组学大数据 Omics Big Data

Spring 2025

上机课Lab 1：How to use computer cluster（supercomputer）

计算机集群（超算）使用介绍

1. 终端安装（Terminal）
   你的笔记本电脑需要有能访问计算机集群的终端程序。苹果电脑不用安装，可以直接打开终端访问计算机集群。如果是 Window 操作系统，请先安装可远程登录 Linux 服务器的终端程序，如 SSH，Putty 或者 Windows Powershell Terminal 等；另外需要安装 Linux 和 Windows 之间的文件传输终端软件，如 WinSCP 或者 FileZilla 等。

   （Your laptop needs a terminal program installed in order to access the computer cluster from your laptop. If you use a Mac, you can open a terminal directly. If you use a computer with Windows operating system, please install a terminal program such as SSH, Putty or Windows Powershell, with the ability to login to a Linux server. In addition, you may need to install software to transfer files between Linux and Windows, such as WinSCP or FileZilla, etc. ）

——》mobaxterm

2. 登录超算 Pi2.0 (Login to Pi2.0)
   见课堂 PPT，如果不清楚，请联系课程教师或助教。
   (See details in course PPT. In case you need help, please contact course teacher or TA)

——》pilogin

3. Pi2.0 超算介绍 (Introduction of Pi2.0 supercomputer)

   Pi2.0 中的队列
   1）cpu 允许单作业 CPU 核数为 1~24000，每核配比 4G 内存，节点可共享使用；单节点配置为 40 核，192G 内存
   2） huge 允许单作业 CPU 核数为 1~80，每核配比 35G 内存，节点可共享使用；单节点配置为 80 核，3T 内存
   3）192c6t 允许单作业CPU 核数为 1~192，每核配比 31G 内存，节点可共享使用；单节点配置为 192 核，6T 内存

4）dgx2 允许单作业 GPU 卡数为 1~128，每卡最高配比 6 核，每核配比 15G 内存；单节点配置为 96 核，1.45T 内存，16 块 32G 显存的 V100 卡（cpu 和 dgx2 队列作业运行时间最长 7 天，huge 和 192c6t 最长 2 天。作业延长需发邮件申请，附上用户名和作业 ID）

5）debug 仅用于短时间测试，请勿批量投递作业进行完整计算。队列每节点配置为 40 核，节点可共享使用，作业允许节点数 1~2，每核配比 4G 内存，最长运行时间为 20 分钟

Queues in Pi2.0

1） cpu: the number of CPU cores allowed for a single task is 1-24,000, with 4GB memory for each core. A node can be shared by multiple tasks. If you need to occupy a whole node for one task, you can do it by setting the number of cores to 40 or the memory size to 192GB for a task.

2） huge: the number of CPU cores allowed for a single task is 1-80, with 35GB memory for each core. A node can be shared by multiple tasks. If you need to occupy a whole node for one task, you can do it by setting the number of cores to 80 or the memory size to 3TB for a task.

3） 192c6t: the number of CPU cores allowed for a single task is 1-192, with 31GB memory for each core. A node can be shared by multiple tasks. If you need to occupy a whole node for one task, you can do it by setting the number of cores to 192 or the memory size to 6TB for a task.

4） dgx2: the number of GPU cards allowed for a single task is 1-128, every card can have maximum number of 6 cores, each core with 15GB memory for each core. A node have 96 cpu cores, 1.5TB memory, 16 GPU cards (V100) each with 32GB memory.
(The maximum time for a task to run in queue cpu or dgx2 is 7 days, 2 days for queue huge and 192c6t. If you need to extend the maximum time for your task, please send email to hpc staff with your username and task ID attached)

5） debug: only for testing. Don’t submit big tasks for production run. Each node has 40 cores. A node can be shared, the maximum number of nodes allowed is 1-2. Each cpu core have 4GB. The maximum running time is 20 minutes for a task.

4、作业递交脚本模板。
(Template for task management (job submission) script)

```
#!/bin/bash
#SBATCH --job-name=xxx #作业名称
#SBATCH --partition=xx #节点类型
#SBATCH -n xx #作业指定使用总核数
#SBATCH --ntasks-per-node=xx #指定每个节点使用核数
#SBATCH --mail-type=end # 作业结束发送邮件
#SBATCH --mail-user=[your_mail_address] #指定发送邮件的地址
```

#SBATCH --output=%j.out #指定输出文件
#SBATCH --error=%j.err #指定报错文件

xxx #实际执行的命令

编写作业脚本（*.slurm）,作业脚本示例：　　。
该作业指定作业名字为 fastqc，指定 cpu 节点，作业使用总核数为 16 个核，每个节点分配 16 个核，作业计算结束给用户发送邮件，有标准的报错文件。

(Please edit a script as follows for a job submission (*.slurm). For example, fastqc.slurm. This task name is fastqc, use cpu queue, with 16 cores each node; the system will send email to users, there is an error file.)

```
#!/bin/bash
#SBATCH --job-name=fastqc
#SBATCH --partition=cpu
#SBATCH -n 16
#SBATCH --ntasks-per-node=16
#SBATCH --mail-type=end
#SBATCH --mail-user=[your_mail_address] #请改成自己邮件的地址
#SBATCH --output=%j.out
#SBATCH --error=%j.err

module load fastqc
fastqc -t 40 -o ./ /lustre/share/class/BIO8402/lab/data/SRR5811639.fastq.gz
```

注：①其他可选参数：　（Note：　optional parameters）

```
#SBATCH -n XX                    #占用节点数（number of nodes）
#SBATCH --exclusive              #独占节点　（number of exclusive nodes）
#SBATCH –time=[dd-hh:mm:ss]      #指定作业最多运行时间　（max run time）
#SBATCH –nodelist=[nodes]        #指定在该节点运行作业（specify a node(s)）
#SBATCH --exclude=[nodes]        #指定不在该节点运行作业 (specify not run on these nodes)
#SBATCH -o %J.out                #输出结果文件 (output file name)
```

★区分-n 和--ntasks-per-node (Note：-n and --ntasks-per-node is different)

②当任务不支持 **MPI** 时时，假如要指定 cpu 节点 8 个核，-n 和--ntasks-per-node 要一起使用才能保证 **8** 个核心都来自同一个节点上。

（If MPI is not required for a task, if you want to assign 8 cores to a cpu node, you should use options -n and -ntasks-per-node simultaneously to ensure the 8 cores are from the same node.）

```
#SBATCH -n 8
#SBATCH --ntasks-per-node=8
```

5、常用命令：(commonly used commends)

(1) 提交任务(task submission)：sbatch blastn.slurm

Sbatch fastqc.slurm

(2) 杀掉任务(kill a task)：scancel jobid

(3) 查看任务(check the status of a task)： squeue

squeue –u “clswcc-wmz” 只展示个人目录下任务情况 (only show tasks under an individual directory)

```
[clswcc-wmz@mu05 new_2000_test]$ squeue -u 'clswcc-wmz'
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1233151       cpu   blastn clswcc-w  R       1:10      1 node302
           1233150       cpu   blastn clswcc-w  R       1:13      1 node273
           1233149       cpu   blastn clswcc-w  R       1:16      1 node235
           1233147       cpu   blastn clswcc-w  R       2:34      1 node115
           1233146       cpu   blastn clswcc-w  R       2:37      1 node087
           1233145       cpu   blastn clswcc-w  R       2:40      1 node085
```

Squeue

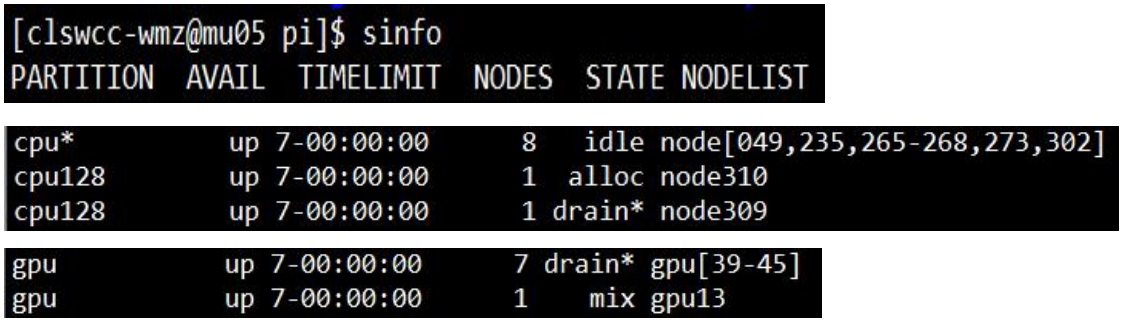
```
[stu542@pilogin4 test_class3]$ sbatch fastqc.slurm
Submitted batch job 42248178
[stu542@pilogin4 test_class3]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          42248178       cpu   fastqc   stu542  R       0:05      1 cas005
[stu542@pilogin4 test_class3]$ squeue -u $(whoami)
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[stu542@pilogin4 test_class3]$ whoami
stu542
[stu542@pilogin4 test_class3]$ squeue -u stu542
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[stu542@pilogin4 test_class3]$
```

- PARTITION: **节点类型 (queue name) cpu**
- ST： 任务状态——**R：正在运行**，PD：排队(task status---R: running, PD: pending)
- NODES: **任务所用节点数目(the number of nodes for a task) 1**
- NODESLIST: 任务所在节点，可通过 **ssh** 登陆该节点，例：ssh node220
  (the list of node to run the task. You can use ssh to login to this node. For example, ssh node220.)

```
[stu542@pilogin4 test_class3]$ ls -lh
total 664K
-rw-rw-r-- 1 stu542 stu542  875 Mar  7 13:09 42248178.err
-rw-rw-r-- 1 stu542 stu542   59 Mar  7 13:09 42248178.out
-rw-rw-r-- 1 stu542 stu542  351 Mar  7 13:05 fastqc.slurm
-rw-r--r-- 1 stu542 stu542  30K Mar  7 12:58 hello.tar
-rw-r--r-- 1 stu542 stu542  20K Mar  7 12:58 sleep.tar
-rw-rw-r-- 1 stu542 stu542 241K Mar  7 13:09 SRR5811639_fastqc.html
-rw-rw-r-- 1 stu542 stu542 350K Mar  7 13:09 SRR5811639_fastqc.zip
-rw-rw-r-- 1 stu542 stu542  426 Mar  7 13:01 test_script.slurm
[stu542@pilogin4 test_class3]$
```

(4) 查看节点状态：sinfo (check the status of a node：sinfo)

sinfo –p cpu /gpu/fat: 查看某一类型节点状态（check the status of nodes in a queue）



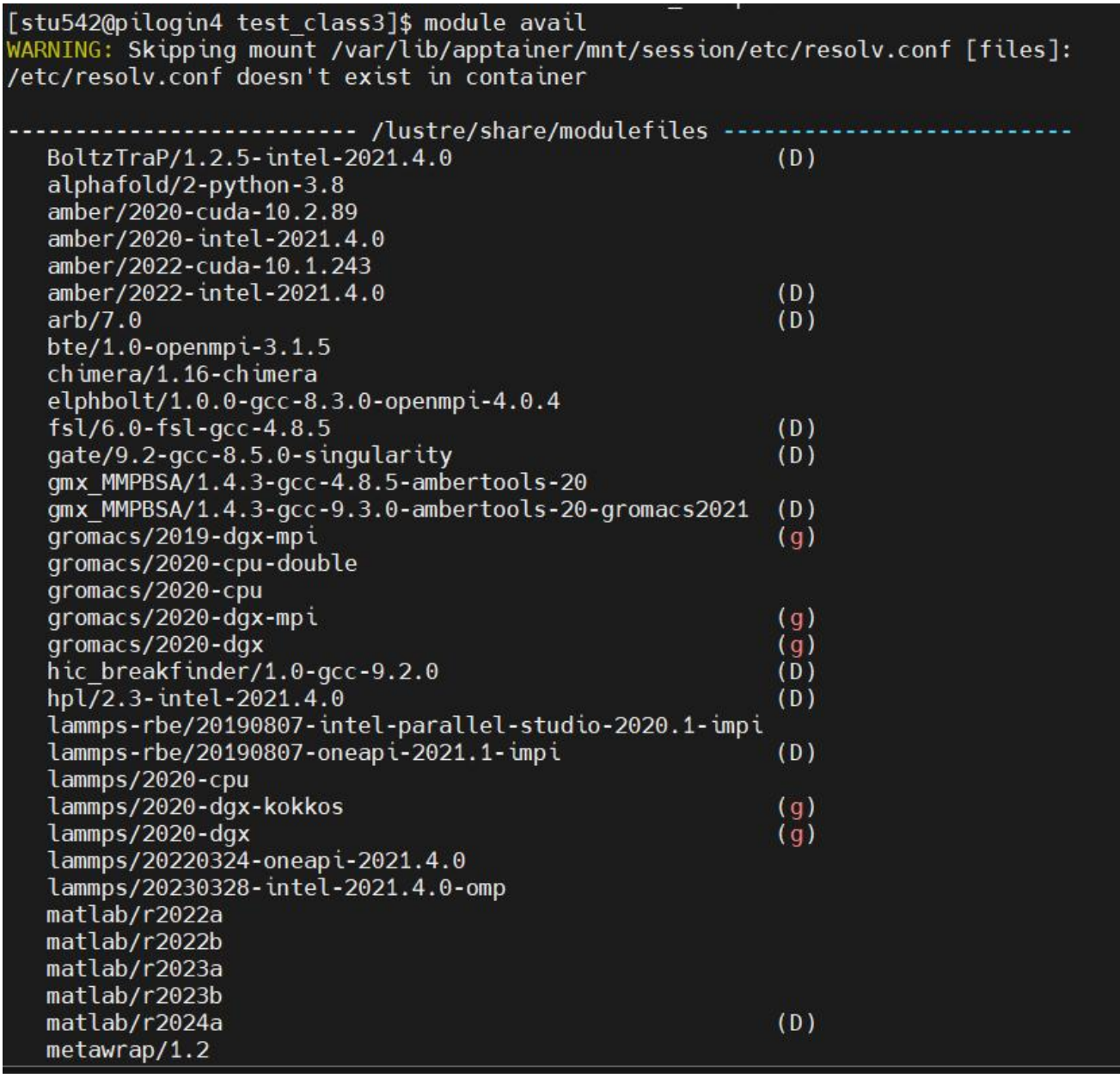
STATE: drain：节点有问题，alloc：正在全部正在使用，mix：部分可以用，idle：闲置，down：节点不可用

(5) 监控和再修改作业 (monitor or revise a task)

scontrol show job JOB_ID 显示该作业的一些信息

scontrol -dd show job JOB_ID 显示该作业的作业脚本

(6) module avail: 显示超算已安装的软件(show the modules installed)


[stu542@pilogin4 test_class3]$ module avail
WARNING: Skipping mount /var/lib/apptainer/mnt/session/etc/resolv.conf [files]:
/etc/resolv.conf doesn't exist in container
/lustre/share/modulefiles
BoltzTraP/1.2.5-intel-2021.4.0 (D)
alphafold/2-python-3.8
amber/2020-cuda-10.2.89
amber/2020-intel-2021.4.0
amber/2022-cuda-10.1.243
amber/2022-intel-2021.4.0 (D)
arb/7.0 (D)
bte/1.0-openmpi-3.1.5
chimera/1.16-chimera
elphbolt/1.0.0-gcc-8.3.0-openmpi-4.0.4
fsl/6.0-fsl-gcc-4.8.5 (D)
gate/9.2-gcc-8.5.0-singularity (D)
gmx_MMPBSA/1.4.3-gcc-4.8.5-ambertools-20
gmx_MMPBSA/1.4.3-gcc-9.3.0-ambertools-20-gromacs2021 (D)
gromacs/2019-dgx-mpi (g)
gromacs/2020-cpu-double
gromacs/2020-cpu
gromacs/2020-dgx-mpi (g)
gromacs/2020-dgx (g)
hic_breakfinder/1.0-gcc-9.2.0 (D)
hpl/2.3-intel-2021.4.0 (D)
lammps-rbe/20190807-intel-parallel-studio-2020.1-impi
lammps-rbe/20190807-oneapi-2021.1-impi (D)
lammps/2020-cpu
lammps/2020-dgx-kokkos (g)
lammps/2020-dgx (g)
lammps/20220324-oneapi-2021.4.0
lammps/20230328-intel-2021.4.0-omp
matlab/r2022a
matlab/r2022b
matlab/r2023a
matlab/r2023b
matlab/r2024a (D)
metawrap/1.2

```
[clswcc-wmz@mu05 ~]$ module avail

------------------------------------------------------------------------ /lustre/usr/modulefiles/pi -------
abinit/7.10(default)       ctffind/3.4(default)       gcc/5.1                    jdk/1.8(default)
abyss/1.9(default)         cub/1.4(default)           gcc/5.2                    julia/0.4(default)
anaconda/2(default)        cub/1.5                    gcc/5.4                    kaldi/0.1(default)
anaconda/3                 cuda/6.5                   gdk/352(default)           knl/15.0(default)
ascend/0.9(default)        cuda/7.5(default)          genetorrent/3.8(default)   lammps/20161117
atlas/3.10(default)        cuda/8.0                   gflags/2.1(default)        lapack/3.5(default)
bamtools/2.4(default)      cudnn/1.0                  ghc/7.4(default)           leveldb/1.18(default)
```

module purge: 清理现有环境
module load： 加载某个软件
module unload: 卸载某个软件
module list: 显示自己目录下已加载的软件

```
[clswcc-wmz@mu05 ~]$ module purge
[clswcc-wmz@mu05 ~]$ module load jdk/1.8 perl/5.22 python/2.7
[clswcc-wmz@mu05 ~]$ module unload perl/5.22
[clswcc-wmz@mu05 ~]$ module list
Currently Loaded Modulefiles:
  1) jdk/1.8      2) gcc/4.9      3) python/2.7
```

6、超算计时方法 (the method to measure the computational time) 1）

CPU 计时方法 (CPU time measuring method)

单个作业所消耗的机时，与其运行时长及使用的 CPU 核心数有关，计算公式为 T = t * n
n 为作业使用的 CPU 核心数，t 为作业的运行时长(单位：小时，作业排队时间不计入消耗的机时)，T 为该作业消耗的机时量(单位：核小时)。
例：一个作业在独占一个 16 核的 cpu 节点，运行一个小时则消耗机时为 16 核小时。Pi2.0 单个 cpu 节点上的核数可能有所增加。请注意查看具体情况。

The computing time for a task depends on the running time and the number of CPU cores : T= t*n
n is the number of CPU cores， t is the running time （unit: hour, queueing time is not included in the computing time）, T is the computing time of the task (unit: core hour).
For example, a task uses a cpu node with 40 cores exclusively, running one hour means 40 core hours for the task.

2）GPU 计时方法 (GPU time measuring method)
运行在 GPU 队列上的作业，机时使用是按照 CPU 耗时折算而来的。一个 GPU 节点上配置了 96 个 CPU 核,包括 16 块 GPU 加速卡。
GPU 机时的计费单位是 “卡小时”(cardhours),卡小时与用于计算 CPU 机时费的单位 “核小时”(corehours) 换算关系为 1 cardhour = 120 corehours
例：一个作业独占一个 GPU 节点运行一个小时则消耗机时为 16*120 核小时！

Tasks running on GPU queue, the total computational time will convert to CPU core hours. A GPU node contains 96 CPU cores, include 16 GPU cards. The unit for GPU node is cardhour. The relationship between cardhour and cpu core hour is 1 cardhour =120 corehours.
For example, if a task uses a GPU node exclusively for one hour, the computing time will be 16*120 cpu hours.

请在你自己的工作目录下进第 7 - 8 项练习。（Do exercise 8-9 under your own working directory）

7. 读写压缩文件的小窍门（压缩状态下分析文件的方法）(tricks using compressed files)

- Linux

  zless 可直接按行读入 gz 文件

  (you can use zless read in .gz files directly)

```
(base) [ccwei@mu01 data]$ zless SRR5811639.fastq.gz |head -n 5
@SRR5811639.1 HWI-ST632:393:C55BGACXX:6:1101:1200:1886 length=50
NTCCAGCTTCAGCAGGATCTGCACTGTCATGGCTGTGCACTCCAGGCTGT
+SRR5811639.1 HWI-ST632:393:C55BGACXX:6:1101:1200:1886 length=50
#1BDFFFFHHHHHJJJJJJJJJJJJJIJJJJJJJJIIJJJJJJGHIEHII
@SRR5811639.2 HWI-ST632:393:C55BGACXX:6:1101:1224:1898 length=50
```

```
[stu542@pilogin4 test_class3]$ zless /lustre/share/class/BIO8402/lab/data/SRR5811639.fastq.gz | head -n 5
@SRR5811639.1 HWI-ST632:393:C55BGACXX:6:1101:1200:1886 length=50
NTCCAGCTTCAGCAGGATCTGCACTGTCATGGCTGTGCACTCCAGGCTGT
+SRR5811639.1 HWI-ST632:393:C55BGACXX:6:1101:1200:1886 length=50
#1BDFFFFHHHHHJJJJJJJJJJJJJIJJJJJJJJIIJJJJJJGHIEHII
@SRR5811639.2 HWI-ST632:393:C55BGACXX:6:1101:1224:1898 length=50
[stu542@pilogin4 test_class3]$ S
```

- Python

  import gzip

```
import gzip
f = gzip.open('file.txt.gz', 'rb')
file_content = f.read()
f.close()
```

8. RNA 序列比对——以 blast 为例 （RNA sequence alignment, with blast）
建库+比对 (database construction + sequence alignment)

```
#!/bin/bash
#SBATCH --job-name=blast
#SBATCH --partition=cpu
#SBATCH -n 40
#SBATCH --ntasks-per-node=40
#SBATCH --output=%j.out
```

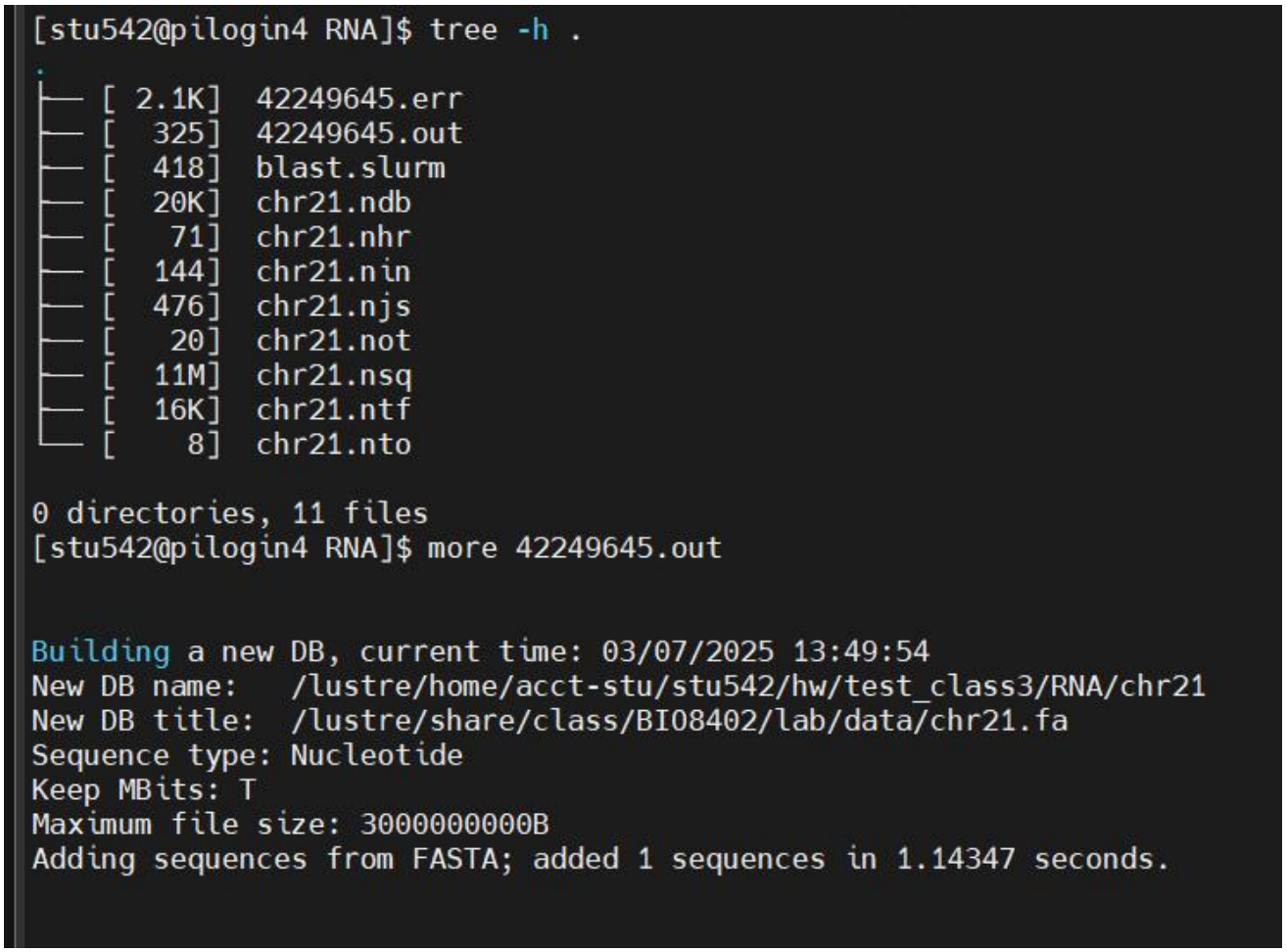
#SBATCH --error=%j.err

module load blast-plus
makeblastdb -in /lustre/share/class/BIO8402/lab/data/chr21.fa -dbtype nucl -out ./chr21
blastn -query /lustre/share/class/BIO8402/lab/data/SRR5029637.fasta -db ./chr21 -evalue 1e-5 -outfmt 7 -max_target_seqs 1 -num_threads 40 -out ./result

```
[stu542@pilogin4 RNA]$ ls
42249645.err  42249645.out  blast.slurm  chr21.ndb  chr21.nhr  chr21.nin  chr21.njs  chr21.not  chr21.nsq  chr21.ntf  chr21.nto
[stu542@pilogin4 RNA]$ ls -lh
total 12M
-rw-rw-r-- 1 stu542 stu542 2.2K Mar  7 13:49 42249645.err
-rw-rw-r-- 1 stu542 stu542  325 Mar  7 13:49 42249645.out
-rw-rw-r-- 1 stu542 stu542  418 Mar  7 13:49 blast.slurm
-rw-rw-r-- 1 stu542 stu542  20K Mar  7 13:49 chr21.ndb
-rw-rw-r-- 1 stu542 stu542   71 Mar  7 13:49 chr21.nhr
-rw-rw-r-- 1 stu542 stu542  144 Mar  7 13:49 chr21.nin
-rw-rw-r-- 1 stu542 stu542  476 Mar  7 13:49 chr21.njs
-rw-rw-r-- 1 stu542 stu542   20 Mar  7 13:49 chr21.not
-rw-rw-r-- 1 stu542 stu542  12M Mar  7 13:49 chr21.nsq
-rw-rw-r-- 1 stu542 stu542  16K Mar  7 13:49 chr21.ntf
-rw-rw-r-- 1 stu542 stu542    8 Mar  7 13:49 chr21.nto
[stu542@pilogin4 RNA]$
```

```
[stu542@pilogin4 RNA]$ tree -h .
.
├── [ 2.1K]  42249645.err
├── [  325]  42249645.out
├── [  418]  blast.slurm
├── [  20K]  chr21.ndb
├── [   71]  chr21.nhr
├── [  144]  chr21.nin
├── [  476]  chr21.njs
├── [   20]  chr21.not
├── [  11M]  chr21.nsq
├── [  16K]  chr21.ntf
└── [    8]  chr21.nto

0 directories, 11 files
```


[stu542@pilogin4 RNA]$ tree -h .
.
├── [2.1K] 42249645.err
├── [325] 42249645.out
├── [418] blast.slurm
├── [20K] chr21.ndb
├── [71] chr21.nhr
├── [144] chr21.nin
├── [476] chr21.njs
├── [20] chr21.not
├── [11M] chr21.nsq
├── [16K] chr21.ntf
└── [8] chr21.nto
0 directories, 11 files
[stu542@pilogin4 RNA]$ more 42249645.out
Building a new DB, current time: 03/07/2025 13:49:54
New DB name: /lustre/home/acct-stu/stu542/hw/test_class3/RNA/chr21
New DB title: /lustre/share/class/BIO8402/lab/data/chr21.fa
Sequence type: Nucleotide
Keep MBits: T
Maximum file size: 3000000000B
Adding sequences from FASTA; added 1 sequences in 1.14347 seconds.

9. 上机实验报告（请完成以下计算，并完成实验报告后递交。）

（**Lab report： please finish the following tasks and submit your report to course website**）

在超算/lustre/share/class/BIO8402/lab/data/路径下，拟南芥 illumina 测序数据为SRR5029637.fasta，拟南芥参考基因组 GCF_000001735.3_TAIR10_genomic.fna 及用 makeblastdb 所建 index，用 BLAST 2.5.0+(blastn)进行比对。

(In Pi2.0, please use makeblastdb to create a database for blast program. Data are under /lustre/share/class/BIO8402/lab/data/, including illumine sequencing data SRR5029637.fasta, the reference genome for Arabidopsis is GCF_000001735.3_TAIR10_genomic.fna. Use BLAST 2.5.0+(blastn) for sequence alignment.)

测序数据：

/lustre/share/class/BIO8402/lab/data/SRR5029637.fasta

```
[stu542@pilogin4 RNA]$ cat /lustre/share/class/BIO8402/lab/data/SRR5029637.fasta | head -n 10
>SRR5029637.1 SEQCORE-1795804:236:HWKKCADXX:1:1101:1245:2059 length=101
AANGCCAAAGACTCATACGGACTTTGGCTACACCATGAAAGCTTTGAGAAGCTAGAAGAAGGTTGGTTAG
TGTTTTGGAGTCGAATATGACTTGATGTCAT
>SRR5029637.2 SEQCORE-1795804:236:HWKKCADXX:1:1101:1185:2076 length=101
ATCCATATTATGCAAGGAGACATTGCTTTTGCTAATTCGAATTGAAGGGTGNTATAAAATCGGTCTATTT
CCAACATCATATCCATAGTTAGCGCATTCAT
>SRR5029637.3 SEQCORE-1795804:236:HWKKCADXX:1:1101:1158:2093 length=101
CTTTGATTATAGATAGAAGAGACCTTAGAGGGCCATCTCAGCCTGTAATGCNGCGAGTTCTTCTTCCTCA
GCAGTCCGCTTCTGGGGAACAGGACGAGCCG
>SRR5029637.4 SEQCORE-1795804:236:HWKKCADXX:1:1101:1358:2058 length=101
[stu542@pilogin4 RNA]$
```

参考基因组：/lustre/share/class/BIO8402/lab/data/GCF_000001735.3_TAIR10_genomic.fna

```
[stu542@pilogin4 RNA]$ cat /lustre/share/class/BIO8402/lab/data/GCF_000001735.3_TAIR10_genomic.fna | head -n 10
>NC_003070.9 Arabidopsis thaliana chromosome 1 sequence
ccctaaaccctaaaccctaaaccctaaacctctGAATCCTTAATCCCTAAATCCCTAAATCTTTAAATCCTACATCCATG
AATCCCTAAATACCTAAttccctaaacccgaaaccggTTTCTCTGGTTGAAAATCATTGTGtatataatgataatttttat
CGTTTTTATGTAATTGCTTATTGTTGTGtgtagatttttttaaaaatatcatttgagGTCAATACAAATCCTATTTCTTGT
GGTTTTCTTTCCTTCACTTAGCTATGGATGGTTTATCTTCATTTGTTATATTGGATACAAGCTTTGCTACGATCTACATT
TGGGAATGTGAGTCTCTTATTGTAACCTTAGGGTTGGTTTATCTCAAGAATCTTATTAATTGTTTGGACTGTTTATGTTT
GGACATTTATTGTCATTCTTACTCCTTTGTGGAAATGTTTGTTCTATCAATTTATCTTTTGTGGgaaaattatttagttg
taGGGATGAAGTCTTTCTTCGTTGTTGTTACGCTTGTCATCTCATCTCTCAATGATATGGGATGGTCCTTTAGCATTTAT
TCTGAAGTTCTTCTGCTTGATGATTTTATCCTTAGCCAAAAGGATTGGTGGTTTGAAGACACATCATATCAAAAAAGCTA
TCGCCTCGACGATGCTCTATTTCTATCCTTGTAGCACACATTTTGGCACtcaaaaaagtatttttagatgtttgttttgc
[stu542@pilogin4 RNA]$
```

makeblastdb 所建 index

BLAST 2.5.0+(blastn)

1） 请填写数据量大小?（10 分）

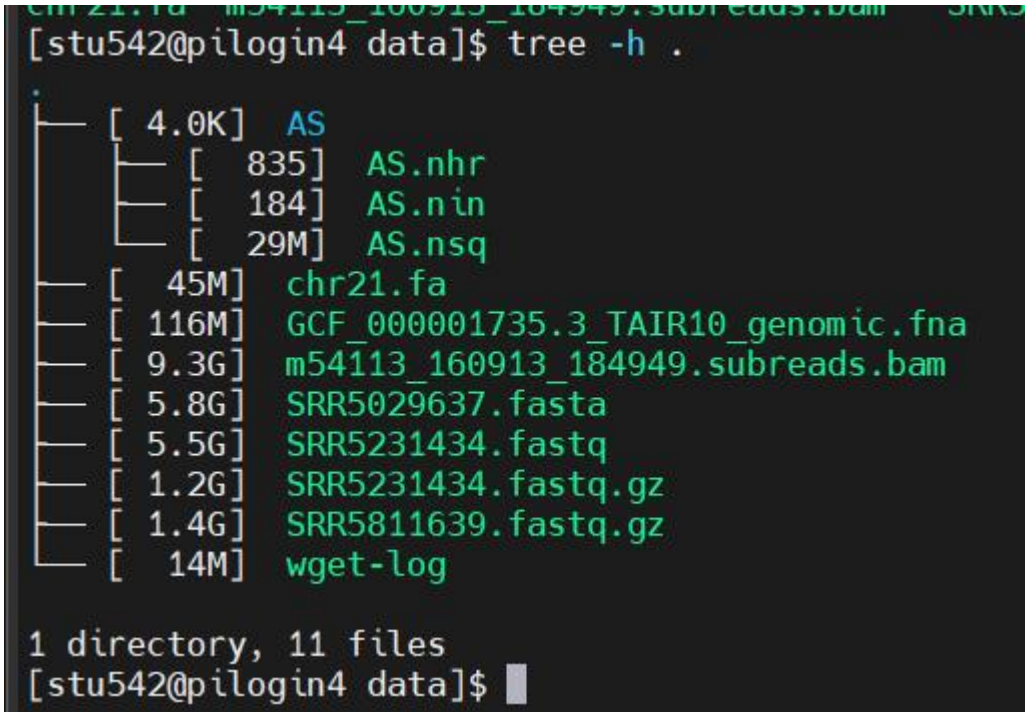
(what is the size of the input file?, 10 points)

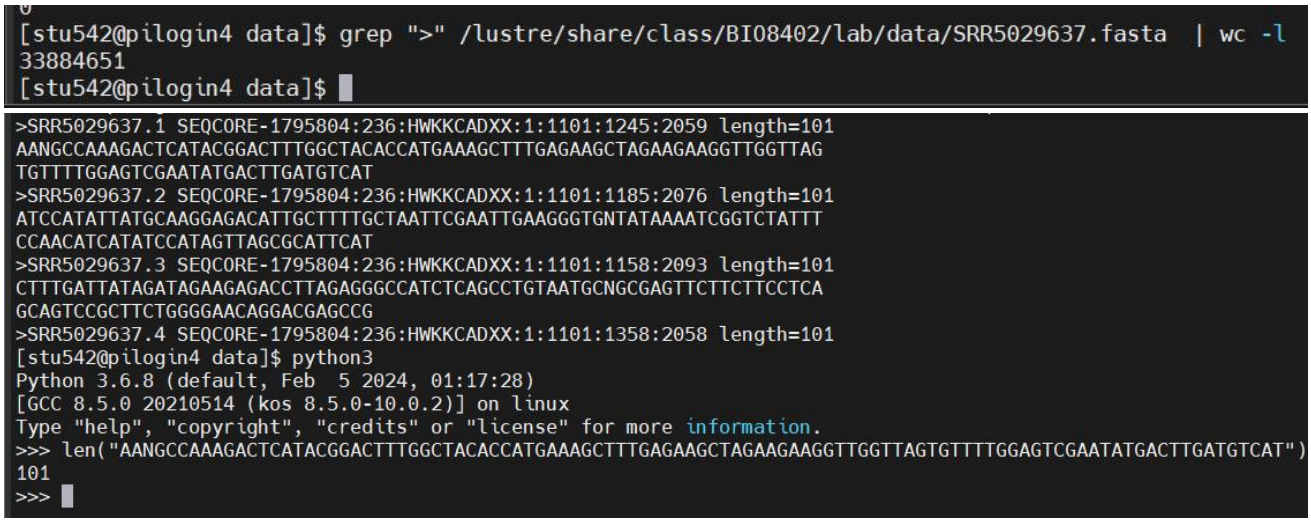
文件大小(file size)：5.8G

read 数 (read number)： **33884651**

每条 read 长度(the read length)：101 （python和record同）

```
42249645.err  42249645.out  blast.sturm  chr21.ndb  chr21.nhr  chr21.nin  chr21.njs  chr21.not  chr21.ns
[stu542@pilogin4 RNA]$ ls -lh /lustre/share/class/BIO8402/lab/data/SRR5029637.fasta
-rw-rwxr--+ 1 stu438 stu438 5.8G Mar  5  2024 /lustre/share/class/BIO8402/lab/data/SRR5029637.fasta
[stu542@pilogin4 RNA]$
```





2）blastn 支持 MPI 吗？若不支持，为保证在同一个节点上计算，当指定核数-n 大于等于 2 时，超算上要和哪个参数一起使用？（10 分）

(Does blastn support MPI? If not, what parameter you need to use to ensure your task will run in one node if you use option -n >= 2?)

Blastn不支持跨节点计算MPI；

★区分-n 和--ntasks-per-node **(Note：-n and --ntasks-per-node is different)**

②当任务**不支持** MPI **时**时，假如要指定 cpu 节点 8 个核，-n 和--ntasks-per-node 要一起使用才能**保证 8 个核心都来自同一个节点上**。

（If MPI is not required for a task, if you want to assign 8 cores to a cpu node, you should use options -n and -ntasks-per-node simultaneously to ensure the 8 cores are from the same node.）

```
#SBATCH -n 8
#SBATCH --ntasks-per-node=8
```

-n和--ntasks-per-node参数一起用

3) 选择什么参数? （20 分）

(Which parameter should you choose?, 20 points)

将拟南芥测序数据 SRR5029637.fasta 与其参考基因组进行 blastn 比对，

在给定条件下选择 slurm 作业参数：任务名为 blastn，指定核数为 8，指定最大虚拟内存数为 200G，作业结束后发邮件通知，希望输出报错文件。

请在要选择的参数处 √，需要填写具体数目的请填写。

（Please align a sequence file SRR5029637.fasta to the Arabidopsis reference genome using blastn. Choose the slurm parameters as required: task name as blastn, number of cores 8, maximum verture memory 200GB, send a mail after task finishs, output an error file. Please put a √ if you need to specify this parameter and fill the cell with a value if needed.）

参数(parameters)	是否选择及数目(parameter value)
#!/bin/bash #SBATCH --job-name=fastqc	√，blastn
#SBATCH --partition=cpu	√，huge
#SBATCH -n 16	√，8
#SBATCH --ntasks-per-node=16	√，8
#SBATCH --mail-type=end	√，结束时通知
#SBATCH --mail-user=[your_mail_address]	√，zht161932@sjtu.edu.cn
#SBATCH --output=%j.out	√，普通输出log
#SBATCH --error=%j.err	√，错误输出文件

Pi2.0 中的队列
1）cpu 允许单作业 CPU 核数为 1~24000，每核配比 4G 内存，节点可共享使用；单节点配置为 40 核，192G 内存
2）huge 允许单作业 CPU 核数为 1~80，每核配比 35G 内存，节点可共享使用；单节点配置为 80 核，3T 内存
3）192c6t 允许单作业 CPU 核数为 1~192，每核配比 31G 内存，节点可共享使用；单节点配置为 192 核，6T 内存

因为是使用cpu队列，
指定最大虚拟内存数为 200G
而每核配比4G内存，所以我们需要申请50核，但是单节点配置最多就是40核，需要多节点
因为blastn不支持MPI，
所以我们用其他队列，比如说是huge——》在限制8核内，每个核起码25G内存
-n以及-ntasks-per-node

4）阅读以下实例分析，考虑应该选择的线程数 （10）
(read the analysis below and give the number of threads you need to set. 10 points)

拟南芥 pacbio 测序数据 subreads-A01.fasta（数据文件未提供），数据量大小：5.2G，reads 数目：561176，每条 read 长度不等，碱基总数：748508361。用 BLAST 2.5.0+与拟南芥参考基因组进行比对，在用整个文件比对前首先用小文件进行测试，选择最重要的参数-n=？-num_threads=?

BLAST 2.5.0+(blastn)

```
[stu542@pilogin3 RNA]$ blastn -help | grep " -n"
 -num_descriptions <Integer, >=0>
 -num_alignments <Integer, >=0>
 -negative_gilist <String>
 -negative_seqidlist <String>
 -negative_taxids <String>
 -negative_taxidlist <String>
 -no_greedy
 -num_threads <Integer, >=1>
[stu542@pilogin3 RNA]$
```

```
[stu542@pilogin3 RNA]$ module load blast-plus
[stu542@pilogin3 RNA]$ blastn
BLAST query/options error: Either a BLAST database or subject sequence(s) must be specified
Please refer to the BLAST+ user manual.
[stu542@pilogin3 RNA]$
```

在本例中，选择了 100 条 reads、561 条 reads 和 1000 条 reads 三个文件分别进行 blastn 比对，选择核数分别为 1，2，4，8，16，线程参数选择为 1，2，4，8，16，32，64，统计出在固定核和固定线程下的计算时间，然后估算出相应参数下整个文件 subreads-A01.fasta 计算完成的时间，并计算出在相应参数下的计算核时。

Supporse we have a sequence file subreads-A01.fasta (data file not provided), the size of the file: 5.2GB, the number of reads: 561176, the length for each read is different, and the total number of bases: 748508361. If you want to align this file to the Arabidopsis reference genome using BLAST 2.5.0+， before you use the whole file you want to test the pipeline with a smaller file. Please choose two important parameters -n=? and -num_tread=?)

100 条reads 的计算时间（s）（computing time for 100 reads）

thread / core	1	2	4	8	16	32	64
n1	460	462	462	463	463	463	463
n2		330	267	257	277	262	259
n4			207	205	207	201	199
n8				196	192	191	191
n16					188	190	190

561 条reads 的计算时间（s）（computing time for 561 reads）

thread / core	1	2	4	8	16	32	64
n1	1220	1340	1354	1350	1345	1343	1229
n2		753	765	776	778	800	765
n4			623	598	573	570	578
n8				522	517	517	517
n16					505	502	503

1000 条reads 的计算时间（s）（computing time for 1000 reads）

thread / core	1	2	4	8	16	32	64
n1	2409	2420	2432	2436	2429	2428	2431
n2		1334	1389	1357	1342	1365	1383
n4			1032	1012	996	1007	1005
n8				937	929	932	931
n16					909	910	912

根据 100 条reads 测试结果估算原文件的计算时间（h）
（Estimated computing time （hours）based on 100 reads）

thread / core	1	2	4	8	16	32	64
n1	717.06	720.18	720.18	721.73	721.73	721.73	721.73
n2		514.41	416.21	400.62	431.79	408.41	403.73
n4			322.68	319.56	322.68	313.32	310.21
n8				305.53	299.29	297.74	297.74
n16					293.06	296.18	296.18

根据 561 条reads 测试结果估算原文件的计算时间（h）
（Estimated computing time （hours）based on 561 reads）

thread / core	1	2	4	8	16	32	64
n1	339.00	372.34	376.23	375.12	373.73	373.17	341.50
n2		209.23	212.57	215.62	216.18	222.29	212.57
n4			173.11	166.16	159.22	158.38	160.61
n8				145.05	143.66	143.66	143.66
n16					140.32	139.49	139.77

根据 1000 条 reads 测试结果估算原文件的计算时间（h）
（Estimated computing time （hours）based on 1000 reads）

thread / core	1	2	4	8	16	32	64
n1	375.52	377.23	379.11	379.73	378.64	378.48	378.95
n2		207.95	216.52	211.53	209.19	212.78	215.59
n4			160.87	157.75	155.26	156.97	156.66
n8				146.06	144.81	145.28	145.13
n16					141.70	141.85	142.16

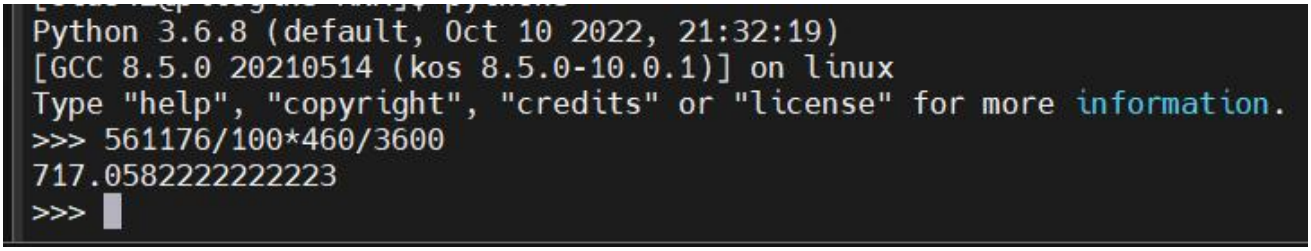
根据 561 条 reads 测试结果估算原文件的计算核时（h）
（Estimated computing core hours based on 561 reads）

thread / core	1	2	4	8	16	32	64
n1	339.00	372.34	376.23	375.12	373.73	373.17	341.50
n2		418.46	425.13	431.25	432.36	444.58	425.13
n4			692.44	664.65	636.87	633.53	642.42
n8				1160.36	1149.25	1149.25	1149.25
n16					2245.15	2231.81	2236.26

根据 1000 条 reads 测试结果估算原文件的计算核时（h）
（Estimated computing core hours based on 1000 reads）

根据 1000 条 reads 测试结果估算原文件的计算核时

thread / core	1	2	4	8	16	32	64
n1	375.52	377.23	379.11	379.73	378.64	378.48	378.95
n2		415.89	433.04	423.06	418.39	425.56	431.17
n4			643.48	631.01	621.03	627.89	626.65
n8				1168.49	1158.52	1162.26	1161.01
n16					2267.15	2269.65	2274.63



从统计结果可以看出，

①从估算出的整个文件 subreads-A01.fasta 比对完成的时间来看，561 条和 1000 条统计出的时间几乎一样，而用 100 条估算出的时间与它们相差较大，可能原因为当条数很少时计算时间并不占主要时间，因此**不可以选择 100 条 reads 当作测试文件**。

②当**核数为 1** 时，线程为 1 计算最快，在此例子中**线程增加没有加快计算速度**。

③当**核数大于 1** 时，程序实现真正的并行计算，当核数增加时，速度变快，但增速的速度会变慢，从核数 1 到核数 2 速度接近变为 2 倍，从核数 8 到核数 16 速度只变快非常少，因此用 16 个核是非常浪费的，**应该在核数 2，4，8 中选择**。

④在核数 2，4，8 的计算时间对比中，考虑到速度排除核数 2，因为核数 4 和 8 比核数 2 时要快四五十个小时。**核数 4 和 8 中，虽然当指定 8 个核时速度有所提升，但相对差别不大，而计算核时却大大提高**。

问题：从速度和经济的综合考虑，这个任务最好应该选择____4__个核，__16__个线程。

5）你会怎么选择？（How will you choose?）50 points

请在教学服务器上进行以下计算，请用上述提到的拟南芥 illumina 测序数据 SRR5029637.fasta 与其参考基因组进行 blastn 比对，命令行如下：

(Please align the sequence file above to the abrabidopsis reference genome using blastn with the command line below:)

```
echo -e " -n2-threads2 starts at `date`.\n" >>n2.log
start=$(date +%s)
blastn -task blastn -query query.fasta -db index -evalue 1e-5 -outfmt 7 –max_target_seqs 1 -num_threads 2 -out 2.blastn
end=$(date +%s)
time=$(( $end - $start ))
echo -e "difftime is $time\n" >>n2.log
echo -e " -n2-threads2 ends at `date`.\n" >>n2.log
```

解读：

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 3 " -task"
 -task <String, Permissible values: 'blastn' 'blastn-short' 'dc-megablast'
                'megablast' 'rmblastn' >
   Task to execute
   Default = `megablast'
```

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 3 " -query"
 -query <File_In>
   Input file name
   Default = `-'
 -query_loc <String>
   Location on the query sequence in 1-based offsets (Format: start-stop)
 -strand <String, `both', `minus', `plus'>
   Query strand(s) to search against database/subject
[stu542@pilogin4 ~]$
```

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 3 " -db"
 -db <String>
   BLAST database name
    * Incompatible with:  subject, subject_loc
```

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 3 " -evalue"
 -evalue <Real>
   Expectation value (E) threshold for saving hits
   Default = `10'
```

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 70 " -outfmt"
 -outfmt <String>
   alignment view options:
     0 = Pairwise,
     1 = Query-anchored showing identities,
     2 = Query-anchored no identities,
     3 = Flat query-anchored showing identities,
     4 = Flat query-anchored no identities,
     5 = BLAST XML,
     6 = Tabular,
     7 = Tabular with comment lines,
     8 = Seqalign (Text ASN.1),
     9 = Seqalign (Binary ASN.1),
    10 = Comma-separated values,
    11 = BLAST archive (ASN.1),
    12 = Seqalign (JSON),
    13 = Multiple-file BLAST JSON,
    14 = Multiple-file BLAST XML2,
    15 = Single-file BLAST JSON,
    16 = Single-file BLAST XML2,
    17 = Sequence Alignment/Map (SAM),
    18 = Organism Report
```

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 6 " -max_target_seqs"
 -max_target_seqs <Integer, >=1>
   Maximum number of aligned sequences to keep
   (value of 5 or more is recommended)
   Default = `500'
    * Incompatible with:  num_descriptions, num_alignments

 *** Discontiguous MegaBLAST options
[stu542@pilogin4 ~]$
```

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 6 " -num_threads"
 -num_threads <Integer, >=1>
   Number of threads (CPUs) to use in the BLAST search
   Default = `1'
    * Incompatible with:  remote
```

```
[stu542@pilogin4 ~]$ blastn -help | grep -A 3 " -out"
 -out <File_Out, file name length < 256>
   Output file name
   Default = `-'
```

使用 **-task blastn** 指定基本核酸序列比对，**-query SRR5029637.fasta** 指定待比对序列文件，**-db arabidopsis_index** 指定参考基因组数据库前缀，**-evalue 1e-5** 设置期望值阈值，**-outfmt 7** 输出为表格格式，**-max_target_seqs 1** 限定只保留最高匹配结果1条，**-num_threads 2** 使用 **2** 个线程加快比对，**-out out_2.blastn** 定义结果输出文件；

理论上还是使用第8题中所涉及的数据以及指令，理论上应该seqtk随机提取出50000条reads，然后执行命令之后实际记录时间，但是既然要评估整体运行时间，也可以直接跑全程。
但是下方表格已经给出了数据；
自行脚本处理part见该题尾部

请变换不同的指定核数和线程参数进行计算，并填写以下三张表格中的括号处。

(Please use different number of cores and threads to do the computing and file the three tables below.)

50000 条 reads 的计算时间（s）

The computing time using 50000 reads(seconds)

thread / core	1	2	4	8	16	32	64
n1	1036	652	513	416	418	421	423
n2		646	509	411	412	419	415
n4			509	413	416	417	419

根据 50000 条 reads 测试结果估算原文件的计算时间（h）

The computing time (CPU hours) estimated for the whole file using 50000 reads

thread / core	1	2	4	8	16	32	64
n1	195.02	122.74	96.57	78.31	78.69	79.25	79.63
n2		121.61	95.82	77.37	77.56	78.88	78.12
n4			（95.82）	（77.75）	（78.31）	（78.50）	（78.88）

33884651这个是原文件的reads数目

```
>>> 33884651/50000*1036/3600
195.0249913111111
>>> 33884651/50000*652/3600
122.73773584444446
>>> 33884651/50000*513/3600
96.57125535
>>> 33884651/50000*416/3600
78.31119342222223
>>> 33884651/50000*418/3600
78.68768954444445
>>> 33884651/50000*421/3600
79.25243372777777
>>> 33884651/50000*423/3600
79.62892985
>>>
```

上面是第1行的预估事件计算

然后第3行是：

```
>>> 33884651/50000*509/3600
95.81826310555556
>>> 33884651/50000*413/3600
77.74644923888889
>>> 33884651/50000*416/3600
78.31119342222223
>>> 33884651/50000*417/3600
78.49944148333334
>>> 33884651/50000*419/3600
78.87593760555556
>>>
```

根据 50000 条 reads 测试结果估算的原文件的计算核时

The computing core hours estimated for the whole file using 50000 reads(hours)

thread / core	1	2	4	8	16	32	64
n1	195.02	122.74	96.57	78.31	78.69	79.25	79.63
n2		243.22	191.64	154.74	155.12	157.75	156.25
n4			（383.27）	（310.99）	（313.24）	（314.00）	（315.50）

此处的第3行也就是乘个系数罢了

```
>>> 33884651/50000*509/3600*4
383.27305242222224
>>> 33884651/50000*413/3600*4
310.98579695555554
>>> 33884651/50000*416/3600*4
313.2447736888889
>>>
KeyboardInterrupt
>>> 33884651/50000*417/3600*4
313.99776593333337
>>> 33884651/50000*419/3600*4
315.50375042222225
```

四舍五入保留2位小数

（1） 请填写 n=4 时，估算的计算时间和计算核时。

(If n=4, estimate the cpu hours and core hours for the computing task.)

结果填写如上

（2）根据上述计算结果，你会选择的指定核数为（1），线程参数为（8），此时原文件计算完成估算时间约为（78.31）小时，计算核时为（78.31），如果此为在超算上的计算结果，每个核时 0.05 元，则此计算任务需要花费（3.92）元。

(based on the estimation result, you will choose the number of cores = ( ), number of threads = ( ), the computing time for the whole file will be about ( ) hours, ( ) core hours. Every core hour costs 0.05￥, and the total cost for this computing task (￥)。)

这是仅仅依据计算运行的时间：运行时间角度

根据 50000 条 reads 测试结果估算原文件的计算时间（h）

The computing time (CPU hours) estimated for the whole file using 50000 reads

thread / core	1	2	4	8	16	32	64
n1	195.02	122.74	96.57	78.31	78.69	79.25	79.63
n2		121.61	95.82	77.37	77.56	78.88	78.12
n4			(95.82)	(77.75)	(78.31)	(78.50)	(78.88)

这是仅仅依据运行核时的角度：经济收费角度

根据 50000 条 reads 测试结果估算的原文件的计算核时

The computing core hours estimated for the whole file using 50000 reads(hours)

thread / core	1	2	4	8	16	32	64
n1	195.02	122.74	96.57	78.31	78.69	79.25	79.63
n2		243.22	191.64	154.74	155.12	157.75	156.25
n4			(383.27)	(310.99)	(313.24)	(314.00)	(315.50)

其实运行时间上最佳的几个都差不多，所以要考虑的实际上就是运行核时，在物理运行时间拉不开的情况下，可以选择尽可能的经济

```
>>> 78.31*0.05
3.9155
>>>
```

```
[stu542@pilogin4 blast_time_test]$ sbatch blast_test.slurm
Submitted batch job 42309524
[stu542@pilogin4 blast_time_test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          42309524       cpu    blast   stu542 PD       0:00      1 (Priority)
```

```
[stu542@pilogin4 blast_time_test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          42309524       cpu    blast   stu542  R       0:18      1 cas027
```

再尝试另外一个脚本，用于seqtk抽取5w条read进行处理：

```
[stu542@pilogin4 5w_test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          42310193       cpu     bash   stu542 PD       0:00      1 (AssocMaxJobsLimit)
          42309524       cpu    blast   stu542  R       8:55      1 cas027
[stu542@pilogin4 5w_test]$
```

先cancel

```
[stu542@pilogin4 ~]$ srun -p cpu -n 4 --pty /bin/bash
srun: job 42310193 queued and waiting for resources

srun: job 42310193 has been allocated resources

bash-4.4$
bash-4.4$ module load miniconda3
bash-4.4$ which conda
/lustre/opt/cascadelake/linux-rhel8-skylake_avx512/gcc-8.5.0/miniconda3-24.3.0-3zpgys4jvd5jttc4curqobgdnwpkxawb/condabin/conda
bash-4.4$
```

```
bash-4.4$ mamba
bash: mamba: command not found
bash-4.4$ conda search seqtk
Loading channels: done
No match found for: seqtk. Search: *seqtk*

PackagesNotFoundError: The following packages are not available from current channels:

  - seqtk

Current channels:

  - https://repo.anaconda.com/pkgs/main/linux-64
  - https://repo.anaconda.com/pkgs/main/noarch
  - https://repo.anaconda.com/pkgs/r/linux-64
  - https://repo.anaconda.com/pkgs/r/noarch

To search for alternate channels that may provide the conda package you're
looking for, navigate to

    https://anaconda.org

and use the search bar at the top of the page.


bash-4.4$ conda search -c bioconda  seqtk
Loading channels: done
# Name                       Version           Build  Channel
seqtk                            r75               0  bioconda
seqtk                            r82               0  bioconda
seqtk                            r82               1  bioconda
seqtk                            r93               0  bioconda
seqtk                            1.2               0  bioconda
seqtk                            1.2               1  bioconda
seqtk                            1.3      h5bf99c6_3  bioconda
seqtk                            1.3      h7132678_4  bioconda
seqtk                            1.3      h7132678_5  bioconda
seqtk                            1.3      h84994c4_1  bioconda
seqtk                            1.3      ha92aebf_0  bioconda
seqtk                            1.3      he4a0461_5  bioconda
seqtk                            1.3      he4a0461_6  bioconda
```

先创建环境再测试软件下载

```
bash-4.4$ conda create -y  -n blast4test
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /lustre/home/acct-stu/stu542/.conda/envs/blast4test



Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate blast4test
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

```
bash-4.4$ source activate blast4test
(blast4test) bash-4.4$ conda install seqtk -c bioconda -n blast4test
Channels:
 - bioconda
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): -
```

```
(blast4test) bash-4.4$ seqtk

Usage:   seqtk <command> <arguments>
Version: 1.3-r106

Command: seq       common transformation of FASTA/Q
         comp      get the nucleotide composition of FASTA/Q
         sample    subsample sequences
         subseq    extract subsequences from FASTA/Q
         fqchk     fastq QC (base/quality summary)
         mergepe   interleave two PE FASTA/Q files
         trimfq    trim FASTQ using the Phred algorithm

         hety      regional heterozygosity
         gc        identify high- or low-GC regions
         mutfa     point mutate FASTA at specified positions
         mergefa   merge two FASTA/Q files
         famask    apply a X-coded FASTA to a source FASTA
         dropse    drop unpaired from interleaved PE FASTA/Q
         rename    rename sequence names
         randbase  choose a random base from hets
         cutN      cut sequence at long N
         listhet   extract the position of each het
```

```
(blast4test) bash-4.4$ seqtk sample -s 2025 /lustre/share/class/BIO8402/lab/data/SRR5029637.fasta 50000 > /lustre/home/acct-stu/stu542/hw/test_class3/blast_time_test/SRR5029637_sub5w.fasta
(blast4test) bash-4.4$ grep ">"  /lustre/home/acct-stu/stu542/hw/test_class3/blast_time_test/SRR5029637_sub5w.fasta | wc -l
50000
(blast4test) bash-4.4$
```

```
(blast4test) bash-4.4$ conda deactivate
bash-4.4$ conda remove -n blast4test --all

Remove all packages in environment /lustre/home/acct-stu/stu542/.conda/envs/blast4test:


## Package Plan ##

  environment location: /lustre/home/acct-stu/stu542/.conda/envs/blast4test


The following packages will be REMOVED:

  _libgcc_mutex-0.1-main
  _openmp_mutex-5.1-1_gnu
  libgcc-ng-11.2.0-h1234567_1
  libgomp-11.2.0-h1234567_1
  seqtk-1.3-h5bf99c6_3
  zlib-1.2.13-h5eee18b_1


Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Everything found within the environment (/lustre/home/acct-stu/stu542/.conda/envs/blast4test), including any conda environment configurations and any non-conda files, will be deleted. Do you wish to continue?
 (y/[n])? y

bash-4.4$
```

```
[stu542@pilogin4 5w_test]$ scancel 42310193
[stu542@pilogin4 5w_test]$
```

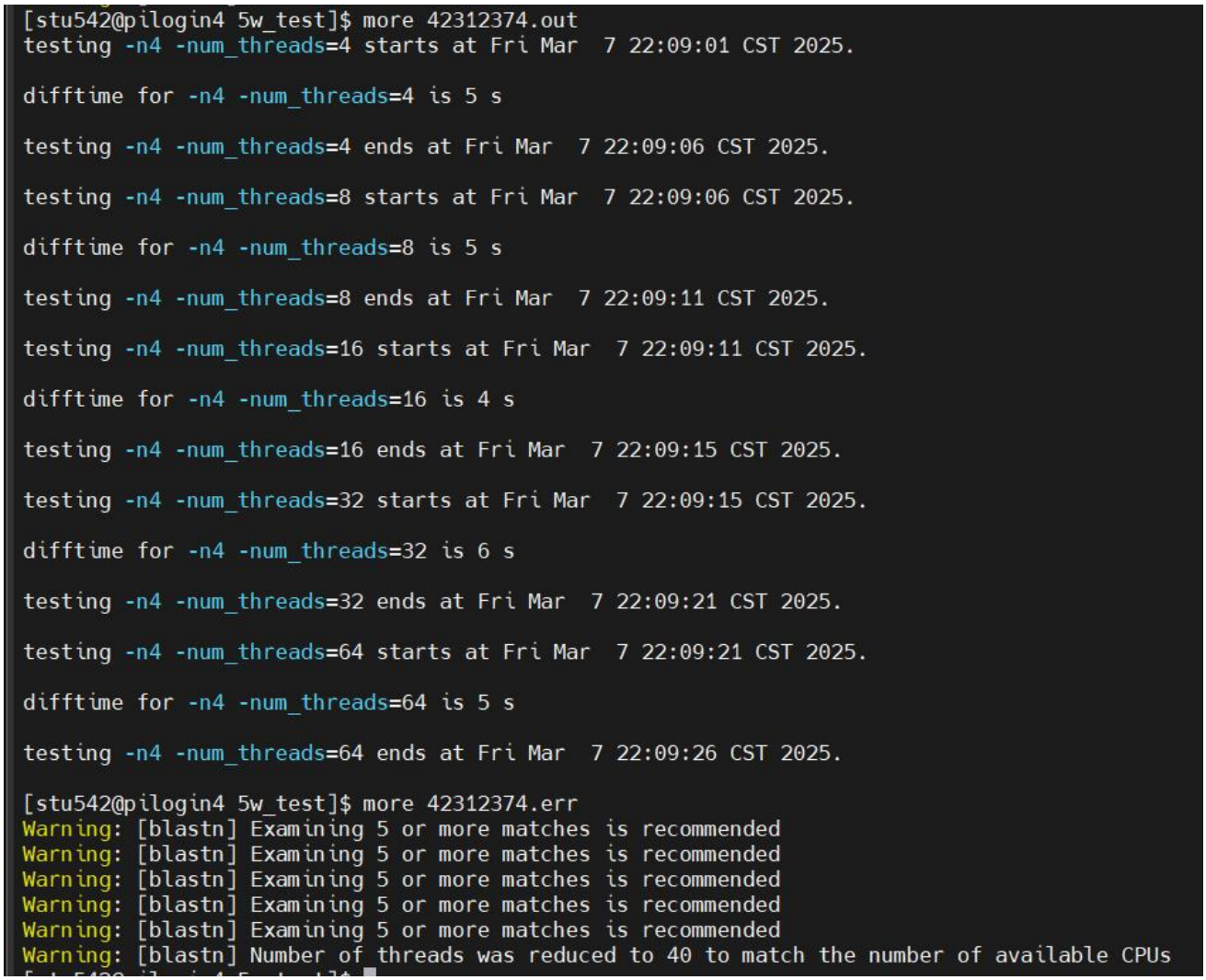
```
bash-4.4$ srun: Force Terminated job 42310193
srun: Job step aborted: Waiting up to 32 seconds for job step to finish.
slurmstepd: error: *** STEP 42310193.0 ON cas024 CANCELLED AT 2025-03-07T22:05:19 ***
exit
[stu542@pilogin4 ~]$
```

有了数据再提交

```
[stu542@pilogin4 5w_test]$ vim blast_test_5w.slurm
[stu542@pilogin4 5w_test]$ sbatch blast_test_5w.slurm
Submitted batch job 42312374
[stu542@pilogin4 5w_test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          42312374       cpu blast_5w   stu542 PD       0:00      1 (Priority)
```

```
[stu542@pilogin4 5w_test]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          42312374       cpu blast_5w   stu542  R       0:02      1 cas027
[stu542@pilogin4 5w_test]$
```

```
[stu542@pilogin4 5w_test]$ ls -lh
total 41M
-rw-rw-r-- 1 stu542 stu542  397 Mar  7 22:09 42312374.err
-rw-rw-r-- 1 stu542 stu542  879 Mar  7 22:09 42312374.out
-rw-rw-r-- 1 stu542 stu542  924 Mar  7 22:08 blast_test_5w.slurm
-rw-rw-r-- 1 stu542 stu542 8.9M Mar  7 22:09 n4t16.out
-rw-rw-r-- 1 stu542 stu542 8.9M Mar  7 22:09 n4t32.out
-rw-rw-r-- 1 stu542 stu542 8.9M Mar  7 22:09 n4t4.out
-rw-rw-r-- 1 stu542 stu542 8.9M Mar  7 22:09 n4t64.out
-rw-rw-r-- 1 stu542 stu542 8.9M Mar  7 22:09 n4t8.out
[stu542@pilogin4 5w_test]$
```



直到32线程数的数据还是可供参考的

个人脚本如下：

```
#!/bin/bash
#SBATCH --job-name=blast_5w
#SBATCH --partition=cpu
#SBATCH -n 4
#SBATCH --ntasks-per-node=4
#SBATCH --output=%j.out
#SBATCH --error=%j.err
#SBATCH --mail-type=end
#SBATCH --mail-user=zht161932@sjtu.edu.cn

module load blast-plus

threads_list=(4 8 16 32 64)

for threads in "${threads_list[@]}";do
  echo -e "testing -n4 -num_threads=$threads starts at $(date).\n"
  start=$(date +%s)
  blastn -query /lustre/home/acct-
stu/stu542/hw/test_class3/blast_time_test/SRR5029637_sub5w.fasta  \
```

```
    -db /lustre/share/class/BIO8402/lab/test/chr21 \
    -evalue 1e-5 \
    -outfmt 7  \
    -max_target_seqs 1 \
    -num_threads "$threads" \
    -out /lustre/home/acct-
stu/stu542/hw/test_class3/blast_time_test/5w_test/n4t${threads}.out
  end=$(date +%s)
  time=$(( end - start ))
  echo -e "difftime for -n4 -num_threads=$threads is $time s\n"
  echo -e "testing -n4 -num_threads=$threads ends at $(date).\n"
done
```

6)、运行太慢且不支持 MPI 怎么办？（10 分）

(What could you do if your program does not support MPI? 10 points)

根据 4 中的估计，原文件计算结束的时间估算约为 156—158 小时，且程序 blastn 不支持跨节点运算，在已经选择了计算速度最快的参数下，可以采取什么方法？请给出你认为合理的解决方法。

(According to the estimation in 4, the total time to finish the whole file is estimated to be 156-158 hours and program blastn does not support MPI, running in multiple nodes for a task. If you already picked the parameters for fatest speed to run it in a

node, what can you do to speed up the computing? Please give a reasonable solution for this.)

已经选择了计算速度最快的参数——假设对于线程数，以及队列中的线程数、cpu数等都已经在小数据试验中找到了最佳的参数，所以这一部分已经优化好了。

那其他还能优化速度的方法：

我们的任务需求本质上是单节点优化问题：

（1）单节点使用更强cpu、更大核心数的队列，例如192c6t队列

（2）既然 BLASTN 不支持跨节点并行，那就可以通过“数据并行”的方式来实现；

也就是将输入的 FASTA 文件拆分成多个较小的文件，每个文件独立运行 BLASTN，最后将所有结果合并

——》总之：

拆分输入文件：

将大的输入文件（如 subreads-A01.fasta）拆分成多个较小的文件。

每个小文件包含一定数量的 reads。

生成的文件名可以是 subreads-A01_part_1.fasta, subreads-A01_part_2.fasta, 等。

并行运行 BLASTN：

对每个小文件分别运行 BLASTN。

每个 BLASTN 任务使用相同的参数。

每个任务在不同的节点或核心上运行。

合并结果：

将所有小文件的比对结果合并成一个文件

```
#==================================================================
#
#   以下内容供同学们未来使用系教学集群时参考（任务递交系统为 torque）
#   For someone use our department computer cluster for this course
#    (the task management system is torque, not slurm)
#==================================================================
```

3. 教学集群访问

   （见课堂 PPT，如果不清楚，请联系课程教师或助教）

4. 教学集群介绍（更详细材料参考教学集群使用说明 PDF 文件）

   a. 集群组成：16 个计算节点，452 个核，14 个 GPU

      1 个管理登录节点mu01

      4 个胖（大内存）节点：fat01-fat04

      4 个 GPU 计算节点：gpu01-gpu04

      8 个 CPU 计算节点：cu01-cu08

      IO 节点：io01-io02

      挂载地址/home，总存储 692TB

b. 胖节点信息

- fat01

Intel(R) Xeon(R) Gold 5115 CPU @ 2.40GHz，4 路 10 核

2.0TB 内存

- fat02-fat04

Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz，双路 10 核

512GB 内存

c. GPU 节点信息

- gpu01

Intel(R) Xeon(R) Gold 5115 CPU @ 2.40GHz，双路 10 核

64GB 内存

2*Tesla V100-PCIE-16GB：5120 流处理器+16GB 显存

- gpu02-gpu04

Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz，双路 10 核

64GB 内存

4*GeForce GTX 1080 Ti：3584 流处理器 + 11GB 显存

d. CPU 节点信息

- cu01-cu04

Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz，双路 16 核

64GB 内存

- cu05-cu08

Intel(R) Xeon(R) Gold 6150 CPU @ 2.70GHz，双路 18 核

256GB 内存

e. Environment Module

- 可用 Environment Modules 切换多个版本的工具软件
- 查看可用软件及版本：module avail
- 加载工具：module load
- 卸载工具：module unload
- 查看当前加载工具：module list

```
[pg019080910011@mu01 fastqc]$ module avail
------------------------------------------------------- /home/apps/modules/etc/modulefiles ------------------
anaconda/3  chem/amber18  chem/gromacs-2019.3  cmake/3.18.2  mpi/mpich-3.2-x86_64  ngs/assemblers  R/3.6.0
bioinfo     chem/gromacs  cmake/3.13.4         cuda/8.0      mpi/openmpi-4.0.0     ngs/basic
[pg019080910011@mu01 fastqc]$
[pg019080910011@mu01 fastqc]$
[pg019080910011@mu01 fastqc]$ module load anaconda/3
```

4. 作业调度系统常用命令

qsub（提交作业）

- **qsub** 常用参数
- -l 指定作业所需要的资源，设定对可消耗资源的限制。如果不设置，则无限制。
- 1）-l mem=220g,vmem=220g（根据文件大小、文件处理方式、计算复杂情况自行估计）
- mem ：任务的所有进程能够分配到的最大物理内存数；
- vmem：任务的所有进程能够使用的最大虚拟内存数；
- 2）-l nodes=fat03 指定节点名字
- 3）-l nodes=1:ppn=10 指定使用节点数及线程数
- 4）-l walltime=144:00:0 指定运行最长时间，walltime 指钟表时间（作业的实际运行时间=walltime/线程数）
-

qstat（查看作业状态，作业结束一段时间后不能查看）

```
[pg019080910011@mu01 fastqc]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
34849.mu01                 qc               pg019080910011         0 R cpuq
```

```
[pg019080910011@mu01 test]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
34849.mu01                 qc               pg019080910011  00:00:51 R cpuq
```

Job ID：作业号
Name：作业名称
User：用户名称
Time Use：作业运行时间
S：作业状态（R 运行、Q 排队、C 终止）
Queue：使用队列
（注意：当 S 为 R 状态时，作业不一定在运行，只有当 Time Use 不为 0 时，作业才真正 running）

qdel（**kill** 作业）
用法：qdel + jobid
tracejob（查看历史作业信息，作业结束之后仍能查看）
用法：tracejob –n 天数 job_id
示例：1）tracejob –n 10 106972　　查找 10 天内的记录，寻找 id 为 106972 的作业信息
2）tracejob 106972　　不加时间参数，查找当天作业
qnodes 或者 **pbsnodes**（查看节点状态）
根据节点可用情况指定作业的提交节点
示例：qnodes | grep job_id　　查看作业分配的节点

请在你自己的工作目录下（如**/home/pg2022/pg122440910006/omics2023**）进行第 **5-7** 项练习。

5. 作业调度系统脚本示例。脚本模板如下。

```
#!/bin/bash

#PBS -N XX                      作业名称
#PBS -q XX                      队列名称必须指定
#PBS -l nodes=1:ppn=40          作业使用1个节点，分配40个核
#PBS -o XX                      输出日志文件
#PBS –e XX                      标准错误文件
blastn -task blastn -query ./subreads-A01.fasta -db ../index -evalue 1e-5 -outfmt 7
–max_target_seqs 1 -num_threads 16 -out ./16.blastn
```

6.

请在你自己熟悉的编辑器中按如上模板编辑如下脚本，保存为test.pbs

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -N qc
#PBS -o qc.out
#PBS -e qc.err
#PBS -M ccwei@sjtu.edu.cn
#PBS -q cpuq

cd /home/faculty/ccwei/courses/2023/omics/lab/test
module load anaconda
fastqc -t 40 -o ./ /home/faculty/ccwei/courses/2023/omics/lab/data/SRR5811639.fastq.gz
```

注意事项：

1）脚本命名：*test.pbs*

2）cd 命令后请加绝对路径，以免结果文件位置不明。

递交该 blast 作业到集群上运行：

qsub test.pbs

7. RNA 序列比对——以 blast 为例

建库+比对

```
#! /bin/bash
#PBS -l nodes=1:ppn=40
#PBS -N blast
#PBS -o blast.out
#PBS -e blast.err
#PBS -q cpuq

module load bioinfo
cd /home/pg2019/pg019080910011/tmp/

makeblastdb -in /home/faculty/ccwei/courses/2020/omics/proj1_human/chr/chr21.fa \
   -dbtype nucl -out ./chr21

blastn -query ./SRR5231434.fasta  -db ./chr21 -evalue 1e-5 -outfmt 7 \
   -max_target_seqs 1 -num_threads 40 -out ./result
```