FY2022
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/

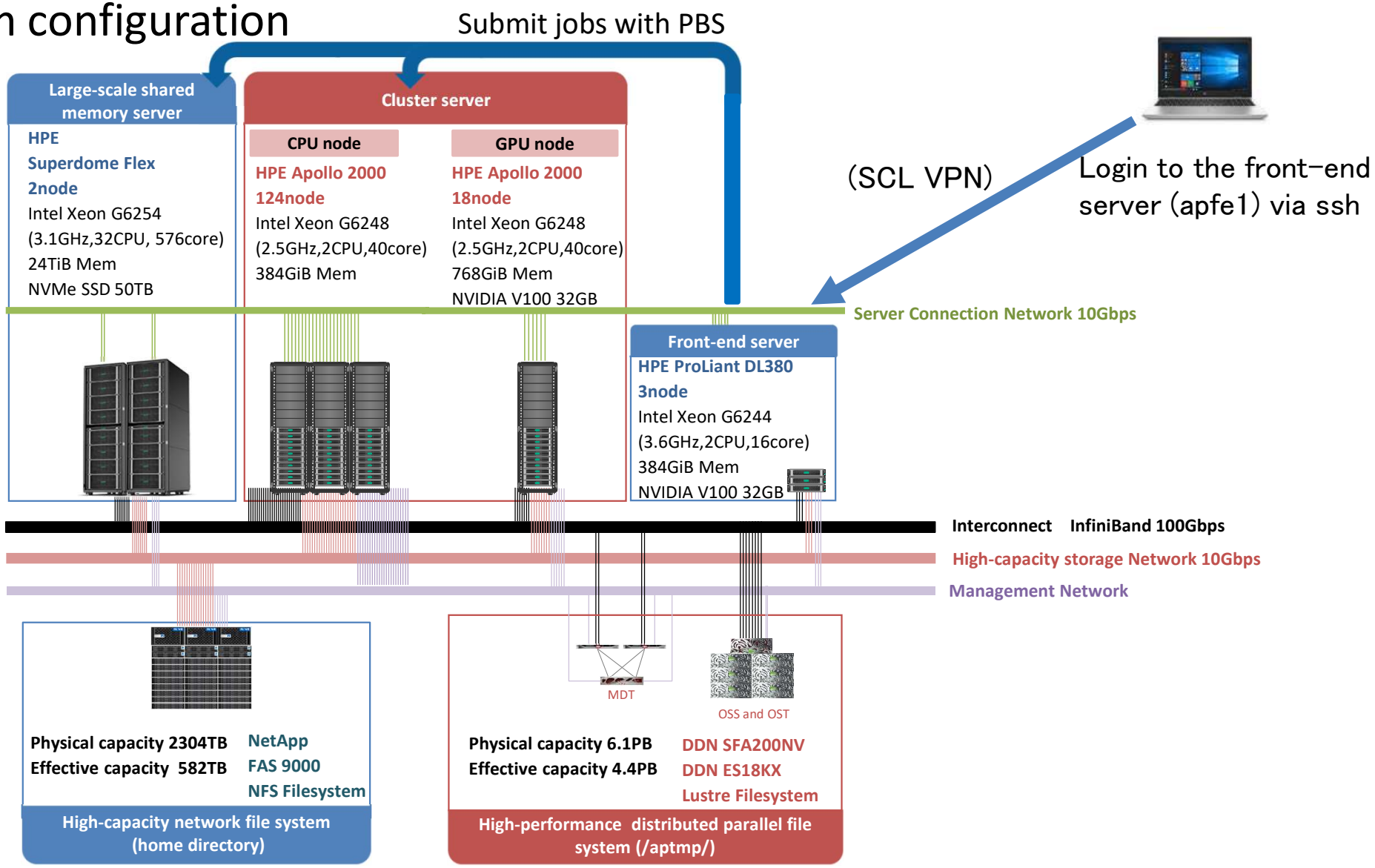
Co-organized by
Bioinformatics Center, Institute for Chemical Research, Kyoto University,
NPO Bioinformatics Japan

ver 1.0.1

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	HPE Apollo2000	HPE ProLiant DL380
# of nodes	2	124	18	3
hostname	sdf1, sdf2	ap001～ap124	apg01～apg18	apfe1
CPU	Intel Xeon G6254 3.1GHz x 32CPU (18cores/CPU)	Intel Xeon G6248 2.5GHz x 2CPU (20cores/CPU)		Intel Xeon G6244 3.6GHz x 2CPU (8cores/CPU)
# of cores/node	576	40	40	16
memory size /node	24TiB (42GiB/core)	384GiB (9GiB/core)	768GiB (18GiB/core)	384GiB
GPU	-	-	NVIDIA V100 (5,120cores、32GB)	NVIDIA V100 (5,120cores、32GB)
SSD	50TB	-	-	-
OS	Red Hat Enterprise Linux(＊) 7.6			

(＊) 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.

- Report of research results (users of compute server)
  - 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

Target	Calculation Rule
Basic fee	￥1,000/month
Compute node (CPU time)	￥0.00125/sec （There is a ceiling limit of ￥30,000/month for users in ICR, and ￥40,000/month for users outside ICR.）
Home directory	Free (upto 20GB available)
Cloud storage service	Free (upto 20GB available)
Optional Service	
Home directory expansion service	￥1,000/fiscal year for +100GB (upto 300GB can be added)
Cloud storage expansion service	￥1,000/fiscal year for +100GB (upto 300GB can be added)

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

2.3 How to Login to the Supercomputer System

- To use the supercomputer system, login to the front-end server apfe1.
  - (Linux, Mac) In a terminal

```
$ ssh username@apfe1.scl.kyoto-u.ac.jp
```

(× login.scl.kyoto-u.ac.jp)

(Windows) Install terminal software such as TeraTerm, Putty, RLogin, etc.

  - To use GUI applications (LibreOffice, gnuplot, FigTree, IGV, etc) on apfe1.
    - (Linux) Login via ssh –Y(C).
    - (Mac) Install Xquartz and login via ssh –Y(C).
    - (Windows) Install Xming.

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

```
$ ldapchsh bash        # changing to bash
```

2.3 How to Login to the Supercomputer System(Continued)

- To login to the front-end server 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
  (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)

- For jobs running on apfe1, there is a limit on memory size and cpu time (9GB, 30 minutes).
- Use PBS interactive batch job (§ 4.7) to login to the compute node and to work on it interactively.
  - As a general rule, do not login to the compute node directly by ssh.

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

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-2 (uid 10152):
     Filesystem  space  quota  limit  grace  files  quota  limit  grace
fas9000-02_NFS:/HOME/user2/
                948M   18432M  20480M          373    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)/)

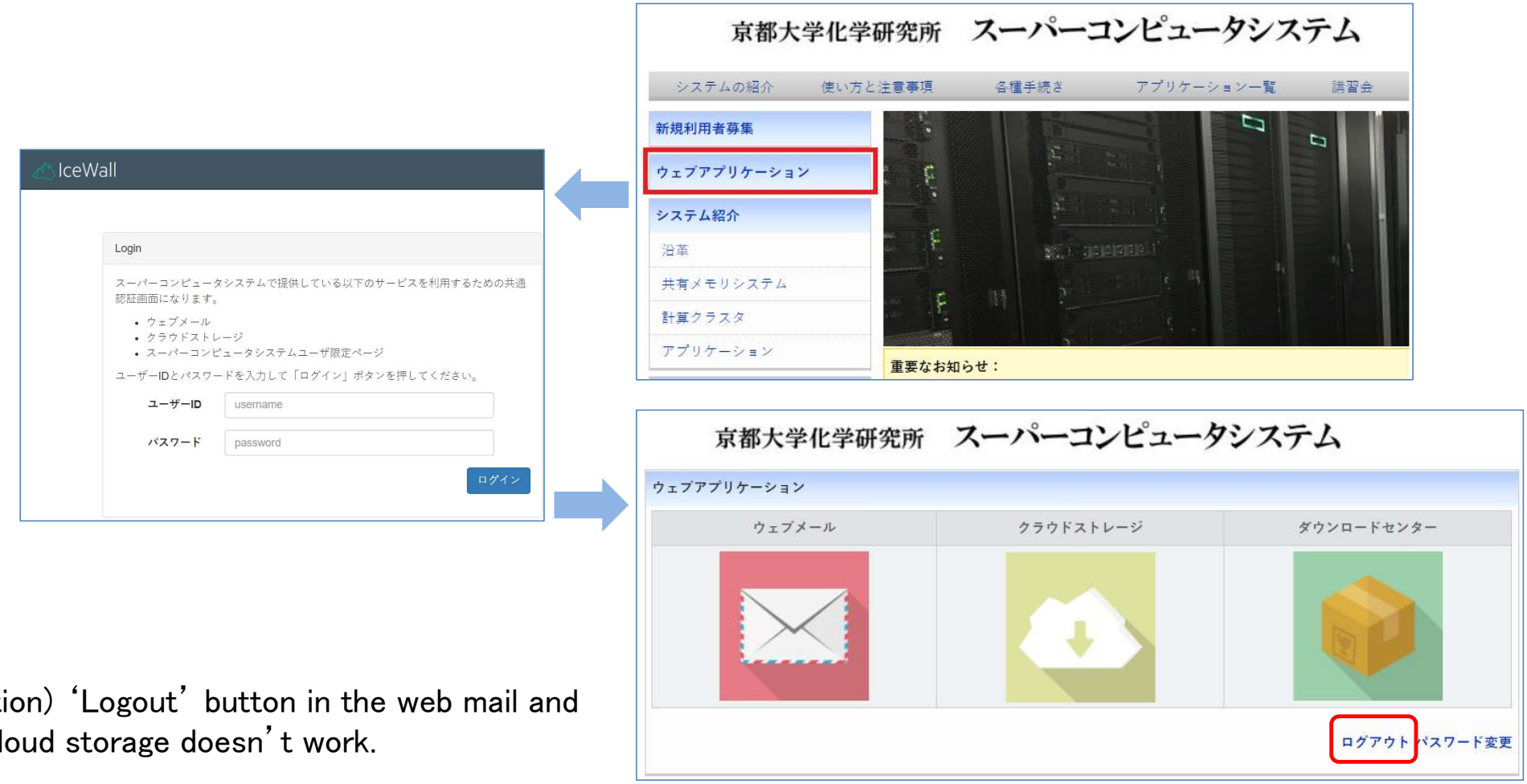
- Use this area for calculation.
- Not charged.
- No limitation on disk usage or storage period.
  - ➢ You may be asked to organize (compress or delete) your files/directories, if available disk space is getting low.
  - ➢ Pay attention not only to the file size, but also to the number of files (and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

```
$ 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
```

  - Number of files

```
$ find (directory)/ | wc -l                    # 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.
```

2.5 Web Applications

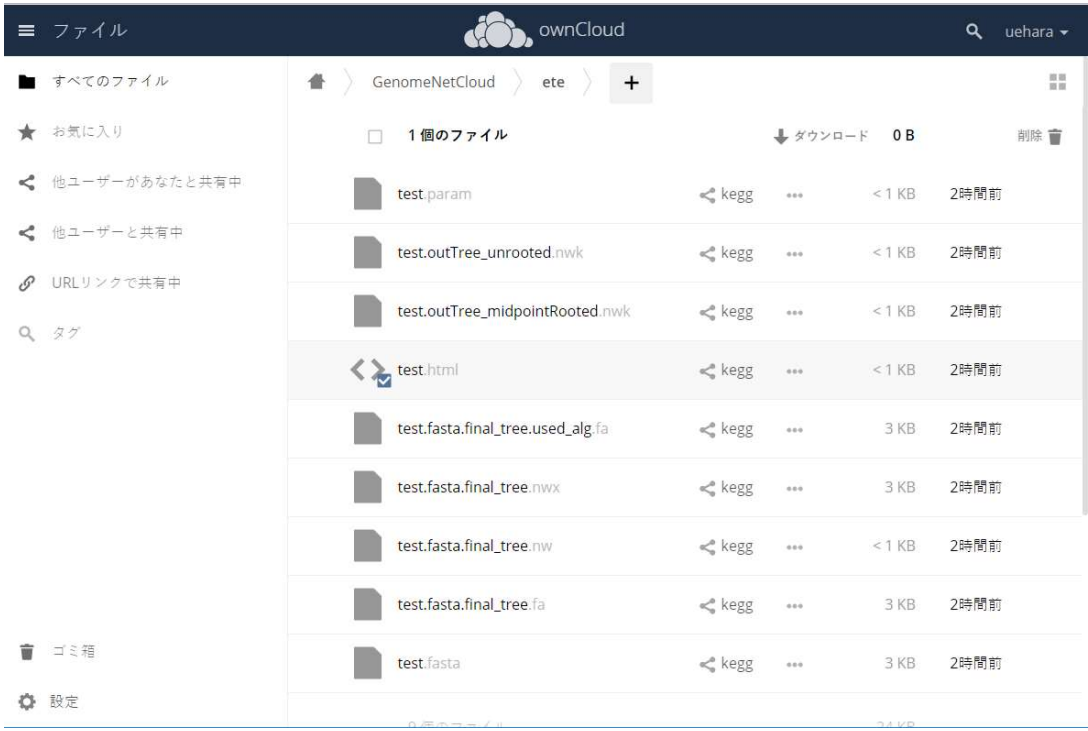


(Caution) 'Logout' button in the web mail and the cloud storage doesn't work.

2.6 ownCloud Calculation Service

- Execute calculations asynchronously using ownCloud.
  - Put input files in the specified ownCloud folder (“GenomeNetCloud”→”Tool name”), the calculation will be executed and results file are created.
  - Please contact us before using this service.

- The currently available services.
  - ETE (phylogenetic tree construction)



2.7 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）.

```
lect-1@apfe1 ~> qstatmyjobs
User: lect-1
                          jobs             ncpus                         mem(gb)                   walltime(h)
Queue   vacant(use%)  sum/sum_max     avail  max  sum/sum_max     avail   max  sum/sum_max     default max
-----   ------------  -----------     --------------------------  -----------------------------  ----------------
QUICK     1443( 78%)   0/2               4    4    0/unlimited       72     72   0/unlimited        1 01
SMALL      965( 82%)   0/unlimited      12   12    0/36             108    108   0/324              6 12
APC        857( 84%)   0/unlimited      35   40    0/unlimited       60    720   0/unlimited     2880 unlimited
APG         68( 89%)   0/4              16   40    0/unlimited        8    720   0/unlimited     2880 unlimited
SDF        398( 59%)   0/8               -  144    0/288              -   6144   0/8192          2880 unlimited
=========================================================================================================================
TOTAL                  0/unlimited                 0/500                         0/12288
```

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 (case sensitive)

$ 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.*
- Weekly updated.
- 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

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 apfe1 (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 ＊)	12TB (3TB)

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

4.2 Job Submission

$ qsub [options] (job script file)

- options
  - −q xxxx queue name (QUICK, 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 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	QUICK	SMALL	APC	APG	SDF
# of compute servers	Cluster (CPU): 124 Cluster (GPU):18 Large shared memory :2	Cluster (CPU): 124 Cluster (GPU):16	Cluster (CPU):121 Cluster (GPU):16	Cluster (GPU):16	Large shared memory :2
Job type	Test job	Small job	Normal job	GPU job	Large memory job
Queue priority	100	70	50	50	90
Limitations per job					
Max. # of cores (default)	4 (1)	12 (1)	40 (※1) (1)	40 (1)	144 (18)
Max. memory (default)	72 GB (9 GB)	108 GB (9 GB)	720 GB (※1,2) (9 GB)	720 GB (18 GB)	6 TB (768 GB)
Max. elapsed time (default)	1 h	12 h (6 h)	No limit (2880 h)	No limit (2880 h)	No limit (2880 h)
Limitations per user					
Max. # of jobs (soft limit)	2 (1)	-	-	4 (1)	8 (4)
Total # of cores (soft limit)	-	36	500	-	288 (144)
Total memory (soft limit)	-	324 GB	4.5 TB	-	8 TB (6 TB)

(※1) You can use 500 cores and 4.5TB memory in a MPI job using multiple nodes.

⚠(※2) Of all 137 nodes in the APC queue, only 16 nodes (GPU-equipped nodes) can use memory larger than 360GB.

4.3 Queue List (continued)

Queue name	cdb
# of compute servers	dl560: 1
Job type	GenomeNet Service Developler
Queue priority	50
Limitations per job	
Max. # of cores (default)	32 (1)
Max. memory (default)	1.3TB (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

- The home directory of dl560 is different from that of apfe1 (/aptmp/ is common).
- Jobs executed directly on dl560 without PBS (cdb queue) have a CPU time limit of 30 minutes.
- Of dl560's 64 cores and 1.5TB memory, cdb queue can be used up to 58 cores and 1.3TB 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=...)

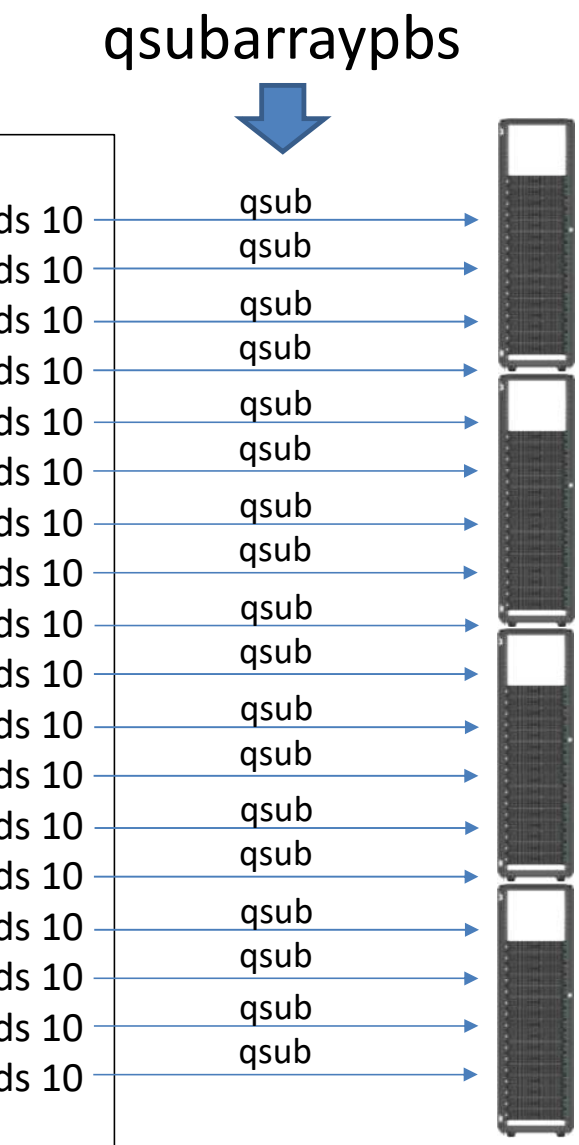
4.4 Job Script Example

```
#!/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+/2.14.0                  ← 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 not a genuine PBS command, but an original one unique to our supercomputer system.

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

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

- Many jobs written in a job file are executed simultaneously using PBS array job feature.
- Options (basically the same as the qsub command)
  - -q (queue name) … QUICK, SMALL, APC, APG, SDF (required)
  - -l select=1:ncpus=(number of cores used by a job per line):mem=(the size of memory used by a job per line)
  - Other qsub options (-N, -r, -v, -l, -p, -P, -W)
- A jobs file
  - Write one job per line (you can list multiple commands on a line by separating then with ; or &&).
  - If you want to redirect the output of the command to a file, write it in sh/bash format.
    (command) 1>xxx.out 2>xxx.err
  - Lines beginning with # are comment lines and ignored (#PBS doesn't work)
- Supplement
  - module load the application to be executed before submitting qsubarraypbs.
  - 'cd $PBS_O_WORKDIR/' is executed at the beginning of each sub-jobs submitted by qsubarraypbs.
  - Jobs for the maximum number of cores per user for that queue will be executed simultaneously.
  - Error message and failed sub-jobs are output to the pbslog/ directory.
  - The PBS mail notification function (-M, -m option) is not available.
  - (a jobs file) should not exceed 10,000 lines.(due to the limitation of PBS array job)

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-splilter --n-parts (number of splits) (FASTA file)
        # specify number of splits (--n-parts) or split size (--part-size)
```

- 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
…
```

4.6 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.
  - Use interactive batch jobs only in the SMALL queue

```
[appadm@apfe1]$ qsub -I -q SMALL -l select=1:ncpus=10:mem=30gb -l walltime=12:00:00
qsub: waiting for job 205274.apfe3 to start
qsub: job 205274.apfe3 ready

cd /scratch/pbs_jobdir/pbs.205274.apfe3.x8z
[appadm@ap118]$ cd /scratch/pbs_jobdir/pbs.205274.apfe3.x8z
[appadm@ap118]$
```

4.7 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

＊ 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

- Delete jobs
  $ qdel [Job ID] [Job ID …]

- 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

4.8 qstatmyjobs command（＊）

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

```
$ qstatmyjobs
User: user
                      jobs                 ncpus                          mem(gb)             walltime(h)
Queue  vacant(use%) sum/sum_max    avail max  sum/sum_max     avail   max  sum/sum_max    default max
-----  ------------ -------------- ---------------------------  ----------------------------- -----------------
QUICK    2798( 52%)   0/2               4   4    0/unlimited      72    72    0/unlimited        1 01
SMALL    2656( 50%)   0/unlimited      12  12    0/36            108   108    0/324              6 12
APC      2536( 51%)  10/unlimited      32  40  110/unlimited     360   720  640/unlimited     2880 unlimited
APG       356( 44%)   0/4              40  40    0/unlimited     720   720    0/unlimited     2880 unlimited
SDF        62( 87%)   1/8               - 144   20/288             -  6144 6000/8192          2880 unlimited
=================================================================================================================
TOTAL                11/unlimited              130/500                     6640/12288
```

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

4.8 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

4.9 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.apfe3

Job: 191784.apfe3
...
04/19/2020 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.

4.9 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[].apfe3
…
20200502:05/02/2020 13:22:44;0010;Server@apfe3;Job;213483[3583].apfe3;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 not a genuine PBS command, but an original one unique to our supercomputer system.

4.10 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 a output directory.
  - mkdir ${TMPDIR}/output/
    flye -o ${TMPDIR}/output/ …
    mv ${TMPDIR}/output/ /aptmp/xxx/ # move the results to /aptmp/.