

Table of Contents

- ▶ [Connection](#)
 - ▶ [Login to RCCS](#)
 - ▶ [Register SSH Public Key and Password for Web](#)
 - ▶ [Login Shell](#)
- ▶ [Whole System of RCCS](#)
- ▶ [RCCS Resources](#)
 - ▶ [CPU Points and Queue Factor](#)
 - ▶ [Checking Resources](#)
 - ▶ [Individual Limitation of Resources](#)
- ▶ [Queueing System](#)
 - ▶ [Overview of Queueing System](#)
 - ▶ [Queue Classes](#)
 - ▶ [Show Job Status](#)
 - ▶ [Submit Your Jobs](#)
 - ▶ [Delete Jobs](#)
 - ▶ [Hold/Release Jobs](#)
 - ▶ [Get Information of Finished Jobs](#)
- ▶ [Build and Run](#)
 - ▶ [Command to Build](#)
 - ▶ [Available MPI](#)
 - ▶ [Available Math Libraries](#)
 - ▶ [Running Parallel Program](#)
 - ▶ [Development Tools](#)
 - ▶ [Environment Modules](#)
- ▶ [Package Programs](#)
- ▶ [Special Commands of RCCS](#)
 - ▶ [Queueing System Related](#)
 - ▶ [Show Resources Used](#)
 - ▶ [Utility Commands for Batch Jobs](#)
 - ▶ [Manipulation of Files on Computation Nodes](#)
- ▶ [Inquiry](#)

Connection

Login to RCCS

- ▶ Frontend nodes (ccfep.ims.ac.jp) can be accessed using ssh with public key authentication.
- ▶ GPU equipped frontend nodes ('ccgpu', 'ccgpuv') can be accessed from 'ccfep'.
- ▶ All computers will stop during 9:00-19:00 on the first Monday of each month because of maintenance. The maintenance time might be extended.
- ▶ Access to frontend nodes is allowed only from IPv4 address assigned to Japan or other registered IP addresses. See [Application for SSH connection from outside Japan](#) for details.

Register SSH Public Key and Password for Web

Please prepare a public/private key pair of ssh. If you do not know the procedure, please search in internet by yourself.

First registration / Missing your username or password for web

1. Open <https://ccportal.ims.ac.jp/en/user/password> to request mail for registration in web browser.
2. Fill your email which is written in your application, then press button "E-mail new password".
3. After you receive an email from RCCS, open URL in the mail to login in web browser.
4. Fill your new password in "Password" and "Confirm password".
5. Paste your public key in "Public key".
6. Press "Save" button.

Using your username and password for web

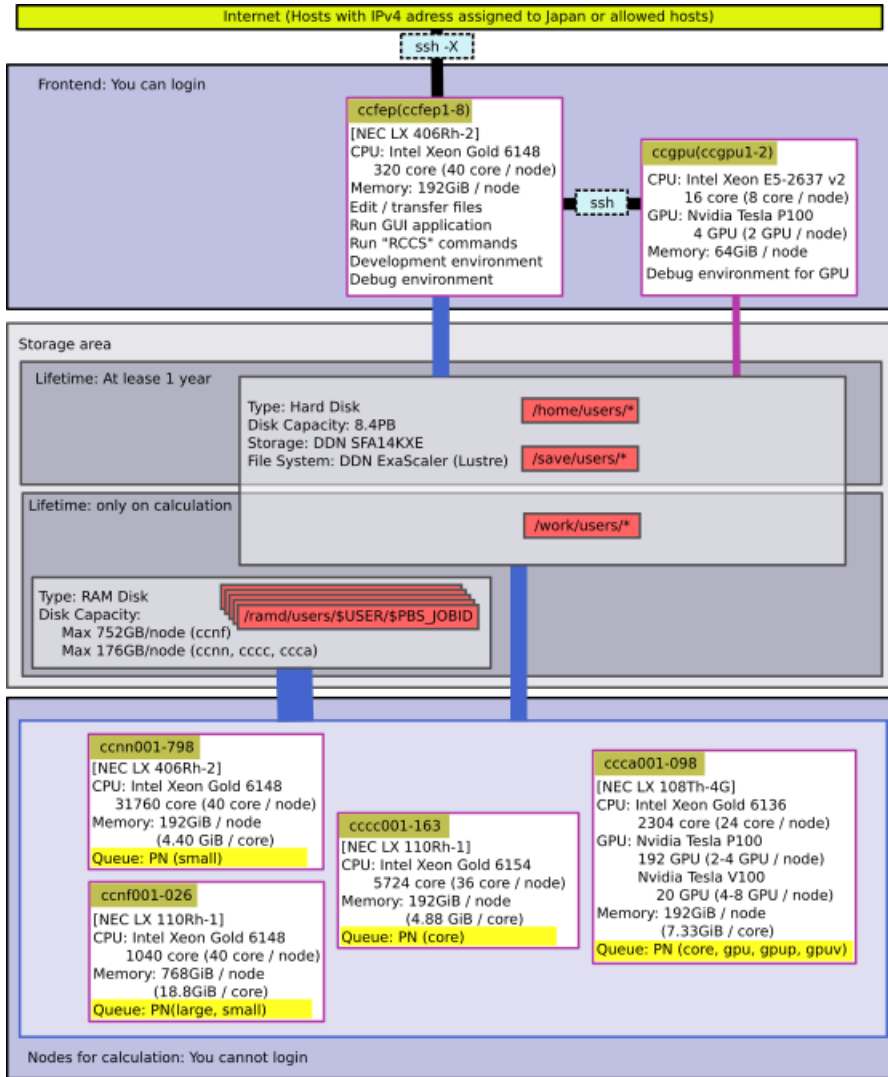
1. Open <https://ccportal.ims.ac.jp/en/frontpage> in web browser, then fill your username and password and press "Log in" button.
2. Press "My account" which is located in top right corner.
3. Press "Edit" tab.
4. To change password, fill current password and new passwords.
5. Paste your public key.
6. Press "Save" button.

Login Shell

- ▶ /bin/csh(tcsh), /bin/bash and /bin/zsh are available.
- ▶ You can select login shell in the same page as ssh public key. It will take some time to change login shell.
- ▶ You can customize your .login or .cshrc in your home directory, but be carefully.

Whole System of RCCS

- ▶ Whole system of RCCS is shown in the bottom figure.
- ▶ Interactive computers are ccfep (8 nodes) and ccgpu (2 nodes). You can build or debug applications.
- ▶ You can login any interactive computers from internet except ccgpu.
- ▶ There are four kinds of disks. Access speed or data lifetime are different. We call them /work, /ramd, /home and /save.
- ▶ Width of lines between disks and computers describes transfer speed. Wide line is faster.
- ▶ Disk /work is suitable to write temporary files in your calculation. But all files will be DELETED after your calculation finish.
- ▶ Disk /ramd is RAM disk whose size is about 176 GB or 752 GB. The sum of memory used in the job and in the RAM disk is under control of the queuing system.
- ▶ The difference between /home and /save is that /home has a copy on other disks.
- ▶ Use of /tmp, /var/tmp or /dev/shm is not allowed. Jobs using those directories will be killed by an administrator.



RCCS Resources

CPU Points and Queue Factor

CPU points are spent when you use CPU or GPU.

Queue factors are determined as follows on each systems.

System	CPU Queue Factor	GPU Queue Factor
cclx (jobtype=large)	42 / (point/(1 node * 1 hour))	-
cclx (jobtype=small)	28 / (point/(1 node * 1 hour))	-
cclx (jobtype=core)	1.0 / (point/(1 core * 1 hour))	-
cclx (jobtype=gpu, gpup)	1.0 / (point/(1 core * 1 hour))	10 / (point(1 GPU * 1 hour))
cclx (jobtype=gpuv)	1.0 / (point/(1 core * 1 hour))	15 / (point(1 GPU * 1 hour))

- ▶ On ccfep, CPU points will be consumed according to CPU usage.
- ▶ On ccgpu1 and ccgpu2, CPU points won't be consumed.
- ▶ The time will be calculated using elaps of processes.
- ▶ We don't charge money for the supercomputer usage.

If you want to know your current CPU points, please try "showlim -c".

Checking Resources

- ▶ Summation of points about jobs on queuing systems and summation of occupied amount of disk is executed within 10 minutes.
- ▶ Summation of points about interactive execution is executed at 5:15.
- ▶ If you use all assigned points, all executing jobs of your group members are killed and new submissions are prevented.
- ▶ If you spend assigned amount of disk, new submissions are prevented.

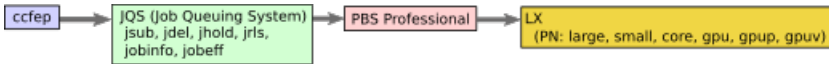
Individual Limitation of Resources

Access to "[Limiting Resources Page](#)" with your web browser.

- ▶ Only representative user can limit maximum resources of each members.
- ▶ Normal user can only view the values of maximum resources.
- ▶ Maximum available number of cpus, point and amount of disks can be limited.

Queueing System

Overview of Queueing System



Queue Classes

Queue class for all users

System	Class	Node	Memory	Limitation for a job	# of cores per group		# of jobs per group	
					Assigned points	# of cores/gpus	Assigned points	# of jobs
cclx	PN (jobtype=large)	ccnf	18.8GB/core	1-10 nodes (40-400 cores)	3,000,000 - 1,000,000 - 300,000 - 100,000 - - 100,000	4000/48 2560/32 1600/20 960/12 320/8	3,000,000 - 1,000,000 - 300,000 - 100,000 - - 100,000	4000 2560 1600 960 320
cclx	PN (jobtype=small)	ccnn ccnf	4.4GB/core	1-32 nodes (40-1280 cores)				
cclx	PN (jobtype=core)	cccc ccca	4.8GB/core	1-36 cores				
cclx	PN (jobtype=gpu, gpup)	ccca	7.3GB/core	1-48 gpus 2-24 cores/node (2 gpus/node) 1-12 cores/node (1 gpu/node)				
cclx	PN (jobtype=gpuv)	ccca	7.3GB/core	1-8 gpus 1-3 cores/gpu (multinode jobs not allowed)				

- ▶ Max elaps time for jobtype=core and ncpus>18 jobs must be <= 1 week (168 hours).
- ▶ Max elaps time for jobtype=gpuv jobs must be <= 1 week (168 hours).
- ▶ Max elaps time for the other jobs is until next maintenance. Only half nodes can be assigned for over 1 week jobs.
- ▶ A job which use <= 526 nodes uses nodes assigned in the same Omni-Path group.
- ▶ 272 nodes of Node "ccnn" are used only for 1-4 node jobs.
- ▶ Jobtype "small" jobs whose walltime is less than 1 day might use Node "ccnf".
- ▶ Jobtype "core" jobs whose walltime is less than 3 days and requested cores is from 6 to 12 might use Node "ccca".
- ▶ GPU Direct Peer-to-peer communication is available in any ccca node (jobtype = gpup, gpuv, or gpu).
 - ▶ jobtype=gpup jobs shall run on NVIDIA Tesla P100 equipped nodes.
 - ▶ jobtype=gpuv jobs shall run on NVIDIA Tesla V100 equipped node.
 - ▶ jobtype=gpu jobs will run on either of P100 or V100 equipped nodes.

Special queue class

The settings of queue class are following.

System	Class	Wall Time	Memory	# of cores per job	# of cores per group
cclx	(occupy)	7 days	4.4GB/core	ask us	allowed number of cores

Show Job Status

To show the summary of all jobs on the system, you should type:

```
ccfep% jobinfo [-s] -h cclx
```

To show the summary of all jobs on the queue class, you should type:

```
ccfep% jobinfo [-s] -q (PN|PNR[0-9])
```

Option "-s" can be omitted.

To show the detail of some/all jobs of the system, you should type:

```
ccfep% jobinfo -l [-g|-a] -h cclx
```

Jobs belong to the same group are shown with option "-g". Jobs of all users are shown with option "-a". Information not related to you is encrypted.

To show the detail of some/all jobs of the queue class, you should type:

```
ccfep% jobinfo -l [-g|-a] -q (PN|PNR[0-9])
```

To show the working directory of your jobs, you should type:

```
ccfep% jobinfo -w -q (PN|PNR[0-9])
```

Submit Your Jobs

Description of the header part

You have to write a script file which is written in C-shell to submit your job. An example for each system is following.

- ▶ csh, bash (/bin/sh), zsh can be used for the job submission script.
- ▶ lines started with #PBS are common, regardless of the shell type.
- ▶ Sample scripts can be found in `ccfep/local/apl/lx/(application name)/samples/`.

Meaning	Header part	Importance
The First Line	(csh) #!/bin/csh -f (bash) #!/bin/sh (zsh) #!/bin/zsh	Required (choose one)
Needed Number of CPU	#PBS -l select=[<i>Nnode</i>]:ncpus= <i>Ncore</i> :mpiprocs= <i>Nproc</i> :ompthreads= <i>Nthread</i> :jobtype= <i>Jobtype</i> :ngpus= <i>Ngpu</i>]	Required
Wall Time	#PBS -l walltime=72:00:00	Required
Mail at Start and End	#PBS -m be	Optional
Prevent to Rerun	#PBS -r n	Optional
Change to Submitted Directory	cd \${PBS_O_WORKDIR}	Recommended

- ▶ Nnode: # of physical node
- ▶ Ncore: # of reserved cores per physical node
- ▶ Nproc: # of processes per node
- ▶ Nthread: # of threads per process
- ▶ Jobtype: large, small, core, gpu, gpup, gpuv
 - ▶ large: 18.8GB / core
 - ▶ small: 4.4GB / core
 - ▶ core: job for less than 18 cores
 - ▶ gpu, gpup, gpuv: GPU jobs
- ▶ Ngpu: # of GPUs

Example of "Needed Number of CPU": Case of 80 mpi processes on 2 nodes

```
#PBS -l select=2:ncpus=40:mpiprocs=40:ompthreads=1:jobtype=small
```

Example of "Needed Number of CPU": Case of GPGPU

```
#PBS -l select=1:ncpus=6:mpiprocs=1:ompthreads=1:jobtype=gpu:ngpus=1
```

You can find some other examples in [this page](#).

Job submission

After you write the script, type following command to submit.

```
ccfep% jsub -q (PN|PNR[0-9]) [-g XXX] [-W depend=(afterok|afterany):JOBID[:JOBID2...]] script.csh
```

If you want to submit your jobs by ' Supercomputing Consortium for Computational Materials Science' group, use -g option. (*XXX* is name of its group)

You can describe dependency of jobs using -W option.

If you want to describe dependency that a job should run after the dependent job exit successfully, use keyword "afterok".

If a job should run after the dependent job including abnormal exit, use keyword "afterany".

Sample script files exist in `ccfep:/local/apl/lx/*samples/`.

Delete Jobs

You can look for Request ID which you want to delete by jobinfo command. Type the following command.

```
ccfep% jdel [-h cclx] RequestID
```

Hold/Release Jobs

You can hold jobs. Type the following command.

```
ccfep% jhold [-h cclx] RequestID
```

You can release jobs. Type the following command.

```
ccfep% jhold [-h cclx] RequestID
```

Get Information of Finished Jobs

You can get information of finished jobs, such as finish time, elaps time, parallel efficiency, by *jobeff* command.

```
ccfep% jobeff -h (cclx|cckf) [-d "last_n_day"] [-a] [-o item1[,item2[,...]]]
```

You can customize items which are displayed in -o option. Available keywords are,

- ▶ queue: Queue name
- ▶ jobid: Job ID
- ▶ user: User name
- ▶ group: Group name
- ▶ node: Top node name
- ▶ Node: All node names
- ▶ start: Start time (YYYY/MM/DD HH:MM)
- ▶ Start: Start time (YYYY/MM/DD HH:MM:SS)
- ▶ finish: Finish time (YYYY/MM/DD HH:MM)
- ▶ Finish: Finish time (YYYY/MM/DD HH:MM:SS)
- ▶ elaps: Elaps
- ▶ cputime: Total CPU time
- ▶ used_memory: Used memory size

- ▶ ncpu: Number of reserved cpus
- ▶ ngpu: Number of reserved gpus
- ▶ nproc: Number of process for mpi
- ▶ nsmp: Number of threads per one process
- ▶ peff: Efficiency of job
- ▶ attention: Bad efficiency job
- ▶ command: Job name
- ▶ point: CPU points

Build and Run

Command to Build

System	Language	Non-Parallel	Auto-Parallel	OpenMP	MPI
cclx (Intel)	Fortran	ifort	ifort -parallel	ifort -qopenmp	mpiifort
	C	icc	icc -parallel	icc -qopenmp	mpiicc
	C++	icpc	icpc -parallel	icpc -qopenmp	mpiicpc
cclx (PGI)	Fortran	pgfortran	pgfortran -Mconcur	pgfortran -mp	
	C	pgcc	pgcc -Mconcur	pgcc -mp	
	C++	pgcpp	pgcpp -Mconcur	pgcpp -mp	

Available MPI

System	Kind
cclx	Intel MPI (MPI 3.0 standard)

OpenMPI is also installed (once you load settings).

Available Math Libraries

System	Math Library
cclx	intel MKL, Intel IPP, Intel TBB

Running Parallel Program

System	Auto Parallel · OpenMP	MPI	Hybrid
cclx	setenv OMP_NUM_THREADS 4 ./a.out	mpirun -np 4 ./a.out	setenv OMP_NUM_THREADS 4 mpirun -np 8 ./a.out

"Hybrid" means combination of auto-parallel/OpenMP and MPI.

Development Tools

Some tools can be used in command line, but it is better to use X Window edition's.

Intel Inspector XE

- ▶ Memory / Thread inspector
- ▶ (GUI command) `inspxe-gui`
- ▶ (CUI command) `inspxe-cl`

Intel Vtune Amplifier XE

- ▶ Hotspot analyzer
- ▶ (GUI command) `amplxe-gui`
- ▶ (CUI command) `amplxe-cl`

Allinea Forge

- ▶ Debugger
- ▶ (GUI command) `ddt`

Environment Modules

Environment Modules ("module" command) is available from July 2018. See [this page](#) for detailed information.

Package Programs

- ▶ The installed package programs for each systems are listed in https://ccportal.ims.ac.jp/en/installed_applications.
- ▶ Sample script files are located in `ccfep:/local/apl/ix/appname/samples/`.
- ▶ Installed directories are located in `/local/apl/ix/appname/` on each systems.
- ▶ Applications compiled by center are listed in https://ccportal.ims.ac.jp/en/how_to_configure with detail description.

Request installation you want to use

Please fill the following items and send it [ccadm\[at\]draco.ims.ac.jp](mailto:ccadm[at]draco.ims.ac.jp).

- ▶ Software name and version that you want to use
- ▶ Overview of the software and its feature
- ▶ Necessity of installation to supercomputers in RCCS
- ▶ URL of the software development

Special Commands of RCCS

Related Queuing System

Showing Job Status

```
ccfep% jobinfo [-c] [-s] [-l|-m|-w [-g|-a]] [-n] -h cclx
```

OR

```
ccfep% jobinfo [-c] [-s] [-l|-m|-w [-g|-a]] [-n] -q (PN|PNR[0-9])
```

- ▶ -c Use current information instead of cached one.
- ▶ -s Show summary.
- ▶ -m Show memory information.
- ▶ -w Show working directory where job was submitted.
- ▶ -l Show list.
- ▶ -g Show all users of the same group.
- ▶ -a Show all users.
- ▶ -n Show node information.
- ▶ -h Specify host.
- ▶ -q Specify queue class.

Submitting Jobs

```
ccfep% jsub -q (PN|PNR[0-9]) [-g XXX] [-W depend=(afterok|afterany):JOBID1[:JOBID2...]] script.csh
```

Deleting Jobs

```
ccfep% jdel [-h cclx] RequestID
```

Holding Jobs

```
ccfep% jhold [-h cclx] RequestID
```

Releasing Jobs

```
ccfep% jrls [-h cclx] RequestID
```

Getting Information of Finished Jobs

```
ccfep% jobeff -h (cclx|cckf) [-d "last_n_day"] [-a [-o item1[,item2,...]]]
```

Submitting Gaussian Jobs

Case of Gaussian 16

```
ccfep% g16sub [-q "QUE_NAME"] [-j "jobtype"] [-g "XXX"] [-walltime "hh:mm:ss"] [-noedit] \  
[-rev "g16xxx"] [-np "ncpus"] [-ngpus "n"] [-mem "size"] [-save] [-mail] input_files
```

- ▶ Command "g09sub" are also available to use Gaussian 09.
- ▶ Default walltime is set to 72 hours. Please set excepted time for calculation and extra to do long job or run early.
- ▶ If you want to know the meaning of options and examples, please just type "g16sub".

Showing Used Resources

```
ccfep% showlim (-cpu|-c|-disk|-d) [-m]
```

- ▶ -cpu|-c: Show used point and limited value.
- ▶ -disk|-d: Show current disk size and limited value.
- ▶ -m: Show values of each members.

Utility Commands for Batch Jobs

Limit the walltime of command

```
/local/apl/lx/ps_walltime -d duration -- command [arguments...]
```

- ▶ -d duration: Duration to execute command is described as "-d 72:00:00".
- ▶ Command will be killed after specified duration.

Showing statistic of current job

```
/local/apl/lx/jobstatistic
```

- ▶ Show similar statistic information in the mail notification of end of your job when you describe in PBS header line.
- ▶ This statistic information is the information when the command is executed.

Output Items

- ▶ resources_used.cpubercent: Efficiency of CPU usage. Maximum value is multiplication the number of threads and 100. Illegal value will be shown when multi nodes are used.
- ▶ resources_used.cput: Sum of actual time for calculation in each CPUs.
- ▶ resources_used.mem: Amount of actual memory size.

- resources_used.ncpus: Number of actually used CPUs.
- resources_used.walltime: Real time for calculation.

Manipulation of Files on Computation Nodes

You can access local files on computation nodes which cannot be directly accessed from the frontend nodes (ccfep) via "remsh" command.

```
remsh hostname command options
```

- hostname: Hostname such as cccc???, ccca???, ccnn???, or ccnf???
- command: Command to be executed on the node. Acceptable commands are: ls, cat, cp, and find.
- options: Commandline arguments to the command.

Example: how to access ramdisk of a computation node, ccnnXXX, by user "zzz"

```
remsh ccnnXXX ls /ramd/users/zzz
```

```
remsh ccnnXXX cat /ramd/users/zzz/99999/fort.10 | tail
```

Host names and jobids of your jobs can be found in the output of "jobinfo" command. (see above for the usage)

Inquiry

Question about password

If you forget your password, please fill the following items and send to [ccadm\[at\]draco.ims.ac.jp](mailto:ccadm[at]draco.ims.ac.jp).

- ▶ Your Name
- ▶ Your User Code
- ▶ Your Group Code
- ▶ Your Affiliation
- ▶ Representative Name
- ▶ Number of Reception

Other questions

You can ask other question on homepage (<https://ccportal.ims.ac.jp/forum/>).

If your question is about your job problem, please write following items.

- ▶ Machine name or queue name
- ▶ Job ID
- ▶ Output of error
- ▶ Directory where you submit your job
- ▶ File name which you submit

If you can't, send your question to [ccadm\[at\]draco.ims.ac.jp](mailto:ccadm[at]draco.ims.ac.jp).
