FY2025
ICR Supercomputer System Usage Manual
For Bioinformatics Users

Hewlett Packard Enterprise, Japan.
Hideya Uehara

spradm@scl.kyoto-u.ac.jp

0774-38-3265　(Extension: 3265)
https://www.scl.kyoto-u.ac.jp/

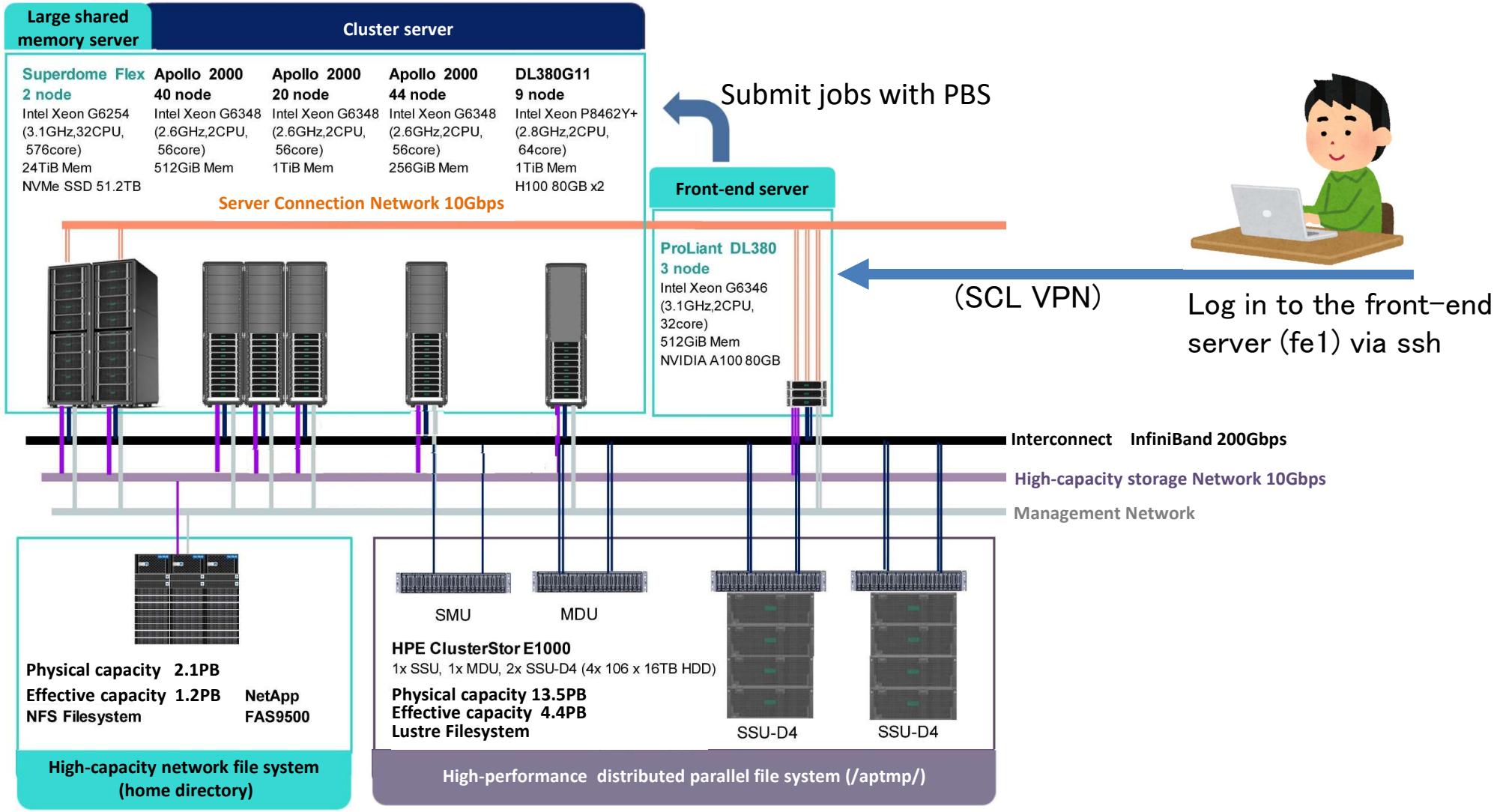
Co-organized by
Bioinformatics Center, Institute for Chemical Research, Kyoto University,
NPO Bioinformatics Japan

ver 1.0.3en

Contents

1 System Overview

1.1 System configuration



1.2 Server Spec

	Large shared memory server	Cluster server (CPU node)	Cluster server (GPU node)	Front-end server
model	HPE Superdome Flex	HPE Apollo2000 G10+	HPE DL380 G11	HPE DL380 G10+
# of nodes	2	104	9	3
hostname	sdf1, sdf2	csXX, cmXX, clXX	ch1～ch9	fe1
CPU	Intel Xeon G6254 3.1GHz x 32CPU (18cores/CPU)	Intel Xeon G6348 2.6GHz x 2CPU (28cores/CPU)	Intel Xeon P8462Y+ 2.8GHz x 2CPU (32cores/CPU)	Intel Xeon G6346 3.1GHz x 2CPU (16cores/CPU)
# of cores/node	576	56	64	32
memory size /node	24TiB (42GiB/core)	256GiB (cs01～cs44) 512GiB (cm01～cm40) 1TiB (cl01～cl20)	1TiB	512GiB
GPU	-	-	NVIDIA H100 x 2 (80GB)	NVIDIA A100 (80GB)
SSD	50TB	-	-	-
OS	Red Hat Enterprise Linux(＊) 8.7			

(＊) A paid version of CentOS that runs the binary program for x86_64 (amd64) Linux.

2 Terms of Use

2.1 Applying for a New Account

- Please get help from your Japanese colleagues.

  Contact Information
  Email: spradm@scl.kyoto-u.ac.jp

- Research results report (users of compute server)
  - ➢ You must submit a research results report （either in February or when you stop using the supercomputer system).
  - ➢ The following users are exempt from submitting the report.
    - Those who belong to a private company.
    - Those who applied after January.
    - Those who did not use compute servers or applications at all.
    - Those who wish to withhold publication due to a patent application or submitting a paper.

2.2 Usage Fee

<table>
<tr><th>Target</th><th>Calculation Rule</th></tr>
<tr><td>Basic fee</td><td>￥1,000/month</td></tr>
<tr><td>Compute node (CPU time)</td><td>￥0.00220/sec</td></tr>
<tr><td>Upper Limit of Charges for CPU Time</td><td>
<table>
<tr><th>Affiliation of the Person Responsible for Payment</th><th></th></tr>
<tr><td>Institute for Chemical Research, Kyoto University</td><td>¥40,000 / month</td></tr>
<tr><td>Kyoto University (excluding the Institute for Chemical Research)</td><td>¥60,000 / month</td></tr>
<tr><td>Academic institutions (excluding Kyoto University)</td><td>¥80,000 / month</td></tr>
<tr><td>Private organizations</td><td>¥200,000 / month</td></tr>
</table>
(CPU Node: Approximately ¥10,000 per node per day)
</td></tr>
<tr><td>Home directory</td><td>Free (upto 100GB available)</td></tr>
<tr><td>Cloud storage service</td><td>Free (upto 100GB available)</td></tr>
<tr><td colspan="2">Optional Service</td></tr>
<tr><td>Home directory expansion service</td><td>￥1,000/fiscal year for +300GB (upto +900GB can be added)</td></tr>
<tr><td>Cloud storage expansion service</td><td>￥1,000/fiscal year for +300GB (upto +900GB can be added)</td></tr>
</table>

- Temporary space (/aptmp/(username)/) is not charged (and no capacity limit).

2.3 How to Log in to the Supercomputer System

- To use the supercomputer system, log in to the front-end server fe1.
  - (Linux, Mac) In a terminal

```
$ ssh username@fe1.scl.kyoto-u.ac.jp
```

(× login.scl.kyoto-u.ac.jp)

(Windows) Terminal software (Command Prompt, MobaXterm, TeraTerm, Putty, RLogin, etc.)

  - To use GUI applications (seaview, LibreOffice, gnuplot, FigTree, IGV, etc) on fe1.
    - (Linux) Log in via ssh –Y(C).
    - (Mac) Install Xquartz and log in via ssh –Y(C).
    - (Windows) Install MobaXterm (recommended) or Xming.

- Changing the login shell (command line interface)
  - You can select csh, tcsh (default), bash, or zsh.

```
$ ldapchsh bash      # switch to bash (please run it on the front-end server fe1).
```

2.3 How to Login to the Supercomputer System(Continued)

- To log in to the front-end server (fe1) from a network outside Kyoto University (KUINS), please use VPN.
  - (Windows, Linux, Mac) https://vpn.scl.kyoto-u.ac.jp/
  - (Mac) Install F5Access from the Mac App Store.

- To transfer files between your PC and the front-end server, use the following software or commands.
  (Windows) WinSCP, FileZilla, MobaXterm
  (Mac) Cyberduck, FileZilla
  (Linux) scp, sftp, rsync
  - Beware of line feed codes!
    A text file created on Windows PC may cause error when you executing it on the supercomputer system, because line feed codes are different between Windows and Linux.
    (Example)
    '¥r' : command not found
    ^M: command not found
  - In such a case, use dos2unix to convert line feed codes in the text file.
    `$ dos2unix (file)`

2.3 How to Login to the Supercomputer System(Continued)

- Each job on the frontend server (fe1) is limited to **30 minutes of CPU time** and **8 GB of memory**.
- Please use the **PBS interactive batch job** feature to work interactively on the compute servers (§ 4.7).
- As a general rule, **do not log in directly to the compute servers** via ssh.
  (However, it is acceptable to log in briefly to check job status using commands such as **top**.)

2.4 Disk Space

	Home directory	Temporary space	SSD
Location	~(username)/	/aptmp/(username)/	${TMPDIR}/
File system	NFS	Lustre	xfs
Charge	✔	-	-
Capasity limitation	✔	-	-
Snapshot	✔	-	-
Shared?	✔	✔	-
Remark			Available only in SDF queue (§ 4.11)

2.4.1 Home Directory

- Snapshot
  - Every directory in your home directory contains a hidden directory .snapshot/(date-time)/

```
$ ls .snapshot/
daily.2022-05-09_0010/  weekly.2022-05-01_0015/  weekly.2022-05-08_0015/
```

    - You can restore accidentally deleted or modified files from .snapshot (use cp command as usual).
    - .snapshot/ is not shown even with ls -a ! (Tab completion doesn't work either).
    - You cannot delete or modify files in snapshot/.

- Checking disk usage (quota command)

```
$ quota -s
Disk quotas for user lect-1 (uid 10151):
     Filesystem  space  quota  limit  grace  files  quota  limit  grace
fas9500-03_NFS:/HOME/user1/
                72366M  81920M   100G         6870   1800k  2000k
```

Current disk usage

Disk usage limit

Current total number of files

Limit on the total number of files

2.4.2 Temporary Space (/aptmp/(username)/)

- Use this area for calculation.
- Not charged.
- No limitation on disk usage or storage period.
  - ➢ You may be asked to organize (compress or delete) your files/directories, if available disk space is getting low.
  - ➢ Pay attention not only to the file size, but also to the number of files (and directories).
- Disk usage

```
$ lfs-quota -h
Disk quotas for usr lect-1 (uid 10151):
    Filesystem   used  quota  limit  grace   files  quota  limit  grace
      /lustre/ 512.1G     0k     0k      -  203327      0      0      -
```

Current disk usage

Current total number of files

- The snapshot is not available on /atpmp/. Please take backups on your own.

2.4.3 Checking the Directory Size and File Count (1/2)

➢ Directory size (Disk usage)

- du command

```
$ du -ks (directory)/      # Total size of (directory)/ [kb]
$ du -kS (directory)/      # Total size of files and directories under each directory in (directory)/[kb]
$ du -hS (directory)/ | sort -h    # sorts by directory size
```

- mpifileutils (for a directory containing a huge numbers of files) New!

```
$ module load mpifileutils
$ mpirun -np 8 dwalk (directory)/ --no-atime
...
[2026-04-27T18:45:34] Walked 101117373 items in 2613.485 seconds (38690.629 items/sec)
[2026-04-27T18:45:35] Items: 101117373
[2026-04-27T18:45:35]   Directories: 10894223
[2026-04-27T18:45:35]   Files: 84023121
[2026-04-27T18:45:35]   Links: 6200029
[2026-04-27T18:45:35] Data: 86.231 TiB (1.076 MiB per file)
```

2.4.3 Checking the Directory Size and File Count (2/2)

- File count
  - du command

```
$ du --inodes -s (directory)/                # Number of all files and directories in (directory)/.
$ du --inodes -S -t 10000 (directory)/    # Show only directories that have more than 10,000 files  in them.
```

  - find command

```
$ find (directory)/ | wc -l                 # Number of all files and directories in (directory)/.
$ find (directory)/ -type f | wc -l      # Number of all files in (directory)/.
$ find (directory)/ -type d | wc -l      # Number of all directories in (directory)/.
```

  - mpifileutils (for a directory containing a huge numbers of files)

```
$ module load mpifileutils
$ mpirun -np 8 dwalk (directory)/ --lite (--no-atime)
…
[2026-04-27T18:45:34] Walked 101117373 items in 2613.485 seconds (38690.629 items/sec)
[2026-04-27T18:45:35] Items: 101117373
[2026-04-27T18:45:35]   Directories: 10894223
[2026-04-27T18:45:35]   Files: 84023121
[2026-04-27T18:45:35]   Links: 6200029
```

2.5 Web Applications

(Caution) 'Logout' button in the web mail and the cloud storage doesn't work.

2.6 System Load Information

- Top page of the supercomputer system

← Detailed information for each node (supercomputer account required)

- To check the real time usage of the system, use qstatmyjobs command（§4.9）.

```
fe1{lect-1}1001: qstatmyjobs -q
User: lect-1
                           JOBS            CPUS                     MEM(gb)                  GPUS           WALLTIME(h)
Queue   avail(use%) mysum/max   avail max mysum/max    avail    max  mysum/max   avail max mysum/max   default max
------  ----------- ----------- --------------------- ----------------------------- --------------------- -------------
SMALL     870( 85%)    0/UNLTD     12  12      0/96       48     48     0/UNLTD     -    -      -/-           6  12
APC       702( 87%)    0/UNLTD     21  56      0/UNLTD    12    980     0/UNLTD     -    -      -/-        2880 UNLTD
APG       480( 16%)    0/UNLTD     64  64      0/UNLTD   980    980     0/UNLTD     2    2      0/UNLTD    2880 UNLTD
SDF       608( 39%)    0/8          - 144      0/288       - 12288     0/UNLTD     -    -      -/-        2880 UNLTD
=====================================================================================================================
TOTAL:              JOBS)    0/UNLTD        CPUS)   0/500           MEM)      0/18432        GPUS)   0/4
```

3 Using applications

3.1 module command

- Applications installed in the supercomputer system are managed by the **module command.**

```
$ module avail                          # lists all the available applications.
$ module avail -L                       # lists only the latest version of each applications.
$ module avail bl                       # lists only the applications whose names start with bl.

$ module load [-s] (module name)    # loads the module file of the application (-s: less loading message).
                                        Some dependent module files may be loaded at the same time.

$ module list                           # show currently loaded modules
$ module switch [-f] (module1) (module2)   # switches the loaded modules.
                                                (add -f option if you cannot switch due to some error.)
$ module unload (module name)           # unloads specified module.
$ module purge                          # unloads all modules.
```

3.1 module command (continued)

- Please include the following line at the beginning of a shell script (PBS job script).
  source /etc/profile.d/modules.sh (sh/bash script)
  source /etc/profile.d/modules.csh (csh/tcsh script)

- Some applications may conflict with each other if they are loaded simultaneously. So, please unload (or purge) module files after each application run.

```
#!/bin/sh
source /etc/profile.d/modules.sh

module load prog1
prog1 xxxx
module purge

module load prog2
prog2 yyyy
module purge
```

3.2 Bioinformatics Applications

- List of installed bioinformatics applications
  https://www.scl.kyoto-u.ac.jp/Appli/#biotool

- Bioinformatics application used in the GenomeNet service

Name	Description	Module name
BLAST+	Homology search	blast+
Clustal Omega	Multiple alignment	clustal-omega
ClustalW2	Multiple alignment	clustalw2
DBGET	Database search system	dbget
ETE Toolkit	Phylogenetic tree construction	ete
FASTA	Homology search	fasta
FastTree	Phylogenetic tree construction	FastTree
FastTreeMP	Multi-thread version of FastTree	FastTreeMP
ghostx	Homology search	ghostx
ghostz	Homology search	ghostz
HMMER	Motif search	hmmer
MAFFT	Multiple alignment	mafft
MUSCLE	Multiple alignment	muscle
PHYLIP	Phylogenetic tree construction	phylip
PRRN	Multiple alignment	prrn
raxml	Phylogenetic tree construction	raxml
SIMCOMP	Compound structure search	simcomp
ssearch	Homology search	ssearch
SUBCOMP	Compound substructure search	subcomp

3.3 Bioinformatics Database

- Some bioinformatics databases are installed in the following directories.
  /db/(type)/(name)/
  - type: blast, bowtie, bowtie2, dbget, diamond, fasta, ghostx, hmmer, motif, rpsblast
  - name: genbank, refseq, mgenes, ncbi, swissprot, trembl, pfam, etc.

  (example) NCBI nr BLAST database: /db/blast/ncbi/nr.*
- Database release information: /db/dbinfo.txt
- The entity of /db/ is located in /lustre/db/YYYYMMDD/ .
  - Directory older than one month will be deleted, except for the first one of each month.
  - The first one of each month is kept in compressed format (.tpxz).
    - How to extract files from .tpxz compressed file.
      /lustre/db/00_How_to_extract_files.txt
      /lustre/db/extract.sh
- Mirrors of FTP sites from NCBI, EBI, and PDBj
  - /db/ftp.ncbi.nih.gov/ # genbank, refseq, genomes, taxonomy, etc.
  - /db/ftp.ebi.ac.uk/ # uniprot, uniref, etc.
  - /db/ftp.pdbj.org/ # PDB
  - /db/logan/ # Logan Project

3.3 Bioinformatics Databases (continued)

- Bioinformatics Databases

Database	Description	Directory Name
GenfBank	Nucleotide sequence database	genbank
GenBank-upd	Nucleotide sequence database	genbank-upd
GenPept	Nucleotide sequence database	genpept
GenPept-upd	Nucleotide sequence database	genpept-upd
RefSeq	Nucleotide and amino acid sequence database	refseq
MGENES	Metagenome sequences	mgenes
NR-NT	Non-redundant nucleotide sequence database	nr-nt
NR-AA	Non-redundant amino acid sequence database	nr-aa
NCBI BLAST database	BLAST database released by NCBI (nr, nt, swissprot, refseq_protein, taxdb, etc)	ncbi
UniProt/Swiss-Prot	Amino acid sequence database	swissprot
UniProt/TrEMBL	Amino acid sequence database	trembl
UniRef	Amino acid sequence database	uniref
dbEST	EST(Expressed Sequence Tags) sequence	dbest
dbGSS	GSS(Genome Survey Sequences) sequence	dbgss
dbSTS	STS(Sequence Taged Sites)) sequence	dbsts
Silva (*)	Ribosomal RNA sequence database	silva
RDP	Ribosomal RNA sequence database	rdp
PR2	Ribosomal RNA sequence database	pr2
PDBSTR	PDB amino acid sequences	pdbstr
Pfam	Protein Domain Family	pfam
NCBI CDD	NCBI Conserved Domain Database	ncbi-cdd

(*) academic use only

4. PBS Job Scheduler

4.1 What is a Job Scheduler?

- In a shared computer system, a job scheduler is used so that multiple users can use computing resources as fairly as possible.
- In our supercomputer system, Altair's PBS Professional is used as a job scheduler.
- Submit a job file using qsub command from fe1 (the front-end server).
- Number of cores and memory available to a user.

Total number of cores for concurrently executed jobs (soft limit＊)	500 (300)
Total memory size for concurrently executed jobs (soft limit＊)	18TB (3TB)

＊ When the number of cores exceeds the soft limit, the scheduling priority will be lowered.

- Maximum number of jobs that can be submitted simultaneously: 10,000 (including sub-jobs of array jobs)

4.2 Job Submission

$ qsub [options] (job script file)

- options
  - −q xxxx queue name (SMALL, APC, APG, SDF)
    * Different queue have different available compute node, resources, and job priorities.
  - −l xxxx resources required by the job
    - resources
      - » select=1:ncpus=(# of cores):mem=(memory size)
        (In case ncpus and mem are not specified, the default values for the queue are used.)
      - » Add ngpus=1 or ngpus=2 to use GPU.
      - » Add −l walltime=(HH):(MM):(SS) to specify the maximum elapsed time for a job.
  - −N xxxx job name
  - −o xxxx standard output file
  - −e xxxx standard error file
  - −j oe margining standard error into standard output
- Job ID will be issued when a job is submitted by qsub command.
  - If you have trouble running a job, please contact us **with the Job ID**.

4.3 Queue List

Queue name	SMALL	APC	APG	SDF
# of compute servers	Cluster (CPU): 104	Cluster (CPU): 101	Cluster (GPU): 9	Large shared memory: 2
Job type	Small job	Normal job	GPU job(※2)	Large memory job
Queue priority	70	50	100	90
Limitations per job				
Max. # of cores (default)	12 (1)	56 (※1) (1)	64 (1)	144 (18)
Max. memory (default)	48 GB (4 GB)	980 GB (※1) (4 GB)	980 GB (4GB)	12 TB (768 GB)
Max. elapsed time (default)	12 h (6 h)	No limit (2880 h)	No limit (2880 h)	No limit (2880 h)
Limitations per user				
Max. # of jobs (soft limit)	-	-	4 (1)	8 (4)
Total # of cores (soft limit)	96	500	-	288 (144)
Total memory (soft limit)		9.8 TB	-	14 TB (6 TB)

(※1) You can use 500 cores and 9.8TB memory in a MPI job using multiple nodes.
(※2) There is a limit of up to 2 GPUs per job and up to 4 GPUs per user.

4.3 Queue List (continued)

Queue name	cdb
# of compute servers	gds1, gds2
Job type	GenomeNet Service Developler
Queue priority	50
Limitations per job	
Max. # of cores (default)	58 (1)
Max. memory (default)	980GB (20 GB)
Max. elapsed time (default)	No limit (720 h)
Limitations per user	
Max. # of jobs	32
Total # of cores	32
Total memory	1.3TB

- This queue is available only to users involved in the development of GenomeNet.
- The home directory of gdsX is different from that of fe1 (/aptmp/ is common).
- Jobs executed directly on gdsX without PBS (cdb queue) have a CPU time limit of 30 minutes.
- Of gdsX's 64 cores and 1TB memory, cdb queue can be used up to 58 cores and 980GB in total.
- If you need to run a large job that will take more than 1 hour of computation time, please be sure to inform other users and obtain their approval.
- When running jupiter, etc., please use an interactive batch job ("qsub -I -q cdb ..."). To avoid occupying unused cores, terminate jobs during unused hours, such as at night (or specify -l walltime=...)

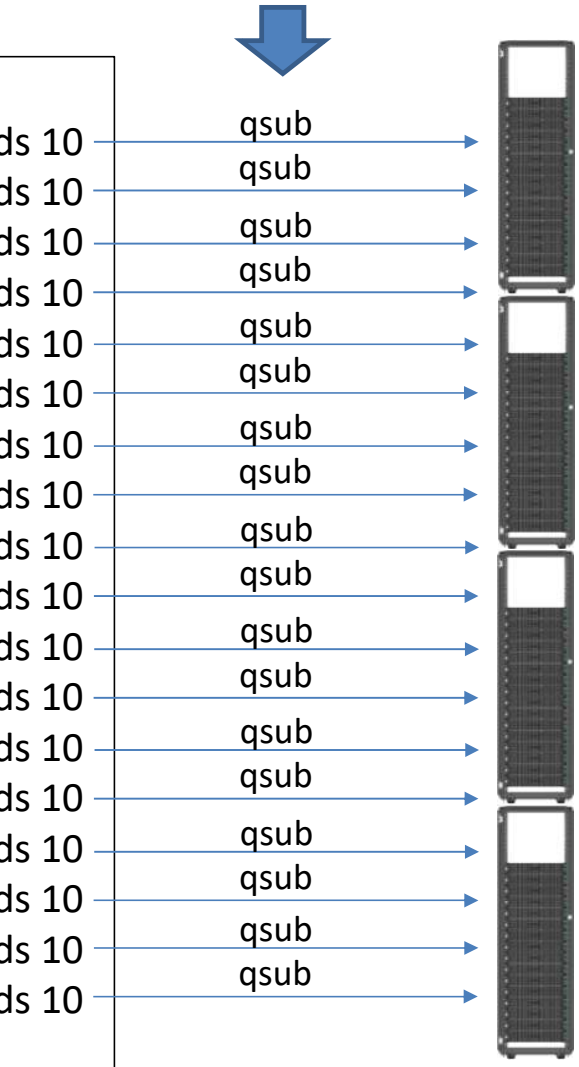
4.4 Job Script Example

```
#!/bin/sh
#PBS -q APC                                ← queue name
#PBS -l select=1:ncpus=10:mem=40gb         ← specify the number of cores and memory size
#PBS -N test                               ← the job name
#PBS -o test.out                           ← standard output file
#PBS -e test.err                           ← standard error file

source /etc/profile.d/modules.sh           ← enable the module command
module load blast+/2.15.0                  ← load blast+ command
cd $PBS_O_WORKDIR/                         ← go to the directory where qsub was executed.

blastp -db db/nr -query query.fa -out result.out -outfmt 7 -num_threads 10
```

- You can specify the options for qsub command as
  #PBS xxxx
  in the job script.
- When running a multi-thread command, explicitly specify the number of cores with the command option, even if only one core is used.

4.5 Processing a large number of jobs / input files

```
$ cat blastp.com
blastp -db db/nr -query query01.fa -out result01.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query02.fa -out result02.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query03.fa -out result03.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query04.fa -out result04.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query05.fa -out result05.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query06.fa -out result06.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query07.fa -out result07.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query08.fa -out result08.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query09.fa -out result09.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query10.fa -out result10.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query11.fa -out result11.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query12.fa -out result12.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query13.fa -out result13.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query14.fa -out result14.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query15.fa -out result15.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query16.fa -out result16.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query17.fa -out result17.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query18.fa -out result18.out -outfmt 7 -num_threads 10
…
```



(＊) This is a command unique to our supercomputer system, not a genuine PBS command.

4.5 Processing a large number of jobs / input files (Continued)

$ qsubarraypbs(*) [option] (a jobs file)

- This command uses the PBS array job feature to execute a large number of jobs listed in a jobs file in parallel.

- Options (basically the same as the qsub command)
  - -q (queue name) … SMALL, APC, APG, SDF (required)
  - -l select=1:ncpus=(number of cores per job):mem=(memory per job)
  - --max (maximum number of simultaneous jobs)
  - Other qsub options: -N, -r, -v, -l, -p, -P, -W

- A jobs file
  - Write **one job per line** (you may include multiple commands on a line by separating with ; or &&).
  - If you want to redirect the output of the command to files, use standard shell (sh/bash) syntax:
    (command) 1>xxx.out 2>xxx.err
  - Lines beginning with # are comment lines and ignored (#PBS directives are not supported in qsubarraypbs).

4.5 Processing a large number of jobs / input files (Continued)

➢ Notes

- Before running the **qsubarraypbs** command, please load the module file for the application you intend to execute using **module load**.
- When using the SDF queue:

```
$ module load SDF
$ module load (program)
$ qsubarraypbs -q SDF -l select=1:ncpus=xxx:mem=xxx …  (a job file)
$ find /usr/appli/sdf/modulefiles/ -type f     # Programs available for SDF binaries
```

- Inside qsubarraypbs, the working directory is automatically changed to $PBS_O_WORKDIR/.
- Error messages and failed jobs are output to the pbslog/ directory.
- PBS mail notification options (-M, -m) are not available.
- The job file must contain no more than 10,000 lines (due to PBS array job limitations).
- Changing the number of simultaneous jobs

```
$ qalter -Wmax_run_subjobs=(number of concurrent jobs) (JobID)
```

Example:

```
$ qalter -Wmax_run_subjobs=10  2225228[].fe3-adm
```

4.6.1 Splitting an input file

- Splitting a file by a specified number of lines

```
$ wc  -l  (input file)            # count all lines
$ split -l (number of lines to split) -d -a 4 (input file) (output file name)
        # -d  -a 4  … adds 4-digit sequential number to the output file name
```

- Splitting a FASTA file

```
$ fasta_split  (input FASTA file)  (number of splits)
```

```
$ module load fasta-splitter
$ fasta-splitter --n-parts (number_of_parts) (FASTA_file)     # Split into parts of approximately equal file size
$ fasta-splitter --part-size (part_size) (FASTA_file)
```

```
$ module load seqkit
$ seqkit split  -O ./  -p (N)  (FASTA_file)    # Split into N parts with an equal number of sequences
$ seqkit split  -O ./  -s (N)  (FASTA_file)    # Split every N sequences
$ seqkit split2 -O ./  -l (N)  (FASTA_file)    # Split every N residues
```

- Split FASTQ file

```
$ module load fastq-splitter
$ fastq-splilter --n-parts (number of splits) (FASTQ file)    # specify number of splits (--n-parts) or
                                                                split size (--part-size)
```

4.6.2 Creating a sequentially numbered jobs

- seq : output sequential numbers.
- xargs: replace {} with a string read from STDIN and execute the command.

```
$ seq -w 1 10 | xargs -i echo “command sample{}.pep“ >com.txt
$ cat com.txt
command sample01.pep
command sample02.pep
command sample03.pep
command sample04.pep
command sample05.pep
command sample06.pep
command sample07.pep
command sample08.pep
command sample09.pep
command sample10.pep
```

4.6.3 Creating a jobs file from a list of input files

- GNU parallel: execute commands by substituting strings (more powerful than xargs)

```
$ find /db/fasta/mgenes/T*.pep | parallel -k --dry-run "{} {.} {/} {//} {/.} {#}"
/db/fasta/mgenes/T30001.pep /db/fasta/mgenes/T30001 T30001.pep /db/fasta/mgenes T30001 1
/db/fasta/mgenes/T30002.pep /db/fasta/mgenes/T30002 T30002.pep /db/fasta/mgenes T30002 2
/db/fasta/mgenes/T30003.pep /db/fasta/mgenes/T30003 T30003.pep /db/fasta/mgenes T30003 3
/db/fasta/mgenes/T30004.pep /db/fasta/mgenes/T30004 T30004.pep /db/fasta/mgenes T30004 4
/db/fasta/mgenes/T30005.pep /db/fasta/mgenes/T30005 T30005.pep /db/fasta/mgenes T30005 5
...

The following commands give the same result.
$ parallel --dry-run "{} {.} {/} {//} {/.} {#}" ::: /db/fasta/mgenes/T*.pep
```

{}	{.}	{/}	{//}	{/.}	{#}
displayed as is	removing extension	file name	directory name	file name without extension	serial number

Example)

```
$ find /db/fasta/mgenes/T*.pep | parallel -k --dry-run "command {} 1>{/.}.out 2>{/.}.err"
>com.txt
$ cat com.txt
command /db/fasta/mgenes/T30001.pep 1>T30001.out 2>T30001.err
command /db/fasta/mgenes/T30002.pep 1>T30002.out 2>T30002.err
command /db/fasta/mgenes/T30003.pep 1>T30003.out 2>T30003.err
…
```

4.7 Interactive Batch Job

- PBS's interactive batch job feature allow you to log in to a compute node and work interactively on that node.
  - Add the –I option (「I」ntaractive) to qsub.
  - Please use interactive batch jobs only in the SMALL queue whenever possible (to prevent users from remaining logged in continuously).

```
[appadm@fe1]$ qsub -I -q SMALL -l select=1:ncpus=10:mem=30gb -l walltime=12:00:00
qsub: waiting for job 205274.fe3-adm to start
qsub: job 205274. fe3-adm ready

cd /scratch/pbs_jobdir/pbs.205274.fe3.x8z
[appadm@cs18]$ cd /scratch/pbs_jobdir/pbs.205274. fe3-adm.x8z
[appadm@cs18]$ cd $PBS_O_WORKDIR
```

4.8 Show job status, and delete jobs

- Show status of jobs and queue
  $ qstat [option] (Job ID or queue)
  - options
    - −x # show finished jobs
    - −f [Job ID] # full format
    - −t [Job ID] # show each sub−jobs of an array job
    - −n1 [Job ID] # show the hostname on which the job is running
    - −r # show only running jobs
    - −q, −Q # show status of all queues
- Delete jobs
  $ qdel [Job ID] [Job ID …]

＊ Job status
Q: Waiting for execution
R: Running
E: Exiting (transition from 'R' to 'F')
F: Finished job
H: On hold
S: Suspending
B: Array job running
X: Finished sub−jobs of an array job

- Hold/unhold job execution (to temporarily stop an array job submitted by qsubarraypbs command)
  $ qhold [Job ID] # hold a job execution (only for a job with status Q)
  $ qrls [Job ID] # release a holded job

4.9 qstatmyjobs command（＊）

- Check the number of jobs, cores and amount of memory used by the user

```
$ qstatmyjobs
User: appadm
                     JOBS          CPUS                 MEM(gb)                GPUS        WALLTIME(h)
Queue  avail(use%) mysum/max   avail max mysum/max    avail   max mysum/max   avail max mysum/max  default max
------ ----------- ----------- --------------------- ----------------------- -------------------- -------------
SMALL   1193( 76%)     0/UNLTD    12  12      0/96      48     48     0/UNLTD     -   -     -/-          6 12
APC     1025( 78%)    19/UNLTD    36  56    380/UNLTD  780    980  1800/UNLTD     -   -     -/-       2880 UNLTD
APG       56( 89%)     0/UNLTD    24  64      0/UNLTD  765    980     0/UNLTD     2   2     0/UNLTD   2880 UNLTD
SDF      305( 69%)     1/8         - 144     72/288      - 12288  2600/UNLTD     -   -     -/-       2880 UNLTD
===================================================================================================================
TOTAL:          JOBS) 20/UNLTD       CPUS) 452/500          MEM) 4400/18432        GPUS)  0/4

fe3-adm:
                                                                Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
14421.fe3-adm   appadm   APC      MS_FRJZR   15345*   1  30  120gb 2880: R 59:15 cl11-adm/0*30
14425.fe3-adm   appadm   APC      MS_FSI7Q   18073*   1  30  120gb 2880: R 59:13 cl12-adm/0*30
15084.fe3-adm   appadm   APC      MS_H5QR8   32758*   1  30  120gb 2880: R 46:13 cs10-adm/0*30
...
```

（＊）This is a command unique to our supercomputer system, not a genuine PBS command.

4.9 qstatmyjobs command (continued)

- Find the node with the most available memory (-m option)

```
$ qstatmyjobs -m
User: appadm
                          JOBS            CPUS                  MEM(gb)                    GPUS           WALLTIME(h)
Queue  avail(use%)  mysum/max   avail max  mysum/max   avail    max  mysum/max   avail max  mysum/max  default max
------ ------------ ----------- -------------------- ----------------------- ------------------ -------------
SMALL   1193( 76%)     0/UNLTD     12   12      0/96      48     48      0/UNLTD     -    -      -/-          6 12
APC     1025( 78%)    19/UNLTD     12   56   380/UNLTD   880    980   1800/UNLTD     -    -      -/-       2880 UNLTD
APG       56( 89%)     0/UNLTD     24   64     0/UNLTD   765    980      0/UNLTD     2    2    0/UNLTD     2880 UNLTD
SDF      305( 69%)       1/8        - 144     72/288       - 12288  2600/UNLTD     -    -      -/-       2880 UNLTD
=================================================================================================================
TOTAL:           JOBS)  20/UNLTD       CPUS)  452/500            MEM)  4400/18432        GPUS)   0/4

fe3-adm:
                                                                      Req'd  Req'd   Elap
Job ID           Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
---------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
14421.fe3-adm    appadm   APC      MS_FRJZR   15345*   1  30  120gb 2880: R 59:15 cl11-adm/0*30
14425.fe3-adm    appadm   APC      MS_FSI7Q   18073*   1  30  120gb 2880: R 59:13 cl12-adm/0*30
15084.fe3-adm    appadm   APC      MS_H5QR8   32758*   1  30  120gb 2880: R 46:13 cs10-adm/0*30
...
```

4.9 qstatmyjobs command (continued)

Item	Description
Queue	Queue name
vacant	Number of unused cores out of the total number of cores available to the queue
(use%)	Percentage of cores in use out of the total number of cores available to the queue
jobs sum	Total number of jobs you submitted
jobs sum_max	The maximum number of jobs you can run simultaneously
ncpus avail	The maximum number of cores on which a job start running immediately
ncpus max	The maximum number of cores for that queue
ncpus sum	Total number of cores of your job
ncpus sum_max	The maximum number of available cores per user
mem(gb) avail	The maximum memory size on which a job start running immediately
mem(gb) max	The maximum memory size for that queue
mem(gb) sum	Total memory size for your jobs
mem(gb) sum_max	The maximum total memory size per user
walltime(h) default	The default walltime (elapsed time) for that queue
walltime(h) max	The maximum walltime (elapsed time) for that queue

4.10 Checking the information on a finished job

$ tracejob [option] [Job ID]

- option
  -n (number of days) ･･･ how many days go back in the PBS log files

```
$ tracejob -n 2 191784.fe3-adm

Job: 191784.fe3-adm
...
04/19/2024 00:46:35  S    Exit_status=0 resources_used.cpupercent=149 resources_used.cput=00:08:18
                          resources_used.mem=39432kb resources_used.ncpus=4 resources_used.vmem=416888kb
                          resources_used.walltime=00:05:38
```

Exit_status	Exit status of the job (error if not 0)
resources_used.cpupercent	Average CPU usage
resources_used.mem	Amount of memory used
resources_used.ncpus	ncpus specified by qsub
resources_used.walltime	Job execution time

(Note) For an array job, use the PbsExitStatus command in the next page.

4.10 Checking the information on a finished job (continued)

- Checking the exit status of a (array) job.
  $ PbsExitStatus(*) [option] [Job ID]
    - -e : Show only the status that don't have Exit_status=0 (that is, the ones that have error)
    - -m : Show the largest memory used (resource_used.mem)
    - -c : Count the number of output
    - -t : Show the total of walltime

Example)

```
$ PbsExitStatus -e 213483[].fe3-adm
…
20240502:05/02/2024 13:22:44;0010;Server@fe3-adm;Job;213483[3583].fe3-adm;Exit_status=271
resources_used.cpupercent=99 resources_used.cput=00:17:02 resources_used.mem=33733660kb
resources_used.ncpus=2 resources_used.vmem=34207856kb resources_used.walltime=00:17:17
```

- In case Exit_status is not 0, check if resources_used.mem and/or resources_used.walltime exceeds the specified value (or the default value of the queue).

(*) This is a command unique to our supercomputer system, not a genuine PBS command..

4.11 Using SSD (SDF queue only)

- When a job is submitted by PBS, a temporary directory
  /scratch/pbs_tmpdir/(JobID)/
  is automatically created.
  This directory can be accessed via ${TMPDIR}/.

/scratch/ ⋯ SSD (sdf1, sdf2)
⋯ Lustre filesystem (PC cluster)

- ${TMPDIR}/ will be deleted automatically when the job finishes.

- When executing a command with high IO load, such as denovo assembling tools, in SDF queue, please specify ${TMPDIR} as the output/temporary directory of the command.

- Example 1) Specify ${TMPDIR}/ as a temporary directory for the command.
  - spades --tmp-dir=${TMPDIR} ⋯
  - megahit --tmp-dir ${TMPDIR} ⋯
- Example 2) Specify ${TMPDIR}/ as an output directory.
  - mkdir ${TMPDIR}/output/
    flye -o ${TMPDIR}/output/ ⋯
    mv ${TMPDIR}/output/ /aptmp/xxx/ # move the results to /aptmp/.

5. Appendix

5.1 Notes on Using the conda Command (1/2)

- It is totally fine to install whatever apps you want using the **Anaconda** or **Miniconda** commands. However, when you execute "**conda init**", the following configuration may be appended to your **~/.bashrc** file.

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/usr/appli/freeware/miniconda/3.6/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/usr/appli/freeware/miniconda/3.6/etc/profile.d/conda.sh" ]; then
        . "/usr/appli/freeware/miniconda/3.6/etc/profile.d/conda.sh"
    else
        export PATH="/usr/appli/freeware/miniconda/3.6/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

- This configuration can conflict with applications installed in the supercomputer environment and may cause unexpected errors.

- Therefore, please follow the procedure in the next section and write the conda configuration in a separate file (e.g., **~/.bashrc.conda**) instead of **~/.bashrc**.

5.1 Notes on Using the conda Command (2/2)

1. Checking the configuration in ~/.bashrc

   Cut out the section related to conda (the part between **# >>> conda initialize >>>** and **# <<< conda initialize <<<**) from **~/.bashrc**, paste it into another file such as **~/.bashrc.conda**, and save it.

2. Activating the conda environment using **~/.bashrc.conda**

   When you need to use conda commands, load the configuration by executing the following command:
   $ source **~/.bashrc.conda**