142.00   CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| 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  A100-SXM4-40GB      On   | 00000000:87:00.0 Off |                    0 |
| N/A   35C    P0    52W / 400W |      0MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  A100-SXM4-40GB      On   | 00000000:90:00.0 Off |                    0 |
| N/A   34C    P0    52W / 400W |      0MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
yuezhifeng@a100n04:~$ python3
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

1. 先申请资源
2. 申请到了ssh过去
3. 交互式的运行命令
4. 或启动例如python

srun

- 学习srun最好的方法就是看help

- https://slurm.schedmd.com/srun.html
- http://hpc.pku.edu.cn/_book/guide/slurm/srun.html
- https://docs.hpc.sjtu.edu.cn/job/slurm.html#srun-salloc

```
yuezhifeng@dgxmgt01:~$ srun -p jupiter -w a100n01 --time=02:00:00 --ntasks-per-node 1 --gres=gpu:1 nvidia-smi
Tue Sep 21 18:14:20 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.142.00   Driver Version: 450.142.00   CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| 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  A100-SXM4-40GB      On   | 00000000:07:00.0 Off |                    0 |
| N/A   32C    P0    54W / 400W |      0MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

srun直接运行命令

scancel

```
yuezhifeng@a100n01:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1662   jupiter     bash    liujz  R 19-19:20:31      1 a100n04
              2681   jupiter     bash yuezhife  R       3:19      1 a100n01
              1756      mars     bash    liujz  R 17-20:13:09      1 v100n01
yuezhifeng@a100n01:~$ scancel 2681
srun: Force Terminated job 2681
yuezhifeng@a100n01:~$ srun: Job step aborted: Waiting up to 32 seconds for job step to finish.
slurmstepd: error: *** STEP 2681.0 ON a100n01 CANCELLED AT 2021-09-21T18:18:30 ***
exit
yuezhifeng@dgxmgt01:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1662   jupiter     bash    liujz  R 19-19:20:49      1 a100n04
              1756      mars     bash    liujz  R 17-20:13:27      1 v100n01
```

1. 先看自己的任务，拿到 JOBID

2. 直接取消。无论sbatch、salloc、srun启动的任务的都能这么取消

scontrol

- 学习scontrol最好的方法就是看help

- https://slurm.schedmd.com/scontrol.html
- http://hpc.pku.edu.cn/_book/guide/slurm/sacct.html
- https://docs.hpc.sjtu.edu.cn/job/slurm.html#srun-salloc

`scontrol`: 查看和修改作业参数

Slurm	功能
scontrol show job JOB_ID	查看排队或正在运行的作业的信息
scontrol hold JOB_ID	暂停JOB_ID
scontrol release JOB_ID	恢复JOB_ID
scontrol update dependency=JOB_ID	添加作业依赖性，以便仅在JOB_ID完成后才开始作业
scontrol –help	查看所有选项

sacct

- 学习scontrol最好的方法就是看help

- https://slurm.schedmd.com/sacct.html
- http://hpc.pku.edu.cn/_book/guide/slurm/sacct.html
- https://docs.hpc.sjtu.edu.cn/job/slurm.html#srun-salloc

`sacct` 查看作业记录

Slurm	功能
sacct -l	查看详细的帐户作业信息
sacct –states=R	查看具有特定状态的作业的账号作业信息
sacct -S YYYY-MM-DD	在指定时间后选择处于任意状态的作业
sacct –format="LAYOUT"	使用给定的LAYOUT自定义sacct输出
sacct –help	查看所有选项

默认情况下，sacct显示过去 **24小时** 的账号作业信息。

一 使用集群计算资源的方式

二 集群硬件和软件栈介绍

三 GPU集群容器工具介绍与容器示例

四 交互式使用计算资源的详细步骤

五 集群使用注意事项和账户申请流程

学习enroot

```
yuezhifeng@dgxmgt01:~$ srun --time=02:00:00 --ntasks-per-node 1 --pty bash -i
yuezhifeng@a100n01:~$ enroot
Usage: enroot COMMAND [ARG...]

Command line utility for manipulating container sandboxes.

 Commands:
   batch  [options] [--] CONFIG [COMMAND] [ARG...]
   bundle [options] [--] IMAGE
   create [options] [--] IMAGE
   exec   [options] [--] PID COMMAND [ARG...]
   export [options] [--] NAME
   import [options] [--] URI
   list   [options]
   remove [options] [--] NAME...
   start  [options] [--] NAME|IMAGE [COMMAND] [ARG...]
   version
yuezhifeng@a100n01:~$ enroot start
Usage: enroot start [options] [--] NAME|IMAGE [COMMAND] [ARG...]

Start a container and invoke the command script within its root filesystem.
Command and arguments are passed to the script as input parameters.

In the absence of a command script and if a command was given, it will be executed directly.
Otherwise, an interactive shell will be started within the container.

 Options:
   -c, --conf CONFIG    Specify a configuration script to run before the container starts
   -e, --env KEY[=VAL]  Export an environment variable inside the container
       --rc SCRIPT      Override the command script inside the container
   -r, --root           Ask to be remapped to root inside the container
   -w, --rw             Make the container root filesystem writable
   -m, --mount FSTAB    Perform a mount from the host inside the container (colon-separated)
```

- https://github.com/NVIDIA/enroot
- https://www.pugetsystems.com/labs/hpc/Run-Docker-Containers-with-NVIDIA-Enroot

学习enroot import

```
yuezhifeng@a100n01:~$ enroot import
Usage: enroot import [options] [--] URI

Import a container image from a specific location.

 Schemes:
   docker://[USER@][REGISTRY#]IMAGE[:TAG]  Import a Docker image from a registry
   dockerd://IMAGE[:TAG]                  Import a Docker image from the Docker daemon

 Options:
   -a, --arch    Architecture of the image (defaults to host architecture)
   -o, --output  Name of the output image file (defaults to "URI.sqsh")
```

```
yuezhifeng@a100n01:~$ enroot import docker://ubuntu
[INFO] Querying registry for permission grant
[INFO] Authenticating with user: <anonymous>
[INFO] Authentication succeeded
[INFO] Fetching image manifest list
[INFO] Fetching image manifest
[INFO] Downloading 2 missing layers...

100% 2:0=0s 35807b77a593c1147d13dc926a91dcc3015616ff7307cc30442c5a8e07546283

[INFO] Extracting image layers...

100% 1:0=0s 35807b77a593c1147d13dc926a91dcc3015616ff7307cc30442c5a8e07546283

[INFO] Converting whiteouts...

100% 1:0=0s 35807b77a593c1147d13dc926a91dcc3015616ff7307cc30442c5a8e07546283

[INFO] Creating squashfs filesystem...

Parallel mksquashfs: Using 1 processor
Creating 4.0 filesystem on /OceanStor100D/home/yzf_cdc/yuezhifeng/ubuntu.sqsh, block size 131072.
[=============================================================================================================

Exportable Squashfs 4.0 filesystem, gzip compressed, data block size 131072
        uncompressed data, uncompressed metadata, uncompressed fragments,
        uncompressed xattrs, uncompressed ids
        duplicates are not removed
Filesystem size 71234.87 Kbytes (69.57 Mbytes)
        99.96% of uncompressed filesystem size (71266.81 Kbytes)
Inode table size 107494 bytes (104.97 Kbytes)
        100.00% of uncompressed inode table size (107494 bytes)
Directory table size 64760 bytes (63.24 Kbytes)
        100.00% of uncompressed directory table size (64760 bytes)
No duplicate files removed
Number of inodes 3264
Number of files 2502
Number of fragments 288
Number of symbolic links  184
Number of device nodes 0
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 578
Number of ids (unique uids + gids) 1
Number of uids 1
        root (0)
Number of gids 1
        root (0)
```

拉取docker镜像

并保存成了enroot镜像格式

学习enroot start

```
yuezhifeng@a100n01:~$ enroot start ubuntu.sqsh cat /etc/passwd | wc
     19      24     945
yuezhifeng@a100n01:~$ cat /etc/passwd | wc
     46      73    2539
```

容器文件内容和物理机文件内容是不一样的

测试用enroot跑alphafold2：脚本

```
yuezhifeng@a100n01:~$ cat run_test_alphafold2
enroot start -r -w -m /OceanStor100D/apps_gpu/alphafold/:/OceanStor100D/apps_gpu/alphafold/ \
                   -m ./test_enroot_alphafold2:/test_enroot_alphafold2 \
                   /OceanStor100D/apps_gpu/alphafold/alphafold.sqsh \
                   -bfd_database_path=/OceanStor100D/apps_gpu/alphafold//bfd/bfd_metaclust_clu_complete_id30_c90_final_seq.sorted_opt \
                   --mgnify_database_path=/OceanStor100D/apps_gpu/alphafold//mgnify/mgy_clusters.fa \
                   --template_mmcif_dir=/OceanStor100D/apps_gpu/alphafold//pdb_mmcif/mmcif_files \
                   --obsolete_pdbs_path=/OceanStor100D/apps_gpu/alphafold//pdb_mmcif/obsolete.dat \
                   --pdb70_database_path=/OceanStor100D/apps_gpu/alphafold//pdb70/pdb70 \
                   --uniclust30_database_path=/OceanStor100D/apps_gpu/alphafold//uniclust30/uniclust30_2018_08/uniclust30_2018_08 \
                   --uniref90_database_path=/OceanStor100D/apps_gpu/alphafold//uniref90/uniref90.fasta \
                   --data_dir=/OceanStor100D/apps_gpu/alphafold/ \
                   --output_dir=/test_enroot_alphafold2/ \
                   --fasta_paths=/test_enroot_alphafold2/query_example.fasta \
                   --model_names=model_1 \
                   --max_template_date=2020-05-14 \
                   --preset=full_dbs \
                   --benchmark=false \
                   --logtostderr
```

映射（挂载）文件夹到容器

待启动的容器镜像

输入输出的指定

测试用enroot跑alphafold2：启动

```
yuezhifeng@a100n01:~$ bash run_test_alphafold2
/opt/conda/lib/python3.7/site-packages/absl/flags/_validators.py:206: UserWarning: Flag --preset has a non-None defa
ult value; therefore, mark_flag_as_required will pass even if flag is not specified in the command line!
  'command line!' % flag_name)
I0922 21:37:17.402646 139762730841920 templates.py:837] Using precomputed obsolete pdbs /OceanStor100D/apps_gpu/alph
afold//pdb_mmcif/obsolete.dat.
I0922 21:37:17.546205 139762730841920 tpu_client.py:54] Starting the local TPU driver.
I0922 21:37:17.547806 139762730841920 xla_bridge.py:214] Unable to initialize backend 'tpu_driver': Not found: Unabl
e to find driver in registry given worker: local://
I0922 21:37:17.752825 139762730841920 xla_bridge.py:214] Unable to initialize backend 'tpu': Invalid argument: TpuPl
atform is not available.
I0922 21:37:18.637268 139762730841920 run_alphafold.py:261] Have 1 models: ['model_1']
I0922 21:37:18.637454 139762730841920 run_alphafold.py:273] Using random seed 4787518950948357729 for the data pipel
ine
I0922 21:37:18.639081 139762730841920 jackhmmer.py:130] Launching subprocess "/usr/bin/jackhmmer -o /dev/null -A /tm
p/tmpm6_i6r3y/output.sto --noali --F1 0.0005 --F2 5e-05 --F3 5e-07 --incE 0.0001 -E 0.0001 --cpu 8 -N 1 /test_enroot
_alphafold2/query_example.fasta /OceanStor100D/apps_gpu/alphafold//uniref90/uniref90.fasta"
I0922 21:37:18.673105 139762730841920 utils.py:36] Started Jackhmmer (uniref90.fasta) query
```

alphafold先寻找TPU

测试用enroot跑alphafold2：查看GPU是否使用

```
PS C:\Users\yuezhifeng> ssh yuezhifeng@gpuc.cebsit.ac.cn
Password:
Welcome to NVIDIA DGX Unknown Platform Version 5.0.2 (GNU/Linux 5.4.0-80-generic x86_64)

  System information as of Wed 22 Sep 2021 09:41:17 PM CST

  System load:              10.35
  Usage of /:               20.6% of 271.65GB
  Memory usage:             7%
  Swap usage:               0%
  Temperature:              44.0 C
  Processes:                1033
  Users logged in:          6
  IPv4 address for docker0: 172.17.0.1
  IPv4 address for eno1:    172.16.1.23
  IPv4 address for eno4:    172.16.102.23
  IPv6 address for eno4:    2400:dd02:1011:1006:38b3:f18c:5e5:e81
  IPv4 address for ibs4:    172.16.2.23
  IPv4 address for tunl0:   10.233.116.0

Health of this system could not be determined. Please use 'sudo nvsm show alerts' to see any al
 have.

Last login: Wed Sep 22 21:39:10 2021 from 10.10.48.196
yuezhifeng@dgxmgt01:~$ ssh a100n01
Welcome to NVIDIA DGX Server Version 5.0.5 (GNU/Linux 5.4.0-80-generic x86_64)

  System information as of Wed 22 Sep 2021 09:41:21 PM CST

  System load:                4.18
  Usage of /:                 11.0% of 1.72TB
  Memory usage:               2%
  Swap usage:                 0%
  Processes:                  2578
  Users logged in:            2
  IPv4 address for docker0:   172.17.0.1
  IPv4 address for enp226s0:  172.16.1.212
  IPv4 address for enp97s0f0: 172.16.102.212
  IPv6 address for enp97s0f0: 2400:dd02:1011:1006:e744:7c94:651:1f7f
  IPv4 address for ibp225s0f0: 172.16.2.212
  IPv4 address for tunl0:     10.233.90.0

The system has 0 critical alerts and 8 warnings. Use 'sudo nvsm show alerts' for more details.

Last login: Wed Sep 22 21:39:15 2021 from 172.16.1.23
yuezhifeng@a100n01:~$ nvidia-smi
Wed Sep 22 21:41:25 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.142.00   Driver Version: 450.142.00   CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| 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  A100-SXM4-40GB      On   | 00000000:07:00.0 Off |                    0 |
| N/A   32C    P0    59W / 400W |  36526MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A   3810580      C   python                          36523MiB |
+-----------------------------------------------------------------------------+
yuezhifeng@a100n01:~$
```

```
yuezhifeng@a100n01:~$ nvidia-smi
Wed Sep 22 21:41:25 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.142.00   Driver Version: 450.142.00   CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| 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  A100-SXM4-40GB      On   | 00000000:07:00.0 Off |                    0 |
| N/A   32C    P0    59W / 400W |  36526MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A   3810580      C   python                          36523MiB |
+-----------------------------------------------------------------------------+
```

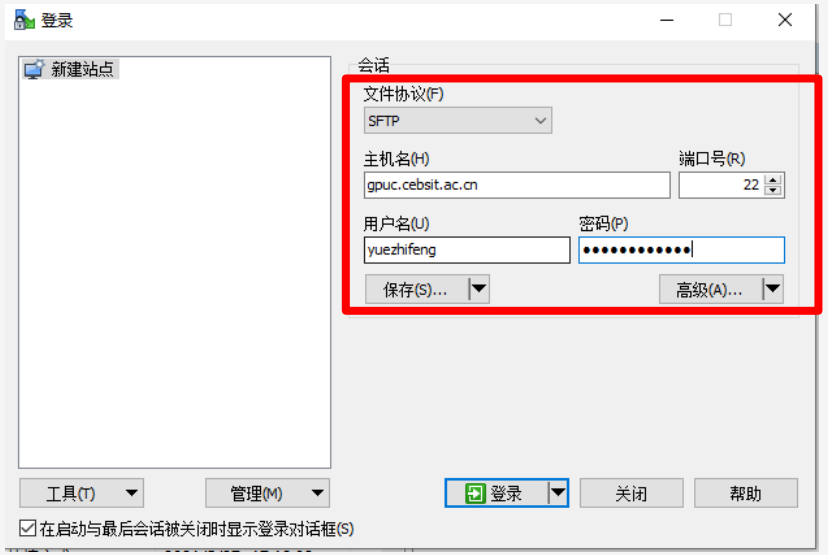
数据已经到显存

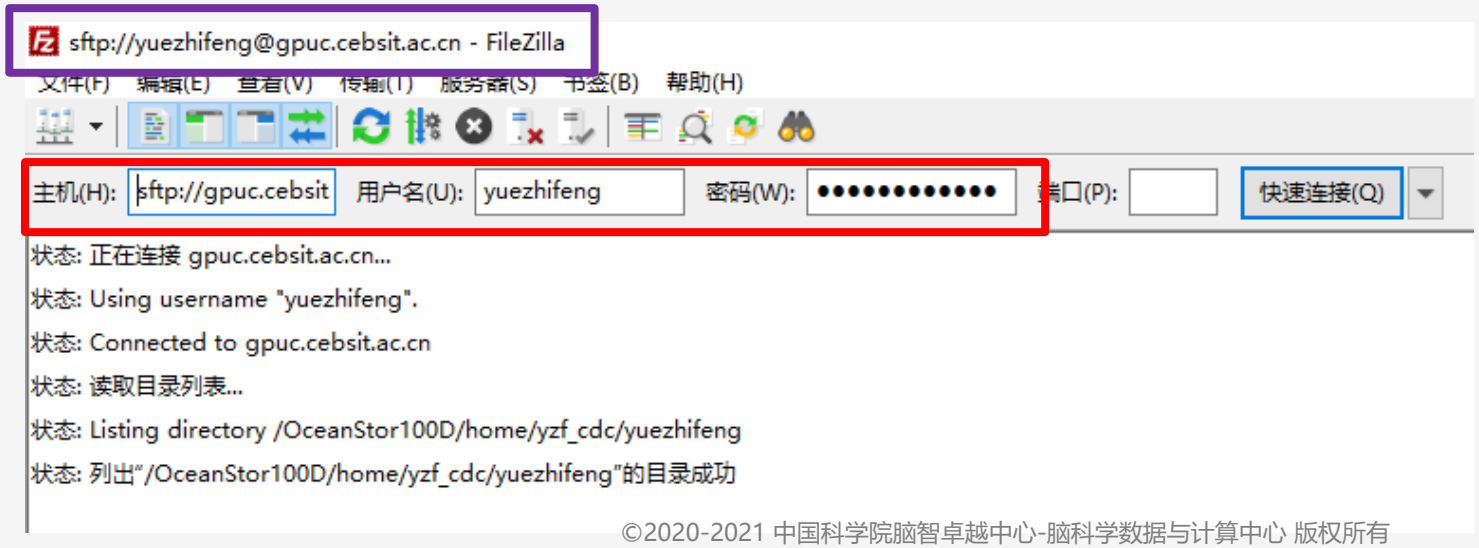
测试用enroot跑alphafold2：查看GPU是否使用

```
Wed Sep 22 22:06:17 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.142.00   Driver Version: 450.142.00   CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| 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  A100-SXM4-40GB      On   | 00000000:07:00.0 Off |                    0 |
| N/A   50C    P0   215W / 400W |  37354MiB / 40537MiB |     99%      Default |
|        3          262         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A   3810580      C   python                           37351MiB |
|    0   N/A  N/A   3810580------C   python-------------------------37351MiB |
+-----------------------------------------------------------------------------+
```

GPU计算单元也用起来了

SFTP方式下载结果数据（WinSCP or FileZilla）





一 使用集群计算资源的方式

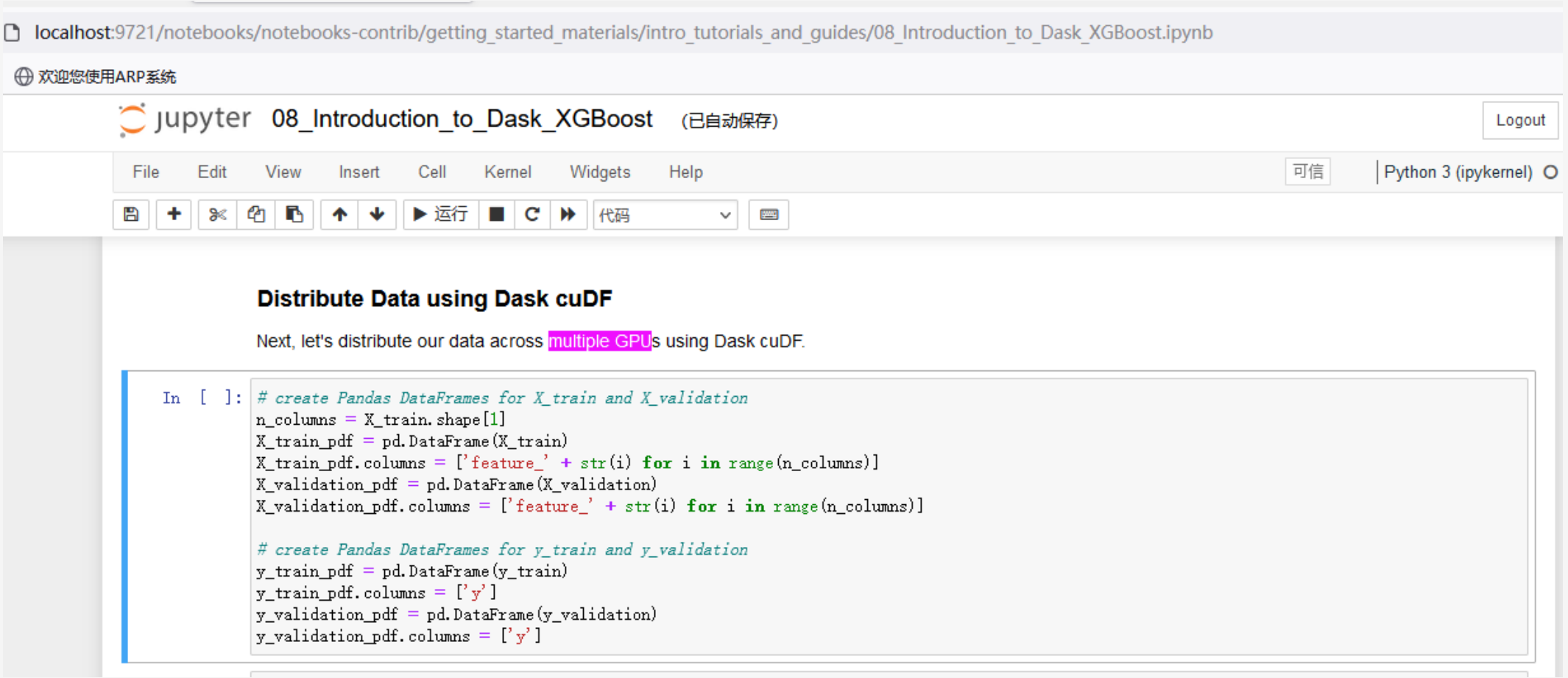
二 集群硬件和软件栈介绍

三 GPU集群容器工具介绍与容器示例

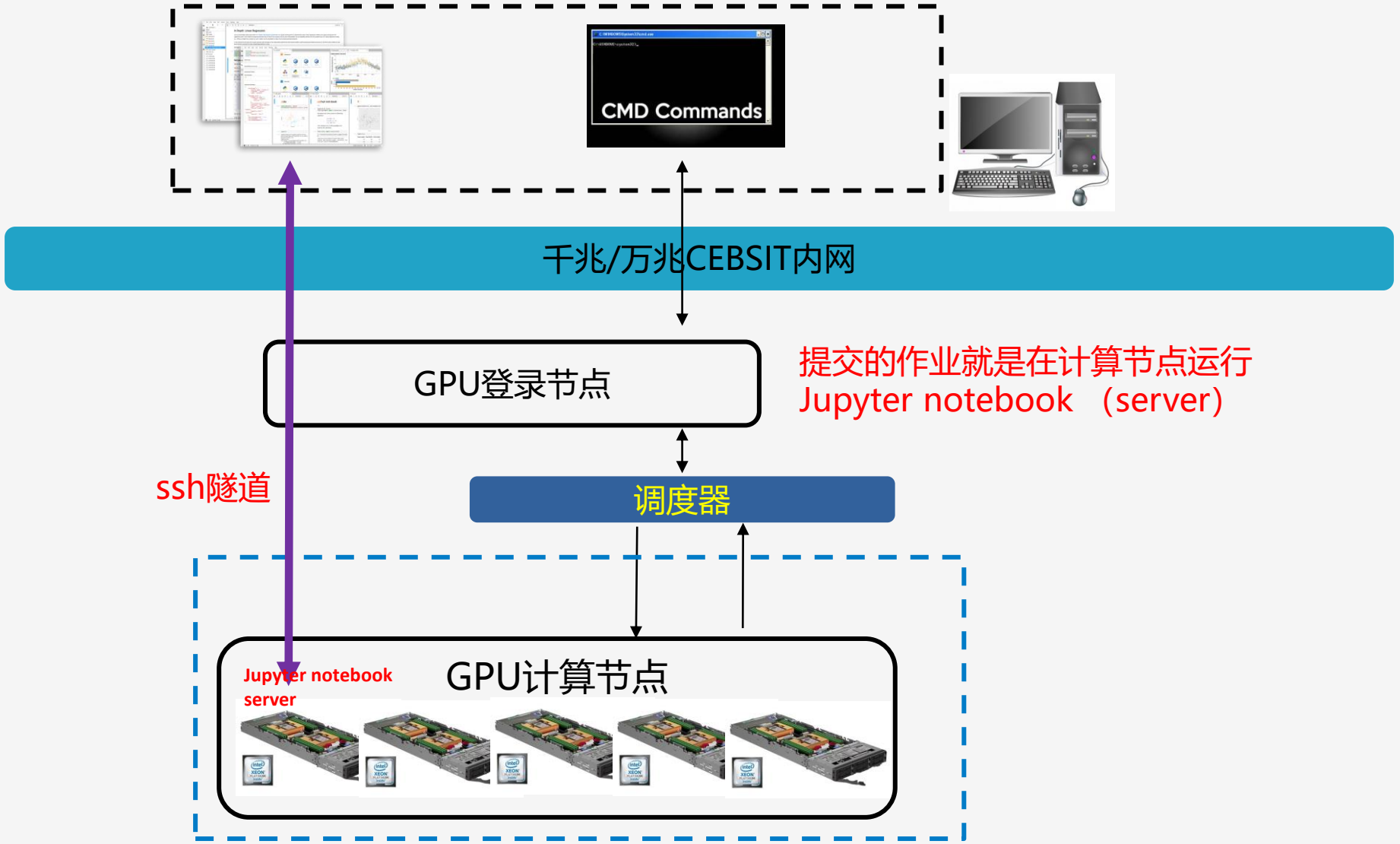
四 交互式使用计算资源的详细步骤

五 集群使用注意事项和账户申请流程

在Jupyter中进行Rapids和Dask编程



通过Slurm在计算节点启动Jupyter Notebook



下载和安装miniconda

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\yuezhifeng> ssh yuezhifeng@gpuc.cebsit.ac.cn
Password:
Welcome to NVIDIA DGX Unknown Platform Version 5.0.2 (GNU/Linux 5.4.0-80-generic x86_64)

  System information as of Thu 23 Sep 2021 10:06:15 PM CST

  System load:              17.73
  Usage of /:               20.6% of 271.65GB
  Memory usage:             7%
  Swap usage:               0%
  Temperature:              52.0 C
  Processes:                924
  Users logged in:          5
  IPv4 address for docker0: 172.17.0.1
  IPv4 address for eno1:    172.16.1.23
  IPv4 address for eno4:    172.16.102.23
  IPv6 address for eno4:    2400:dd02:1011:1006:38b3:f18c:5e5:e81
  IPv4 address for ibs4:    172.16.2.23
  IPv4 address for tunl0:   10.233.116.0

Health of this system could not be determined. Please use 'sudo nvsm show alerts' to see any alerts the system might
 have.

Last login: Thu Sep 23 21:43:33 2021 from 10.10.48.196
yuezhifeng@dgxmgt01:~$ wget -c https://repo.anaconda.com/miniconda/Miniconda3-py39_4.10.3-Linux-x86_64.sh
--2021-09-23 22:06:30--  https://repo.anaconda.com/miniconda/Miniconda3-py39_4.10.3-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 2606:4700::6810:8303, 2606:4700::6810:8203, 104.16.131.3, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|2606:4700::6810:8303|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 66709754 (64M) [application/x-sh]
Saving to: 'Miniconda3-py39_4.10.3-Linux-x86_64.sh'

Miniconda3-py39_4.10.3-Linux 100%[=============================================>]  63.62M  4.32MB/s    in 13s

2021-09-23 22:06:44 (4.91 MB/s) - 'Miniconda3-py39_4.10.3-Linux-x86_64.sh' saved [66709754/66709754]

yuezhifeng@dgxmgt01:~$ bash ./Miniconda3-py39_4.10.3-Linux-x86_64.sh

Welcome to Miniconda3 py39_4.10.3

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
==================================
End User License Agreement - Miniconda
==================================
```

activate的方式选择

```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> no

You have chosen to not have conda modify your shell scripts at all.
To activate conda's base environment in your current shell session:

eval "$(/OceanStor100D/home/yzf_cdc/yuezhifeng/miniconda3/bin/conda shell.YOUR_SHELL_NAME hook)"

To install conda's shell functions for easier access, first activate, then:

conda init

If you'd prefer that conda's base environment not be activated on startup,
   set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniconda3!
yuezhifeng@dgxmgt01:~$ eval "$(/OceanStor100D/home/yzf_cdc/yuezhifeng/miniconda3/bin/conda shell.bash hook)"
(base) yuezhifeng@dgxmgt01:~$
```

推荐选 no

用conda安装rapids、dask和jupyterlab

```
(base) yuezhifeng@dgxmgt01:~$ conda create -n rapids-21.08 -c rapidsai -c nvidia -c conda-forge \
>                              rapids=21.08 python=3.7 cudatoolkit=11.0 \
>                              jupyterlab dask dask-xgboost
```

指定软件包的来源、版本等

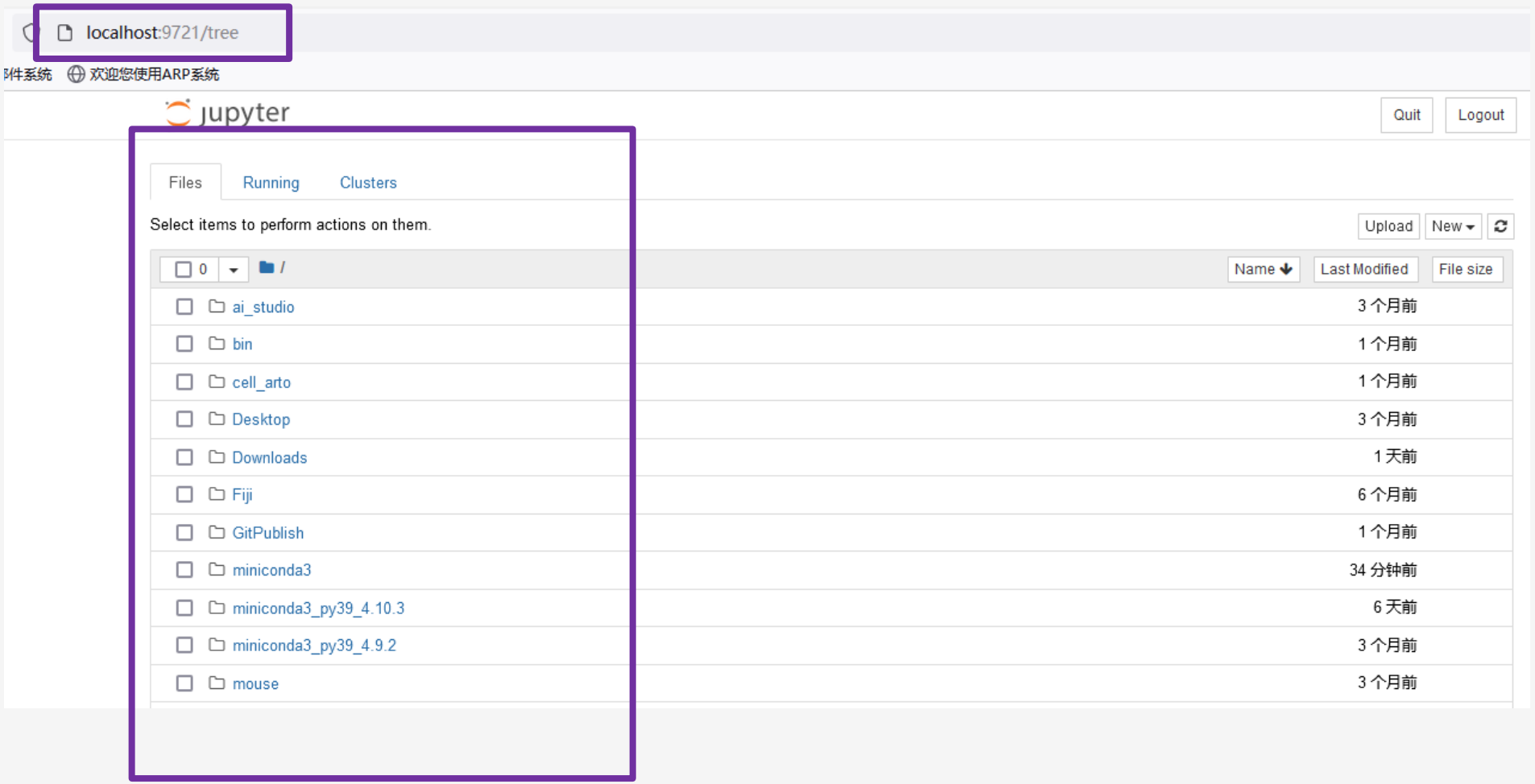
启动jupyter server示例

```
yuezhifeng@dgxmgt01:~$ sinfo -O cpusstate,nodehost,gres,gresused
CPUS(A/I/O/T)       HOSTNAMES           GRES                GRES_USED
8/248/0/256         a100n01             gpu:8               gpu:(null):2(IDX:0-1
32/224/0/256        a100n02             gpu:8               gpu:(null):8(IDX:0-7
32/224/0/256        a100n03             gpu:8               gpu:(null):8(IDX:0-7
144/112/0/256       a100n04             gpu:8               gpu:(null):8(IDX:0-7
40/0/0/40           v100n01             gpu:8(S:0-1)        gpu:(null):8(IDX:0-7
yuezhifeng@dgxmgt01:~$ srun -p jupiter -w a100n01 --time=24:00:00 --ntasks-per-node 128 --gres=gpu:8 --pty bash -i
srun: job 2765 queued and waiting for resources
srun: job 2765 has been allocated resources
yuezhifeng@a100n01:~$ eval "$(/OceanStor100D/home/yzf_cdc/yuezhifeng/miniconda3_py39_4.10.3/bin/conda shell.bash hook)"
(base) yuezhifeng@a100n01:~$ conda activate rapids-21.08
(rapids-21.08) yuezhifeng@a100n01:~$ jupyter-notebook --no-browser --port=9721 --ip=$hostname
[W 22:38:40.872 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[W 2021-09-23 22:38:49.080 LabApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-09-23 22:38:49.080 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-09-23 22:38:49.080 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-09-23 22:38:49.080 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2021-09-23 22:38:49.087 LabApp] JupyterLab extension loaded from /OceanStor100D/home/yzf_cdc/yuezhifeng/miniconda3_py39_4.10.3/envs/rapids-21.08/lib/python3.7/site-packages/jupyterlab
[I 2021-09-23 22:38:49.087 LabApp] JupyterLab application directory is /OceanStor100D/home/yzf_cdc/yuezhifeng/miniconda3_py39_4.10.3/envs/rapids-21.08/share/jupyter/lab
[I 22:38:49.093 NotebookApp] Serving notebooks from local directory: /OceanStor100D/home/yzf_cdc/yuezhifeng
[I 22:38:49.093 NotebookApp] Jupyter Notebook 6.4.4 is running at:
[I 22:38:49.093 NotebookApp] http://a100n01:9721/?token=89ece75aea02aa78846dd70e371b6356279873774dc5ad05
[I 22:38:49.093 NotebookApp]  or http://127.0.0.1:9721/?token=89ece75aea02aa78846dd70e371b6356279873774dc5ad05
[I 22:38:49.093 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 22:38:49.121 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///OceanStor100D/home/yzf_cdc/yuezhifeng/.local/share/jupyter/runtime/nbserver-3532710-open.html
    Or copy and paste one of these URLs:
        http://a100n01:9721/?token=89ece75aea02aa78846dd70e371b6356279873774dc5ad05
     or http://127.0.0.1:9721/?token=89ece75aea02aa78846dd70e371b6356279873774dc5ad05
```

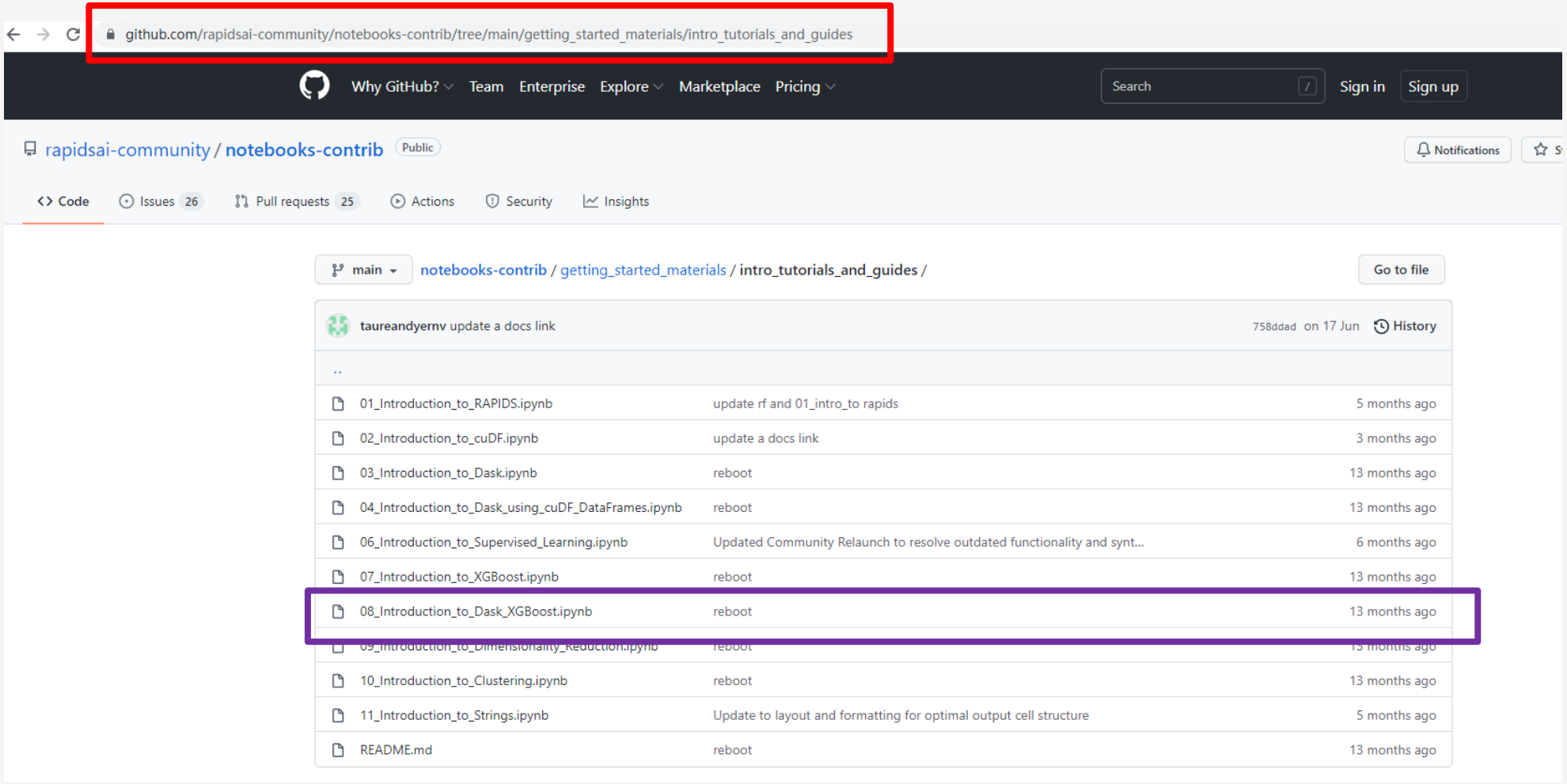
建立ssh隧道

```
PS C:\Users\yuezhifeng> ssh -t -t yuezhifeng@gpuc.cebsit.ac.cn -L 9721:localhost:9721 ssh a100n01 -L 9721:localhost:9721
Password:
bind [::1]:9721: Cannot assign requested address
Welcome to NVIDIA DGX Server Version 5.0.5 (GNU/Linux 5.4.0-80-generic x86_64)

  System information as of Fri 24 Sep 2021 01:57:41 PM CST

  System load:                31.34
  Usage of /:                 11.4% of 1.72TB
  Memory usage:               4%
  Swap usage:                 0%
  Processes:                  2577
  Users logged in:            1
  IPv4 address for docker0:   172.17.0.1
  IPv4 address for enp226s0:  172.16.1.212
  IPv4 address for enp97s0f0: 172.16.102.212
  IPv6 address for enp97s0f0: 2400:dd02:1011:1006:e744:7c94:651:1f7f
  IPv4 address for ibp225s0f0: 172.16.2.212
  IPv4 address for tunl0:     10.233.90.0

The system has 0 critical alerts and 8 warnings. Use 'sudo nvsm show alerts' for more details.

yuezhifeng@a100n01:~$
```

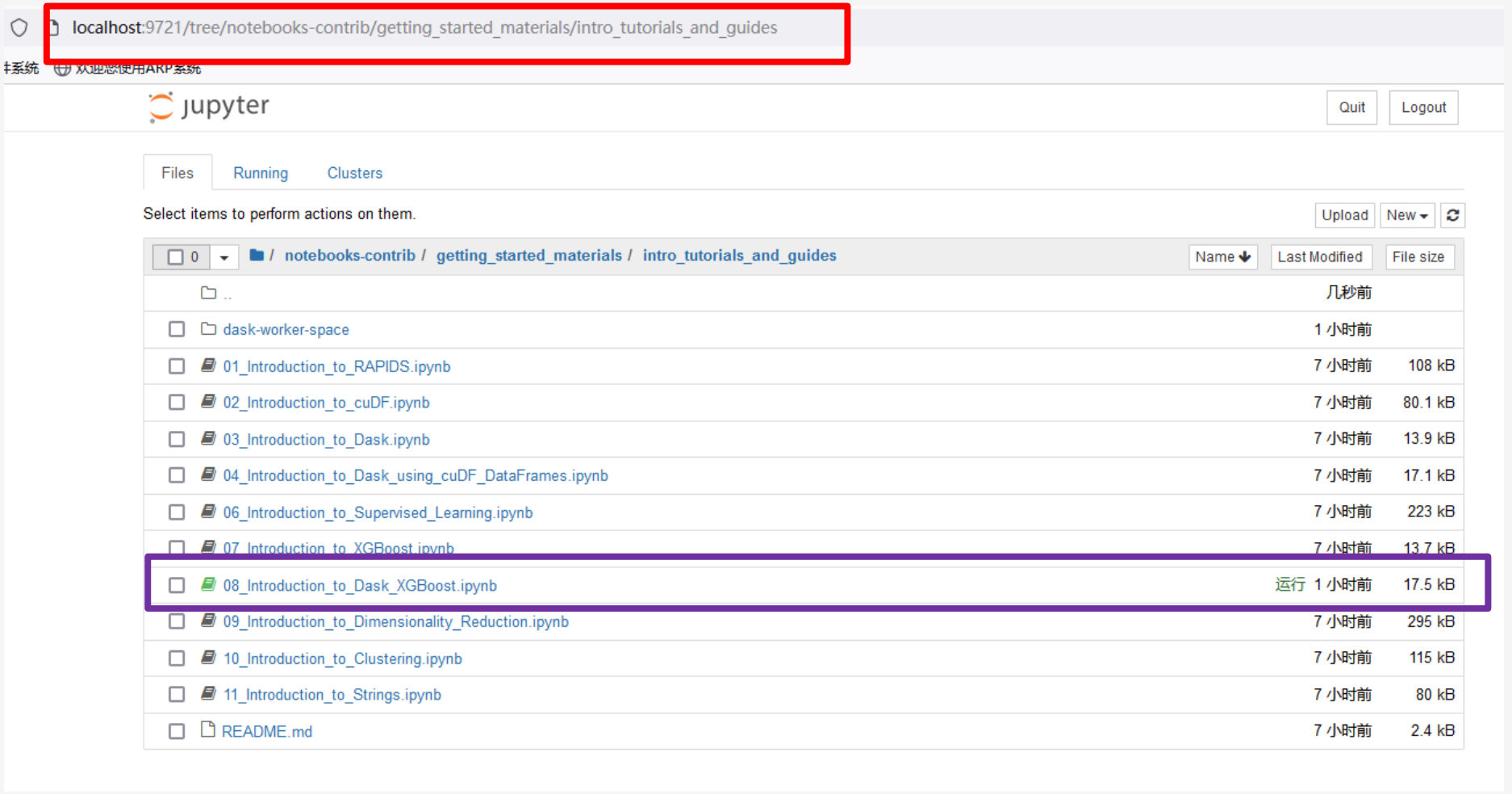
浏览器连接jupyter server成功



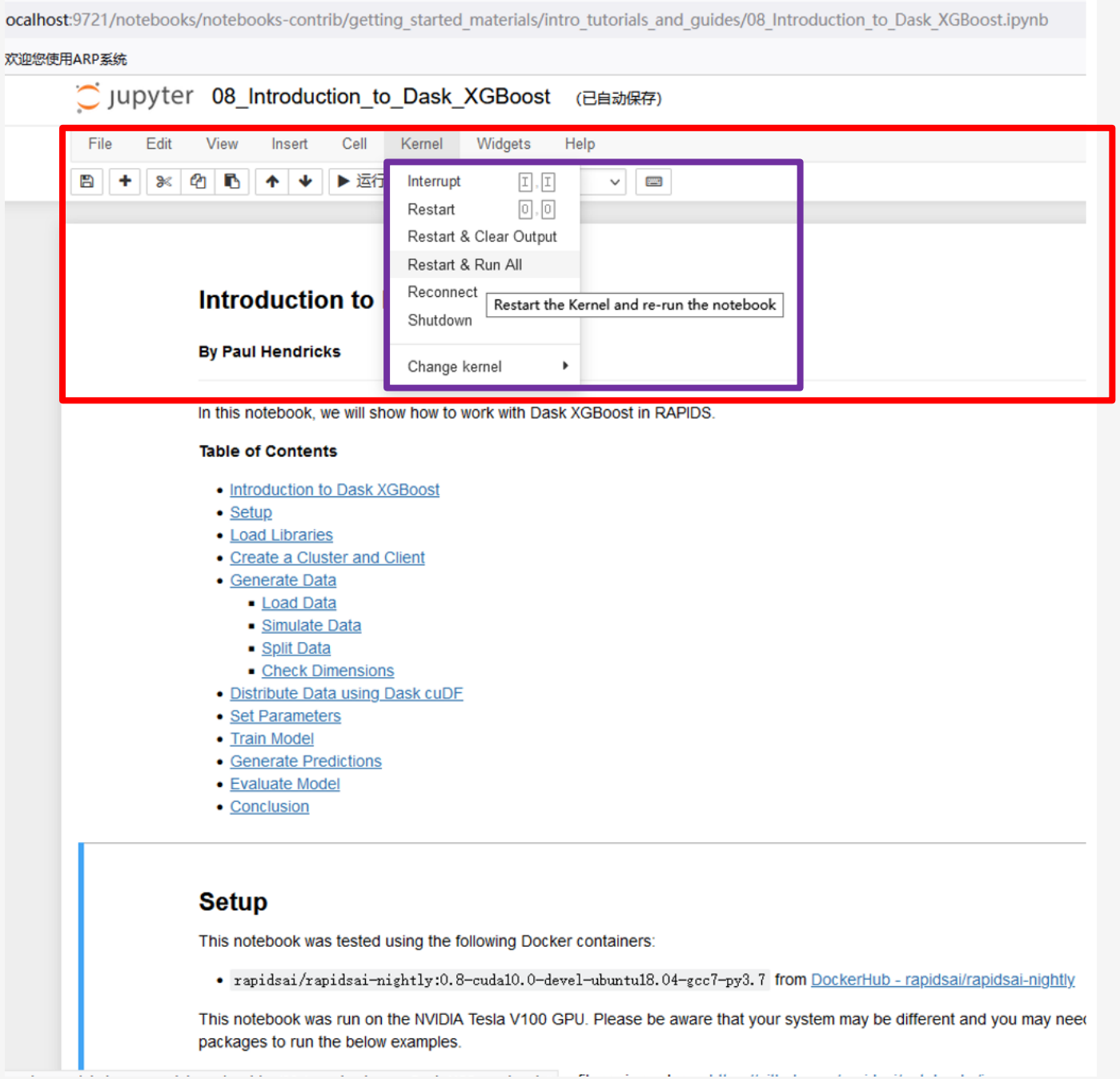
An example



成功打开运行在GPU节点上的notebook



连接成功



成功载入Dask和Rapids的库

Load Libraries

Let's load some of the libraries within the RAPIDs ecosystem and see which versions we have.

In [3]:
```
import cudf; print('cuDF Version:', cudf.__version__)
import dask; print('Dask Version:', dask.__version__)
import dask_cudf; print('Dask cuDF Version:', dask_cudf.__version__)
import dask_xgboost; print('Dask XGBoost Version:', dask_xgboost.__version__)
import numpy as np; print('numpy Version:', np.__version__)
import pandas as pd; print('pandas Version:', pd.__version__)
import sklearn; print('Scikit-Learn Version:', sklearn.__version__)
# import xgboost as xgb; print('XGBoost Version:', xgb.__version__)
```

```
cuDF Version: 21.08.03
Dask Version: 2021.07.1
Dask cuDF Version: 21.08.03
Dask XGBoost Version: 0.1.5
numpy Version: 1.21.2
pandas Version: 1.2.5
Scikit-Learn Version: 0.24.2
```

Dask Cluster的建立

Create a Cluster and Client

Let's start by creating a local cluster of workers and a client to interact with that cluster.

In [4]:
```
from dask.distributed import Client
from dask_cuda import LocalCUDACluster


# create a local CUDA cluster
cluster = LocalCUDACluster()
client = Client(cluster)
client
```

```
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
distributed.preloading - INFO - Import preload module: dask_cuda.initialize
```

Out[4]:

Client

Client-7bff6134-1c80-11ec-990a-5cff[illegible]5e22539

Connection method: Cluster object

Cluster type: LocalCUDACluster

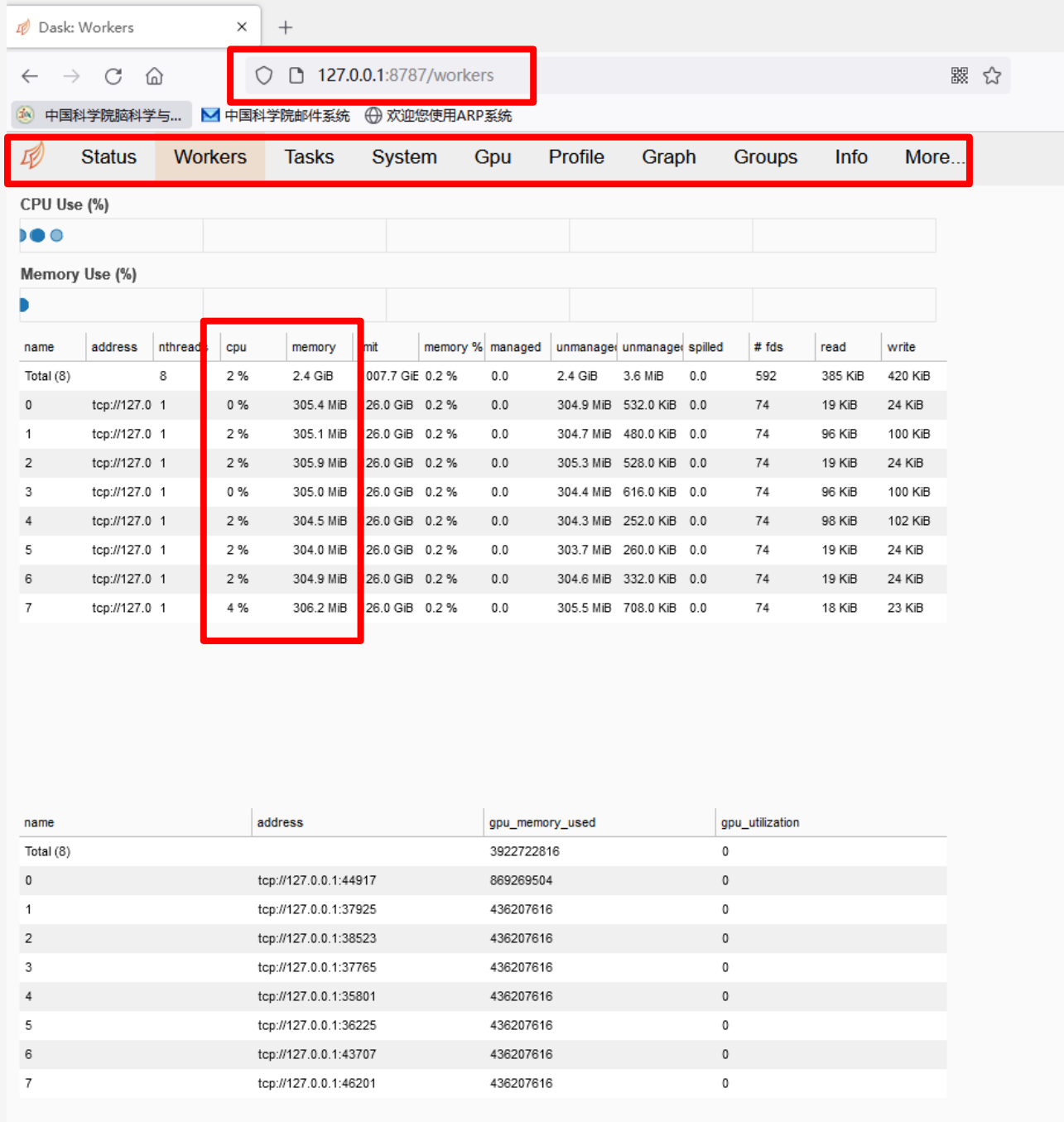
Dashboard: http://127.0.0.1:8787/status

Cluster Info

建立新的ssh隧道来访问Dask Cluster

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\yuezhifeng> ssh -t -t yuezhifeng@gpuc.cebsit.ac.cn -L 8787:localhost:8787 ssh a100n01 -L 8787:localhost:8787
Password:
bind [::1]:8787: Cannot assign requested address
Welcome to NVIDIA DGX Server Version 5.0.5 (GNU/Linux 5.4.0-80-generic x86_64)

  System information as of Thu 23 Sep 2021 11:12:14 PM CST

  System load:                  3.39
  Usage of /:                   11.3% of 1.72TB
  Memory usage:                 2%
  Swap usage:                   0%
  Processes:                    2586
  Users logged in:              2
  IPv4 address for docker0:     172.17.0.1
  IPv4 address for enp226s0:    172.16.1.212
  IPv4 address for enp97s0f0:   172.16.102.212
  IPv6 address for enp97s0f0:   2400:dd02:1011:1006:e744:7c94:651:1f7f
  IPv4 address for ibp225s0f0: 172.16.2.212
  IPv4 address for tunl0:       10.233.90.0

The system has 0 critical alerts and 8 warnings. Use 'sudo nvsm show alerts' for more details.

Last login: Thu Sep 23 22:42:20 2021 from 172.16.1.23
yuezhifeng@a100n01:~$
```

成功访问到Dask Status



name	address	nthread	cpu	memory	mit	memory %	managed	unmanage	unmanage	spilled	# fds	read	write
Total (8)		8	2 %	2.4 GiB	007.7 GiE	0.2 %	0.0	2.4 GiB	3.6 MiB	0.0	592	385 KiB	420 KiB
0	tcp://127.0	1	0 %	305.4 MiB	26.0 GiB	0.2 %	0.0	304.9 MiB	532.0 KiB	0.0	74	19 KiB	24 KiB
1	tcp://127.0	1	2 %	305.1 MiB	26.0 GiB	0.2 %	0.0	304.7 MiB	480.0 KiB	0.0	74	96 KiB	100 KiB
2	tcp://127.0	1	2 %	305.9 MiB	26.0 GiB	0.2 %	0.0	305.3 MiB	528.0 KiB	0.0	74	19 KiB	24 KiB
3	tcp://127.0	1	0 %	305.0 MiB	26.0 GiB	0.2 %	0.0	304.4 MiB	616.0 KiB	0.0	74	96 KiB	100 KiB
4	tcp://127.0	1	2 %	304.5 MiB	26.0 GiB	0.2 %	0.0	304.3 MiB	252.0 KiB	0.0	74	98 KiB	102 KiB
5	tcp://127.0	1	2 %	304.0 MiB	26.0 GiB	0.2 %	0.0	303.7 MiB	260.0 KiB	0.0	74	19 KiB	24 KiB
6	tcp://127.0	1	2 %	304.9 MiB	26.0 GiB	0.2 %	0.0	304.6 MiB	332.0 KiB	0.0	74	19 KiB	24 KiB
7	tcp://127.0	1	4 %	306.2 MiB	26.0 GiB	0.2 %	0.0	305.5 MiB	708.0 KiB	0.0	74	18 KiB	23 KiB

name	address	gpu_memory_used	gpu_utilization
Total (8)		3922722816	0
0	tcp://127.0.0.1:44917	869269504	0
1	tcp://127.0.0.1:37925	436207616	0
2	tcp://127.0.0.1:38523	436207616	0
3	tcp://127.0.0.1:37765	436207616	0
4	tcp://127.0.0.1:35801	436207616	0
5	tcp://127.0.0.1:36225	436207616	0
6	tcp://127.0.0.1:43707	436207616	0
7	tcp://127.0.0.1:46201	436207616	0

“分布”数据到多GPU

jupyter 08_Introduction_to_Dask_XGBoost (已自动保存)

File Edit View Insert Cell Kernel Widgets Help

运行 Markdown

Distribute Data using Dask cuDF

Next, let's distribute our data across multiple GPUs using Dask cuDF.

In [11]:
```
# create Pandas DataFrames for X_train and X_validation
n_columns = X_train.shape[1]
X_train_pdf = pd.DataFrame(X_train)
X_train_pdf.columns = ['feature_' + str(i) for i in range(n_columns)]
X_validation_pdf = pd.DataFrame(X_validation)
X_validation_pdf.columns = ['feature_' + str(i) for i in range(n_columns)]

# create Pandas DataFrames for y_train and y_validation
y_train_pdf = pd.DataFrame(y_train)
y_train_pdf.columns = ['y']
y_validation_pdf = pd.DataFrame(y_validation)
y_validation_pdf.columns = ['y']
```

In [12]:
```
# Dask settings
npartitions = 8

# create Dask DataFrames for X_train and X_validation
X_train_dask_pdf = dask.dataframe.from_pandas(X_train_pdf, npartitions=npartitions)
X_validation_dask_pdf = dask.dataframe.from_pandas(X_validation_pdf, npartitions=npartitions)

# create Dask cuDF DataFrames for X_train and X_validation
X_train_dask_cudf = dask_cudf.from_dask_dataframe(X_train_dask_pdf)
X_validation_dask_cudf = dask_cudf.from_dask_dataframe(X_validation_dask_pdf)

# create Dask DataFrames for y_train and y_validation
y_train_dask_pdf = dask.dataframe.from_pandas(y_train_pdf, npartitions=npartitions)
y_validation_dask_pdf = dask.dataframe.from_pandas(y_validation_pdf, npartitions=npartitions)

# create Dask cuDF DataFrames for y_train and y_validation
y_train_dask_cudf = dask_cudf.from_dask_dataframe(y_train_dask_pdf)
y_validation_dask_cudf = dask_cudf.from_dask_dataframe(y_validation_dask_pdf)
```

In [13]:
```
# Optional: persist training and validation data into memory
X_train_dask_cudf = X_train_dask_cudf.persist()
X_validation_dask_cudf = X_validation_dask_cudf.persist()
y_train_dask_cudf = y_train_dask_cudf.persist()
y_validation_dask_cudf = y_validation_dask_cudf.persist()
```

多GPU训练模型

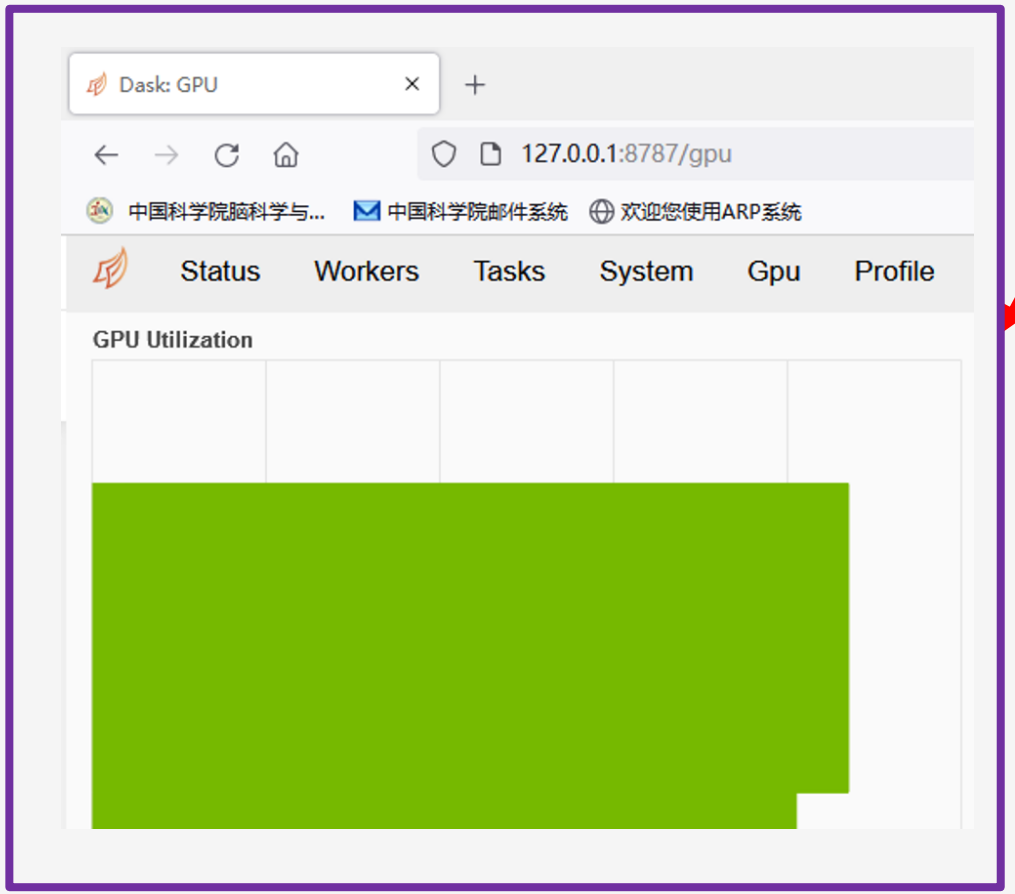
Train Model

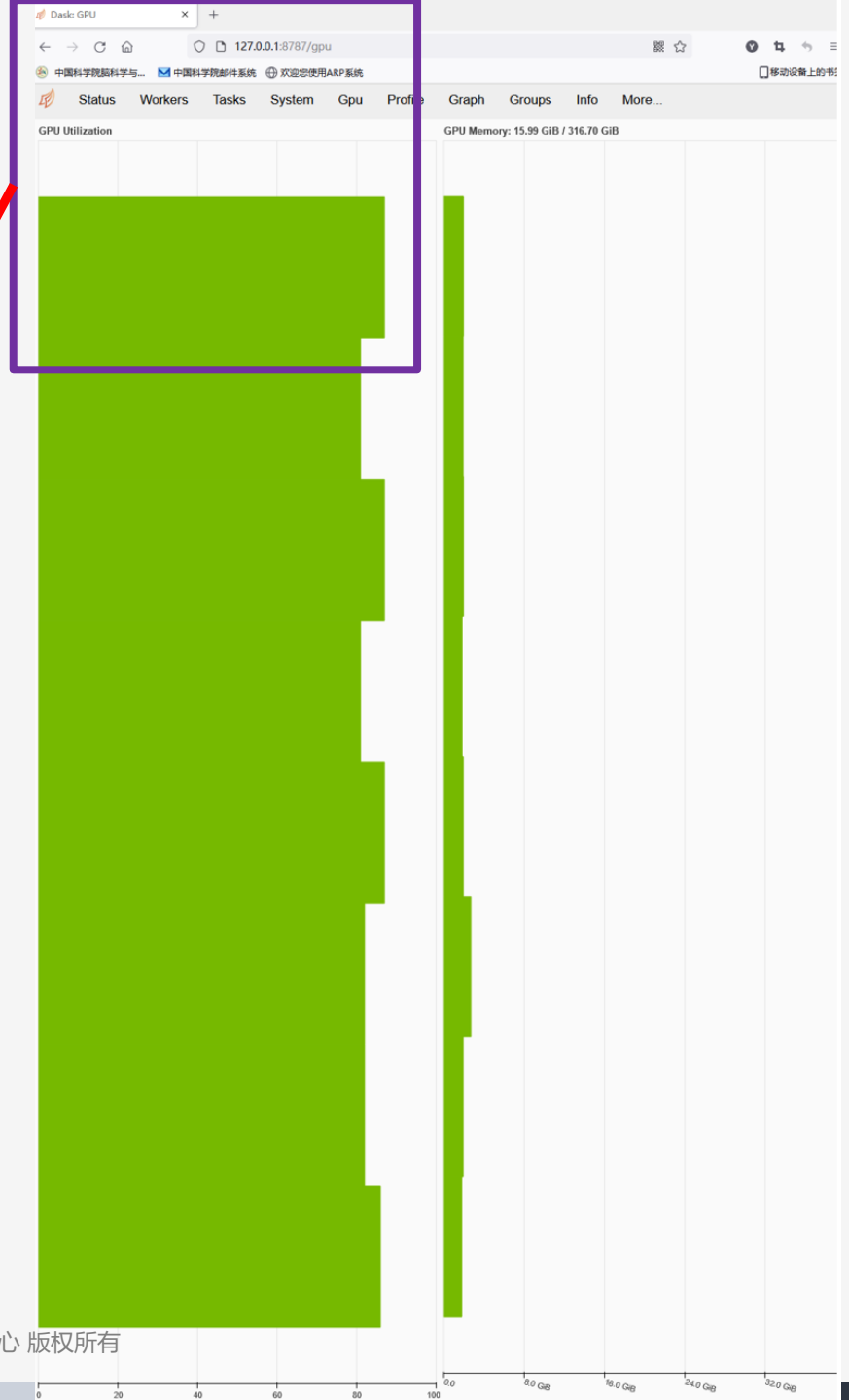
Now it's time to train our model! We can use the `dask_xgboost.train` function and pass in the parameters, training dataset, the number of boosting iterations, and the list of items to be evaluated during training.

In [15]:
```
# model training settings
num_round = 100
```

In [16]:
```
%%time


bst = dask_xgboost.train(client, params, X_train_dask_cudf, y_train_dask_cudf, num_boost_round=num_round)
```

观察GPU的利用率





测试结束

```
Accuracy: 0.98312
```

Conclusion

In this notebook, we showed how to work with Dask XGBoost in RAPIDS.

To learn more about RAPIDS, be sure to check out:

- Open Source Website
- GitHub
- Press Release
- NVIDIA Blog
- Developer Blog
- NVIDIA Data Science Webpage

In [ ]:

一 使用集群计算资源的方式

二 集群硬件和软件栈介绍

三 GPU集群容器工具介绍与容器示例

四 交互式使用计算资源的详细步骤

五 集群使用注意事项和账户申请流程

使用注意事项

千兆/万兆CEBSIT内网

不能在登陆节点做计算任务或其它高CPU/GPU消耗任务

GPU登录节点

- ✖ **计算任务**
- ✖ **大文件解压缩**
- ✖ **大规模编译代码**
- ✖ **Jupyter**
- ✖ **无活动保持ssh在线**
- ✖ **内网穿透**

Account application

Step 1: 填写申请表，课题组/平台负责人签字

Step 2: 将申请表拍照/扫描发送至hpc@ion.ac.cn

Step 3: 集群管理员收到申请后会把集群培训资料发送给您，经过学习之后可以通过邮件或者电话预约参加考试。

Step 4: 考试通过后联系集群管理员，加入GPU集群用户微信群。

Step 5: 开通集群账号，账号开通后集群管理员会回复用户名和密码。

注：邮箱：hpc@ion.ac.cn　电话：021-64032612