Institute for Materials Research, Tohoku University
Accelerator Server Manual

May 27th, 2021

Center for Computational Materials Science,
Institute for Materials Research,
Tohoku University

Contents

1 Outline of accelerator server 1-4

1.1 Configuration and specification 1-5

1.2 Configuration of nodes 1-6

2 Login method 2-7

2.1 Register SSH public key 2-8

2.2 Login method to front end node 2-8

2.3 How to change password 2-8

3 Outline of storage 3-9

3.1 Configuration of storage 3-10

4 Job submit/management commands 4-12

4.1 Job submit command 4-13

4.1.1 Job submit command (qsub command) 4-13

4.1.2 Format of an execution script 4-15

4.1.3 Interactive mode 4-17

4.1.4 Submitting jobs to Shared-Queue CA_001 and CA_001g 4-18

4.2 Job management command 4-19

4.2.1 Display your own job information 4-19

4.2.2 Display job information 4-19

4.2.3 Display queue status 4-22

4.2.4 Display server status 4-23

4.2.5 Cancel the job before the job finished 4-24

4.3 Display information about used and remained time of job execution (jobtime command) 4-25

4.4 References for submitting job and script 4-26

4.4.1 How to execute a MPI job 4-26

4.5 List of queues 4-27

5 Compilers and Libraries 5-29

5.1 Compiler 5-30

5.1.1 Intel Compiler 5-30

5.1.2 PGI Compiler 5-33

5.1.3 GNU Compiler 5-35

5.1.4 nvcc Compiler 5-37

5.2 Libraries 5-39

5.2.1 Intel MKL 5-39
5.2.2 cuBLAS 5-40
5.2.3 cuDNN 5-40

6 Usage of applications 6-42
6.1 Applications 6-44
6.2 VASP 6-46
6.3 QUANTUM ESPRESSO 6-51
6.4 LAMMPS 6-54
6.5 Gaussian16 6-57
6.6 CRYSTAL 6-60
6.7 WIEN2k 6-61
6.8 SIESTA 6-63
6.9 ABINIT 6-64
6.10 CPMD 6-65
6.11 MaterialsStudio 6-66
6.11.1 How to set License server 6-67
6.11.2 How to set Gateway 6-68
6.11.3 How to execute 6-70
6.11.4 How to execute CASTEP 6-70
6.11.5 How to execute DMol3 6-73
6.11.6 Confirming Job Status 6-75
6.11.7 Canceling Job 6-75
6.12 Wannier90 6-77

7 Usage of Machine Learning Environment 7-78
7.1 Machine Learning Environment 7-79
7.2 Chainer 7-80
7.3 Keras 7-81
7.4 Caffe 7-82
7.5 Jupyter Notebook 7-83
7.6 DIGITS 7-85

8 Usage of Python 8-86
8.1 About using Python 8-87
8.2 Install pyenv 8-87
8.3 Setting environment variables 8-87
8.4 Operational check 8-87
8.5 Basic usage 8-87

8.6 Execution method 8-88

1

1 Outline of accelerator server

1.1 Configuration and specification

1.2 Configuration of nodes

1.1 Configuration and specification

The specification of the accelerator server.

Server name	Accelerator server	Front end node	Parallel computing
Model name	Cray CS-Storm 500GT	Cray CS500	HPE ProLiant DL360 Gen10
Number of servers	29 servers	2 servers	17 servers
CPU	Intel Xeon Gold 6150 ・Frequency : 2.7 GHz ・Number of CPU cores : 18 Core ・Number of socket : 2 Sockets/Server	Intel Xeon E5-2695v4 ・Frequency : 2.1GHz ・Number of CPU cores : 18 Core ・Number of socket : 2 Sockets/Server	Intel Xeon Gold 6154 ・Frequency : 3.0 GHz ・Number of CPU cores : 18 Core ・Number of socket : 2 Sockets/Server
Accelerator	NVIDIA Tesla V100 for PCIe ・Computational performance : 7.0 TFlops ・Number of GPU cores : 5,120 Core ・Number of socket : 10 Sockets/Server	-	-
Total amount of main memories	768 GiB/Server	128 GIB/Server	576 GiB/Server

1.2 Configuration of nodes

Nodes of the accelerator server are as follows.

Node type	Usage	Number of nodes	Installation site
Front end node	For submitting jobs	2 nodes	Center for computational materials science Room101
Computational node (queue：CA_001, A_004)	For calculating	23 nodes	Center for computational materials science Room101
Computational node (queue：IA_001g, CA_001g, DA_002g)	For calculating	6 nodes	Second building Room 713
Computational node (queue：IC_001, DC_002, C_002, C_004)	For calculating	17 nodes	Center for computational materials science Room101

2

2 Login method

2.1 Register SSH public key

2.2 Login method to front end node

2.3 How to change password

2.1 Register SSH public key

If you are connected to the supercomputer for the first time, generate a pair of public and private keys using Public key registration system.

2.2 Login method to front end node

Log in the ssh relay server ‘cms-ssh.sc.imr.tohoku.ac.jp’.

```
$ ssh -l username cms-ssh.sc.imr.tohoku.ac.jp
```

Log in the gpu that is front end node for the accelerator server from the ssh relay server.

```
$ ssh gpu
```

Please see web manual for more details.

2.3 How to change password

Use passwd command to change your login-password.
Please change your password, according to the following policies.

(1) Minimum password length is 10.
(2) Use one or more lower-case letters.
(3) Use one or more upper-case letters.
(4) Use one or more numerical digits.
(5) Use one or more special characters (Non-alphanumeric), such as !, #, $.

```
$ passwd
Current Password:[ current password]
New password:[ new password]
Retype new password:[ new password]
```

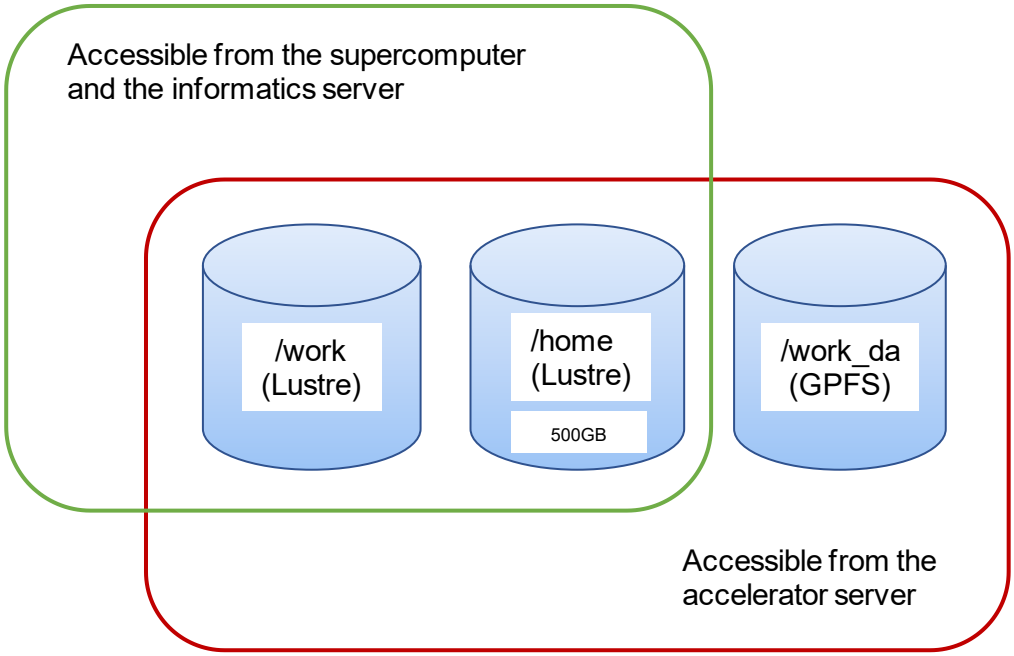
3

3 Outline of storage

3.1 Configuration of storage

3.1 Configuration of storage

The configuration of the storages in the supercomputing system is as follows.



List of storage area

Accessible machine	Area	Quota	Description
①Supercomputer ②Accelerator server ③Informatics server	home/UID	500GB	This area is user's home directory and stores data of supercomputing system.
①Supercomputer ②Accelerator server ③Informatics server	work/xxx	none	This is Lustre area. Please use the scratch if the sum of the output file exceeds 500GB. Move the necessary data to user's home directory and remove the unnecessary data from this area.

	work/scratch/xxx	none	This is Lustre area aimed for outputting a temporary file, such as a Gaussian. Be careful that if your file is not accessed more than one month, the file is automatically deleted.
Accelerator server	work_da	none	This is GPFS area. To use the queue IA_001g,DA_002g, please submit a job from this area. Please use this area for debugging before submitting a job to Luster area from the accelerator server. Move the necessary data to user's home directory and remove the unnecessary data from this area.

(*)UID：user account

xxx：Any directory or file that the user has created.

(*)The files not accessed more than one month are automatically deleted in scratch area of /work.

4

4 Job submit/management commands

4.1 Job submit command

4.2 Job management command

4.3 Display information about used and remained time of job execution (jobtime command)

4.4 References for submitting job and script

4.5 List of queues

4.1 Job submit command

4.1.1 Job submit command (qsub command)

Submit command places a job on the queue of the accelerator server.

Submitting options also can be described in the execution script.

For more details, see each manual.

* Please submit job to appropriate queue with qsub command, not to execute your program on the front end node directly. If you execute a program on the front end node, it may be canceled by the administrator because it has effects on other users.

（1）Usage

```
$ qsub [-q queue name] [-l select= number of nodes] [-N Job name] [-M e-mail address] [-m mail point] [-l walltime= limit of walltime] [execution script file]
```

（2）List of options

Option	Value
-q queue name	Set the queue name. As for details of queue, see a list of queues.
-l select= number of nodes	Specify number of the nodes to be used. No this option means default (see 4.5 List of queues)
-N job name	Set the job name. The name of job is up to 236 characters. The real-time job reference system displays up to 64 characters. If no '-N' option is specified, the system assign the default job name.
-M e-mail address	Set e-mail address to be received. To receive an e-mail, '-m' option is required.
-m mail point	Specify the set of events that causes mailing to be sent to the list of users specified in the '-M' option. To receive an e-mail, '-m' option is required.
-l walltime= walltime	Specify the limit of walltime If no this option is specified, the walltime is default value specified in the queue. (see 4.5 List of queues) Appropriate value makes a queued job running frequently.
-l license name= number of use licenses	Specify the number of licenses when you use applications that requires managed licenses. If no this options specified, the job is regarded not to use applications

	that licenses are managed As for specifying licenses, see how to execute applications.

(3) Example

Execute following jobs

・queue: A_004, number of nodes: 2, limit of walltime: 1 hour, script file: hello.sh

```
$ qsub -q A_004 -l select=2 -l walltime=1:00:00 hello.sh
```

Description in the execution script.

```
#!/bin/sh
#PBS -q A_004
#PBS -l select=2
#PBS -l walltime=1:00:00
 :
 :
 :
```

・queue: DA_002g, script file: hello.sh, recipient address: userA@test.com

Mail is sent when the job begins execution (option is '-m b') and terminates (option is '-m e').

```
$ qsub -q DA_002g -M userA@test.com -m be hello.sh
```

Description in the execution script.

```
#!/bin/sh
#PBS -q DA_002g
#PBS -M userA@test.com
#PBS -m be
 :
 :
 :
```

・queue: A_004, job name: TEST, script file: hello.sh

```
$ qsub -q A_004 -N TEST hello.sh
```

Description in the execution script.

```
#!/bin/sh
#PBS -q A_004
#PBS -N TEST
 :
 :
 :
```

4.1.2 Format of an execution script

This section describes the format of execution script files to run programs on the accelerator server.

To execute the application that requires an execution script file, create the file in advance. /work area have better I/O performance than /home area, so copy your data to the /work area, execute it, and move the result to the source directory with reference to the following example. For more details, see each manual.

（1）Execute a non MPI program

```
#!/bin/sh
#PBS -l select=1
#PBS -q queue
#PBS -N jobname
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

program > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm
-rf $WORKDIR; fi
```

Copy the job input directory to /work area and move there

Execute your program

Move the result to the source directory after execution

・Example　To execute a program ‘a.out’.

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N sample
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

./a.out > result.out 2> result.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm
-rf $WORKDIR; fi
```

(2) Execute a MPI program

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks | -ppn MPI tasks per node ]
-hostfile $PBS_NODEFILE program > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm
-rf $WORKDIR; fi
```

Copy the job input directory to /work area and move there

Execute your program

Move the result to the source directory after execution

・Example To run a program on 2 node and 72 MPI processes using Intel compiler.

```
#!/bin/sh
#PBS -l select=2
#PBS -q A_004
#PBS -N mpi
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 72 -ppn 36 -hostfile $PBS_NODEFILE ./a.out >
result.out 2> result.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm
-rf $WORKDIR; fi
```

4.1.3 Interactive mode

Submit a job with interactive mode.

Add option -I (uppercase i) to the qsub command and specify IA_001g, CA_001, CA_001g or IC_001 for the queue.

(1) Usage

```
$ qsub -I -q queue
```

(2) Example

```
$ qsub -I -q IA_001g
qsub: waiting for job 22351.gpu1 to start
qsub: job 22351.gpu1 ready

-bash-4.2$./a.out
```

4.1.4 Submitting jobs to Shared-Queue CA_001 and CA_001g

CA_001 and CA_001g is a queue used by sharing the node with other jobs.
1CPU and 1GPU are assigned to jobs by default, up to 18 CPU and 5 GPU.
Execution in interactive mode is also available.
Specify CA_001 or CA_001g for the queue.

(1) Usage

```
$ qsub -q queue [ -I ] [ -l select=1[:ncpus= number of CPU][:ngpus= number of GPU] [execution script file]
```

(2) Example

・ queue: CA_001, Command to execute interactive mode with 2 CPU and 1 GPU

```
$ qsub -I -q CA_001 -l select=1:ncpus=2:ngpus=1
qsub: waiting for job 22351.gpu1 to start
qsub: job 22351.gpu1 ready

-bash-4.2$./a.out
```

・ queue: CA_001g, Script to execute program a.out with 18 CPU and 5GPU

To use the queue CA_001g, please submit a job from /work_da area.

```
#!/bin/sh
#PBS -l select=1:ncpus=18:ngpus=5
#PBS -q CA_001g
#PBS -N sample

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 18 -ppn 18 -hostfile $PBS_NODEFILE./a.out > result.out 2> result.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

4.2 Job management command

For more details, see each manual.

4.2.1 Display your own job information

(1) Description

Display your own job information on the supercomputer.

(2) Usage

```
statj [-x]  [ [job_identifier | destination] ...]
```

(3) List of options

Option	Value
-x	Displays status of finished, queued, and running jobs.

(4) Example

```
userA@gpu2:~> statj
                                                  Req'd   Req'd   Elap
Job ID     Username Queue  Jobname  SessID  NDS TSK Memory Time  S Time
---------------  ------------ ---------- ------------------
3413.sdb    userA    P_016  STDIN    231503  1  36  690gb 24:00 R 00:00
```

4.2.2 Display job information

(1) Description

Display information of the jobs on the accelerator server.

(2) Usage

```
Default format:
qstat [-a] [-p] [-J] [-t] [-x] [ [job_identifier | destination] ...]
Long format:
qstat -f [-p] [-J] [-t] [-x] [ [job_identifier | destination] ...]
```

（3）List of options

Option	Value
-a	Display memory usage, elapsed time, status of jobs, etc.
-p	Display the percentage of the job completion.
-J	Display limits status of job array.
-t	Displays status of jobs.
-x	Displays status of finished, queued, and running jobs.
-f	Display status in long format.

(4) Example

```
userA@gpu2:~> qstat -a
                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
3390.gpu1       userA    A_004    abinit      193347   4 144 2760gb 72:00 R 47:28
3401.gpu1       userA    A_004    prog9_1     121974   4 144 2760gb 72:00 R 47:26

userA@gpu2:~> qstat -p
Job id           Name         User        % done  S Queue
---------------  ---------------- ----------------  -------- - -----
3390.gpu1        abinit       userA       2  R A_004
3401.gpu1        prog9_1      userA       0  R A_004

userA@gpu2:~> qstat -t
Job id           Name         User      Time Use S Queue
---------------  ---------------- ----------------  -------- - -----
3390.gpu1        abinit        userA     00:00:01 R A_004
3401.gpu1        prog9_1       userA     00:00:01 R A_004

userA@gpu2:~ > qstat -x
Job id           Name           User     Time Use S Queue
---------------  ---------------- ----------------  -------- - -----
2235.gpu1        prog9_2        userA     00:00:03 F A_016
2236.gpu1        vasp4          userA     00:00:01 F A_016
2237.gpu1        prog9_1        userA     00:00:01 F A_016
        The rest is omitted

userA@gpu2:~> qstat -f 3390.gpu1
Job Id: 3390.gpu1
    Job_Name = abinit
    Job_Owner = userA@gpu2
    resources_used.cpupercent = 10
    resources_used.cput = 00:00:01
    resources_used.mem = 12836kb
    resources_used.ncpus = 72
        The rest is omitted
```

4.2.3 Display queue status

(1) Description

Display information about queues on the accelerator server.

(2) Usage

```
Default format:
statq [destination ...]
Long format:
statq -f [destination ...]
```

(3) List of options

Option	Value
-f	Display status in long format.

(4) Example

```
userA@gpu2:~> statq
Queue      Max  Tot  Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
---------------- ----- ----- --- --- ----- ----- ----- ----- ----- -----
workq        0    0   no  yes    0     0     0     0     0     0 Exec
A_004        0    0   yes  yes    0     0     0     0     0     0 Exec
A_008        0    1   yes  yes    0     1     0     0     0     0 Exec
A_016        0    0   yes  yes    0     0     0     0     0     0 Exec
DA_002g      0    0   yes  yes    0     0     0     0     0     0 Exec
DC_002       0    0   yes  yes    0     0     0     0     0     0 Exec
C_002        0     0   yes  yes   0     0     0     0     0     0 Exec
       The rest is omitted

userA@gpu2:~> statq -f
Queue: workq
    queue_type = Execution
    total_jobs = 0
    state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:0
Exiting:0 Begun
        :0
    enabled = False
    started = True
       The rest is omitted
```

4.2.4 Display server status

(1) Description

Display information about servers of the accelerator server.

(2) Usage

```
Default format:
qstat -B [destination ...]
Long format:
qstat -B -f [destination ...]
```

(3) List of options

Option	Value
-B	Display server status.
-f	Display status in long format.

(4) Example

```
userA@gpu2:~> qstat -B
Server     Max  Tot   Que   Run   Hld   Wat   Trn   Ext Status
---------------- ----- ----- ----- ----- ----- ----- ----- -----
gpu1        0  1155    0     1     0     0     0     0 Active
userA@gpu2:~> qstat -Bf
Server: sdb
    server_state = Active
    server_host = sdb
    scheduling = True
    max_queued = [u:PBS_GENERIC=200]
       The rest is omitted
```

4.2.5 Cancel the job before the job finished

(1) Description

Use qdel command for cancelling the job on the accelerator server.

(2) Usage

```
qdel [ -x ] [ -Wsuppress_email=<N> ] job_identifier
[job_identifier ...]
```

(3) List of options

Option	Value
-x	Delete job and job history of the specified job.
-Wsuppress_email	Set limit on number of e-mails sent on deleting jobs.

(4) Example

```
userA@gpu2:~/work/20180712_sample> qstat
Job id           Name            User             Time Use S Queue
---------------  ---------------- ----------------  -------- - -----
3413.gpu1       abinit          userA          00:00:00 R A_004
3414.gpu1     STDIN_gpu2_22   userA          00:00:00 R A_004
userA@gpu2:~/work/20180712_sample> qdel 3414.gpu1
userA@gpu2:~/work/20180712_sample> qstat
Job id           Name            User             Time Use S Queue
---------------  ---------------- ----------------  -------- - -----
3413.gpu1       abinit          userA          00:00:00 R A_004
userA@gpu2:~/work/20180712_sample>
```

4.3 Display information about used and remained time of job execution (jobtime command)

(1) Description

Use jobtime command for listing information of your completed jobs.

(2) Usage

```
jobtime
```

(3) Information

Section	Details
Last Updated	updated time
User	user ID
Total	available time of job execution
Used	used time
Remained	remained time

(4) Example

```
userA@gpu2:~ >  jobtime

# Last Updated:   2018/10/01 13:45
# User       Total      Used   Remained (H)
username       500    222.32    277.68
```

4.4 References for submitting job and script

4.4.1 How to execute a MPI job

(1) Description

Intel MPI is available for MPI environment.

(2) Usage

Use mpirun command to execute a job.

```
mpirun [ -np parallel number] [ -ppn parallel number per node ]
-hostfile $ PBS_NODEFILE execution program
```

*To improve job performance, specify values so that:

[*Parallel number*(the value of "-np")] = [*Number of nodes*(the value of "#PBS -l select=")] × [*Parallel number per node*(the value of "-ppn")]

(3) Example

```
#!/bin/bash
#PBS -j oe
#PBS -l select=1
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36  -hostfile $PBS_NODEFILE
/usr/local/app/ABINIT/current/src/98_main/abinit < input.files >
result.out 2> result.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf
$WORKDIR; fi
```

4.5 List of queues

Queues of the accelerator server are as follows.

Execlusive-Queue

Queue name	Limit of number of exclusive nodes (default)	Memory Limit [GiB]	Walltime limit [hours] (default)	Limit of concurrent execution	Upper Limit of using GPU	Limit of number of parallel execution	Notes
IA_001g	1(1)	690	24(1)	6	10	36	for interactive
DA_002g	2(1)	1,380	2(2)	6	20	72	For debugging
A_004	4(1)	2,760	72(24)	No limit	40	144	
IC_001	1(1)	510	24(1)	No limit	-	36	for interactive
DC_002	2(1)	1,020	2(2)	No limit	-	72	For debugging
C_002	2(1)	1,020	72(24)	No limit	-	72	
C_004	4(1)	2,040	72(24)	No limit	-	72	

Shared-Queue

Queue name	Limit of number of CPU (default)	Limit of number of GPU (default)	Memory Limit [GiB]	Walltime limit [hours] (default)	Limit of concurrent execution	Limit of number of parallel execution	Notes
CA_001	18(1)	5(1)	345(69)	72(24)	10	18	Interactive mode available
CA_001g	18(1)	5(1)	345(69)	72(24)	10	18	Interactive mode available

Notice: To use the queue IA_001g,CA_001g and DA_002g, please submit a job from /work_da area.

We support researchers who want to develop application software for ultra-large-scale calculations on MASAMUNE-IMR and a new methodology for Computational Materials Science. We also offer following job queues for these calculations and you can use them by submitting your application. As for the application, please contact us using the contact form.

Queue name	Limit of number of exclusive nodes (default)	Memory Limit [GiB]	Walltime limit [hours] (default)	Limit of concurrent execution	Upper Limit of using GPU	Limit of number of parallel execution	Notes
LA_004	4(4)	2,760	168(96)	1	40	144	executed during operation
MA_008	8(8)	5,520	72(24)	1	80	288	
SA_016	16(16)	11,040	72(72)	1	160	576	executed after maintenance operation only

5 Compilers and Libraries

5.1 Compilers

5.2 Libraries

5.1 Compiler

Following compilers are available on the accelerator server.

For more details, see each manual.

Compiler	Version	Notes
Intel Compiler Fortran/C/C++	19.1.3.304 19.1.0.166 19.0.2.187 18.0.3.222 17.0.4.196	default: 17.0.4.196
PGI Compiler Fortran/C/C++	20.4 19.10 19.1 18.10 18.5	default: 19.1
nvcc Compiler (CUDA Toolkit)	10.2.89 10.1.243 9.2.148 9.0.176 8.0.44	default: 9.0.176

5.1.1 Intel Compiler

(1) Setting of program environment

Intel compiler is the default one on the accelerator server.

If you want to change the compiler version, execute the following command.

```
$ module avail intel
intel/17.0.4(default) intel/18.0.3 intel/19.0.2
intel/19.1.0 intel/19.1.3

$ module switch intel/17.0.4 intel/18.0.3
```

(2) How to compile

(A) Command

・serial

Language	Command	Execution type
Fortran	ifort	ifort [options] files
C	icc	icc [options] files
C++	icpc	icpc [options] files

・MPI

言語	コマンド	実行形式
Fortran	mpiifort	mpiifort [options] files
C	mpiicc	mpiicc [options] files
C++	mpiicpc	mpiicpc [options] files

(B) Options

・Optimization options and others

Option	Description
-o outfile	Set output file. If '-o' option is not specified, the default output file is a.out.
-llibrary_name	Specify libraries to be linked.
-Llibrary_path	Specify paths to be searched for the libraries.
-O0|-O1|-O2|-O3	Set optimizing options. -O2 is default.
-fast	Set the whole program execution speed maximum. The following options are set automatically. -ipo, -O3, -no-prec-div, -static, -fp-model fast=2, -xHost
-parallel	Compile the input source file enabling auto threading.
-openmp	Compile the input source file enabling OpenMP.
-xcore-avx512	Specify Intel AVX-512 instruction as target. (Recommended)

・Options for Fortran

Option	Description
-free|-fixed	Set the format of source files.

・Debugging Options

Option	Description
-g	Output debugging information.
-traceback	Output traceback when an error occurs.
-fpe[0-3]	Trap exception handling.　(-fpe0 is most detailed)

（3）Example

・Make the execution module 'hello.out' from Fortran source file 'hello.f'in fixed form.

```
$ ifort -xcore-avx512 -fixed -o hello.out hello.f
```

・Make the execution module ’hello.out’ from the Fortran source file ‘hello.f90’ in free form.

```
$ ifort -xcore-avx512 -free -o hello.out hello.f90
```

・Make the execution module 'hello.out' auto threaded from the Fortran source file 'hello.f' in free form.

```
$ ifort --xcore-avx512 -fixed -parallel -o hello.out hello.f
```

・Make the execution module 'hello.out' auto threaded from the Fortran source file 'hello.f90' in free form.

```
$ ifort -xcore-avx512 -free -parallel -o hello.out hello.f90
```

・Make the execution module 'hello.out' from C source file 'hello.c'.

```
$ icc -xcore-avx512 -o hello.out hello.c
```

・Make the execution module 'hello.out' auto threaded from C source file 'hello.c'.

```
$ icc -xcore-avx512 -parallel -o hello.out hello.c
```

・Make the execution module 'hello.out' from the C++ source file 'hello.cpp'.

```
$ icpc -xcore-avx512 -o hello.out hello.cpp
```

・Make the execution module 'hello.out' auto threaded from the C++ source file.

```
$ icpc -xcore-avx512 -parallel -o hello.out hello.cpp
```

5.1.2 PGI Compiler

(1) Setting of program environment

When the program environment is changed, execute the following commands.

```
$ module avail PrgEnv-pgi
PrgEnv-pgi/18.5 PrgEnv-pgi/18.10 PrgEnv-pgi/19.1(default)
PrgEnv-pgi/19.10 PrgEnv-pgi/20.4

$ module switch intel PrgEnv-pgi
```

(2) How to compile

(A) Command

・serial

Language	Command	Execution type
Fortran	pgf90	pgf90 [options] files
C	pgcc	pgcc [options] files
C++	pgc++	pgc++ [options] files

・MPI

Language	Command	Execution type
Fortran	mpif90	mpif90 [options] files
C	mpicc	mpicc [options] files
C++	mpic++	mpic++ [options] files

(B) Options

・Optimization options and others

Option	Description
-o outfile	Set output file. If '-o' option is not specified, the default output file is a.out.
-llibrary_name	Specify libraries to be linked.
-Llibrary_path	Specify paths to be searched for the libraries.
-O0|-O1|-O2|-O3|-O4	Set optimizing options. -O2 is default.
-fast	Enable general optimization flag.
-Mconcur	Compile the input source file enabling auto threading.
-mp	Compile the input source file enabling OpenMP.

・Option for Fortran

Option	Description
-Mfree|-Mfixed	Set the format of source files.

・Debugging options

Option	Description
-g | -gopt	Output debugging information.

(3) Example

・Make the execution module 'hello.out' from the Fortran source file 'hello.f' in fixed form.

```
$ pgf90 -Mfixed -o hello.out hello.f
```

・Make the execution module 'hello.out' auto threaded from the Fortran source file 'hello.f' in fixed form.

```
$ pgf90 -Mfixed -Mconcur -o hello.out hello.f
```

・Make the execution module 'hello.out' using OpenMP from the Fortran source file 'hello.f' in fixed form.

```
$ pgf90 -mp -Mfixed -o hello.out hello.f
```

・Make the execution module ’hello.out’ from the Fortran source file ‘hello.f90’ in free form.

```
$ pgf90 -Mfree -o hello.out hello.f90
```

・Make the execution module 'hello.out' auto threaded from the Fortran source file 'hello.f90' in free form.

```
$ pgf90 -Mfree -Mconcur -o hello.out hello.f90
```

・Make the execution module 'hello.out' using OpenMP from the Fortran source file 'hello.f90' in free form.

```
$ pgf90 -mp -Mfree -o hello.out hello.f90
```

・Make the execution module 'hello.out' from C source file 'hello.c'.

```
$ pgcc -o hello.out hello.c
```

・Make the execution module 'hello.out' auto threaded from C source file 'hello.c'.

```
$ pgcc -Mconcur -o hello.out hello.c
```

・Make the execution module 'hello.out' using OpenMP from C source file 'hello.c'.

```
$ pgcc -mp -o hello.out hello.c
```

・Make the execution module 'hello.out' from the C++ source file 'hello.cpp'.

```
$ pgc++ -o hello.out hello.cpp
```

・Make the execution module 'hello.out' auto threaded from the C++ source file 'hello.cpp'

```
$ pgc++ -Mconcur -o hello.out hello.cpp
```

・Make the execution module 'hello.out' using OpenMP from the C++ source file 'hello.cpp'.

```
$ pgc++ -mp -o hello.out hello.cpp
```

5.1.3 GNU Compiler

（1）Setting of program environment

Nothing to do.

（2）How to compile

(A) Command

・serial

Language	Command	Execution type
Fortran	gfortran	gfortran [options] files
C	cc	cc [options] files
C++	g++	g++ [options] files

・MPI

Language	Command	Execution type
Fortran	mpif90	mpif90 [options] files
C	mpicc	mpicc [options] files
C++	mpicxx	mpicxx [options] files

(B) Options

・Optimization options and others

Option	Description
-o outfile	Set output file. If '-o' option is not specified, the default output file is a.out.
-llibrary_name	Specify libraries to be linked.
-Llibrary_path	Specify paths to be searched for the libraries.
-O0|-O1|-O2|-O3|-O4	Set optimizing options. -O2 is default.
-fopenmp	Compile the input source file enabling OpenMP.

・Options for Fortran

Option	Description
-ffree-form|-ffixed-form	Set the format of source files.

・Debugging options

Option	Description
-g	Output debugging information.
-g0|-g1|-g2|-g3	Manage debugging information.(-g2 = -g)

（3）Example

・Make the execution module 'hello.out' from Fortran source file 'hello.f' in fixed form.

```
$ gfortran -ffixed-form -o hello.out hello.f
```

・Make the execution module ’hello.out’ from the Fortran source file ‘hello.f90’ in free form.

```
$ gfortran -ffree-form -o hello.out hello.f90
```

・Make the execution module 'hello.out' from C source file 'hello.c'.

```
$ cc -o hello.out hello.c
```

・Make the execution module 'hello.out' from the C++ source file 'hello.cpp'.

```
$ g++ -o hello.out hello.cpp
```

5.1.4 nvcc Compiler

(1) Setting of program environment

The CUDA is available as default.

If you want to change the version, execute the following command.

```
$ module switch cudatoolkit/9.0.176 cudatoolkit/10.1.243
```

・Because the Intel compiler is set up as default, you use Intel compiler for backend compiler.

・Using PGI compiler for backend compiler, execute the following commands to set up PGI compiler.

```
$ module switch intel PrgEnv-pgi
```

(2) How to compile

(C) Options

・Backend compiler, optimization options and others

Option	Description
-ccbin Compiler	Specify backend compiler. Intel compiler: Icpc PGI compiler: pgc++
-O0|1|…	Set optimizing options. (pass to the backend compiler)
-Xcompiler options	Set compiler options except for optimizing options. (pass to the backend compiler)
-gencode options	Specify CUDA's version of generating code.
--machine {32|64} (-m)	Specify the bits.
-I include_path	Specify include search paths.
-L library_path	Specify paths to be searched for the libraries.
-l library_name	Specify libraries to be linked.
--help (-h)	Display list and description of available options.
--version (-V)	Display version information.

(3) Example

・In case of specifying Intel compiler (icc) for backend compiler,

```
$ nvcc -ccbin icpc -m64 -gencode arch=compute_70,code=compute_70
-o simple simple.cpp
```

・In case of specifying PGI compiler (pgc++) for backend compiler,

```
$ nvcc -ccbin pgc++ -m64 -gencode arch=compute_70,code=compute_70
-o simple simple.cpp
```

5.2 Libraries

Following libraries are available for compiling and linking on the accelerator server.
For more details, see each manual.

Library	Version	Compiler	Notes
Intel MKL (Intel Math Kernel Library)	19.1.3.304 19.1.0.166 19.0.2.187 18.0.3.222 17.0.4.196	Intel compiler	
cuBLAS	9.0 8.0	Intel compiler PGI compiler	
cuDNN	7.6.5 for CUDA10.2 7.6.3 for CUDA10.1 7.6.5 for CUDA9.2 7.6.3 for CUDA9.0	Intel compiler PGI compiler	

5.2.1 Intel MKL

Intel MKL (Intel Math Kernel Library) contains BLAS, LAPACK, SparseBLAS, PARDISO, Iterative Sparse Solver, FFT, random number generation and so on.

(1) Setting program environment

Intel MKL is default.

Change the compiler version as follows.

```
$ module switch intel/17.0.4 intel/18.0.3
```

(2) Example

・Make the execution module 'hello.out' from the Fortran source file 'hello.f90' using BLAS in fixed form.

```
$ ifort -mkl -o hello.out -fixed hello.f
```

・Make the execution module 'hello.out' from C source file 'hello.c' using BLAS.

```
$ icc -mkl -o hello.out hello.c
```

5.2.2 cuBLAS

The cuBLAS is BLAS library supporting the CUDA.

（1）Setting of program environment

・The CUDA including the cuBLAS is available as default.

If you want to change the version, execute the following command.

```
$ module switch cudatoolkit/9.0.176 cudatoolkit/8.0.44
```

・Because the Intel compiler is set up as default, you use Intel compiler for backend compiler.

・Using PGI compiler for backend compiler, execute the following commands to set up PGI compiler.

```
$ module switch intel PrgEnv-pgi
```

（2）Example

・In case of specifying Intel compiler (icc) for backend and linking cuBLAS libraries,

```
$ nvcc -ccbin icpc -I../../common/inc -m64 -gencode
 arch=compute_70,code=compute_70  -o simpleCUBLAS
 simpleCUBLAS.cpp -l cublas
```

・In case of specifying PGI compiler (pgc++) for backend and linking cuBLAS libraries,

```
$ nvcc -ccbin pgc++ -I../../common/inc -m64 -gencode
arch=compute_70,code=compute_70  -o simpleCUBLAS
simpleCUBLAS.cpp -l cublas
```

5.2.3 cuDNN

（1）Setting of program environment

・The CUDA including the cuDNN is available as default.

If you want to change the version, execute the following command.

```
$ module switch cudatoolkit/9.0.176 cudatoolkit/10.1.243
```

・Because the Intel compiler is set up as default, you use Intel compiler for

backend compiler.

・Using PGI compiler for backend compiler, execute the following commands to set up PGI compiler.

```
$ module switch intel PrgEnv-pgi
```

(2) Example

・In case of specifying Intel compiler (icc) for backend and linking cuDNN libraries,

```
$ nvcc -ccbin icpc -I../../common/inc -m64 -gencode
 arch=compute_70,code=compute_70  -o simpleCUBLAS
 simpleCUBLAS.cpp -l cudnn
```

・In case of specifying PGI compiler (pgc++) for backend and linking cuDNN libraries,

```
$ nvcc -ccbin pgc++ -I../../common/inc -m64 -gencode
arch=compute_70,code=compute_70  -o simpleCUBLAS
simpleCUBLAS.cpp -l cudnn
```

6

6 Usage of applications

6.1 Applications

6.2 VASP

6.3 QUANTUM ESPRESSO

6.4 LAMMPS

6.5 Gaussian16

6.6 CRYSTAL

6.7 WIEN2k

6.8 SIESTA

6.9 ABINIT

6.10 CPMD

6.11 MaterialsStudio

6.12 Wannier90

6.1 Applications

The following applications are available on the accelerator server.

#	Application	Version	Execution type	Queue
1	VASP	5.4.4 (gpu) 6.1.0 (gpu) 6.1.1 (gpu) 6.1.2 (gpu) 6.2.0 (gpu)	MPI	A_004 CA_001 CA_001g
		4.6.36 5.4.4 6.1.0 6.1.1 6.1.2	MPI	C_002 C_004
2	QUANTUM ESPRESSO	6.1 (gpu tag v1.0) 6.4.1 (gpu) 6.5 (gpu) 6.6 (gpu) 6.7 (gpu)	MPI	A_004
		6.2.1 6.4.1	MPI	C_002 C_004
3	LAMMPS	31 Mar 17 5 Jun 19	MPI	A_004 CA_001 CA_001g C_002 C_004
		12 Dec 18 7 Aug19 3 Mar 20 29 Oct 20	MPI	A_004 CA_001 CA_001g
4	Gaussian16	Rev B.01	SMP	C_002 C_004
		Rev C.01	SMP	A_004 C_002 C_004

5	CRYSTAL	17	MPI SMP	C_002 C_004
6	WIEN2k	17.1 19.1 19.2	SMP	C_002 C_004
7	SIESTA	4.0 4.1.5	MPI	C_002 C_004
8	ABINIT	8.8.2 8.10.3	MPI	C_002 C_004
9	CPMD	4.1 4.3	MPI	C_002 C_004
10	Materials Studio	2020 2019	MPI	C_002 C_004
11	Wannier90	1.2 2.1.0	serial	A_004
		3.1.0	MPI	

6.2 VASP

VASP is available for users who have a license. If you have the license and want to use VASP, contact our center. After checking the license, VASP is ready for you.

Available executables

*Note: When using VASP6.1.1 or VASP6.1.2 on the accelerator server, switch the module to "intel 18.0.3" and "CUDA 10.2.89".

*Note: When using VASP6.2.0 on the accelerator server, switch the module to "intel 19.0.2" and "CUDA 10.2.89".

*Note: When using VASP6.1.1 , VASP6.1.2 or non-collinear version on parallel computing and informatics server, switch the module to "intel 19.1.0".

Version	Path	Queue
VASP5.4.4 gpu ver.	/usr/local/app/VASP5/current/bin/vasp_gpu	A_004 CA_001 CA_001g
VASP5.4.4 gpu+ non-collinear ver.	/usr/local/app/VASP5/current/bin/vasp_gpu_ncl	A_004 CA_001 CA_001g
VASP6.1.0 gpu ver.	/usr/local/app/VASP6/vasp.6.1.0/bin/vasp_gpu	A_004 CA_001 CA_001g
VASP6.1.0 gpu+ non-collinear ver.	/usr/local/app/VASP6/vasp.6.1.0/bin/vasp_gpu_ncl	A_004 CA_001 CA_001g
VASP6.1.1 gpu ver.	/usr/local/app/VASP6/current/bin/vasp_gpu *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP6.1.1 gpu+ non-collinear ver.	/usr/local/app/VASP6/current/bin/vasp_gpu_ncl *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP6.1.2 gpu ver.	/usr/local/app/VASP6/vasp.6.1.2/bin/vasp_gpu *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP6.1.2 gpu+ non-collinear ver.	/usr/local/app/VASP6/vasp.6.1.2/bin/vasp_gpu_ncl *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP6.2.0 gpu ver.	/usr/local/app/VASP6/vasp.6.2.0/bin/vasp_gpu *Switch the module to "intel 19.0.2" and " CUDA 10.2.89".	A_004 CA_001 CA_001g

VASP6.2.0 gpu+ non-collinear ver.	/usr/local/app/VASP6/vasp.6.2.0/bin/vasp_gpu_ncl *Switch the module to "intel 19.0.2" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP4.6.36	/usr/local/app/VASP4/current/vasp	C_002 C_004
VASP4.6.36 Gamma point ver.	/usr/local /app/VASP4/vasp.4.6_gamma/vasp	C_002 C_004
VASP5.4.4 Standard ver.	/usr/local /app/VASP5/current/bin/vasp_std	C_002 C_004
VASP5.4.4 Gamma point ver.	/usr/local /app/VASP5/current/bin/vasp_gam	C_002 C_004
VASP5.4.4 non-collinear ver.	/usr/local /app/VASP5/current /bin/vasp_ncl *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.0 Standard ver.	/usr/local/app/VASP6/vasp.6.1.0/bin/vasp_std	C_002 C_004
VASP6.1.0 Gamma point ver.	/usr/local/app/VASP6/vasp.6.1.0/bin/vasp_gam	C_002 C_004
VASP6.1.0 non-collinear ver.	/usr/local/app/VASP6/vasp.6.1.0/bin/vasp_ncl *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.1 Standard ver.	/usr/local/app/VASP6/current/bin/vasp_std *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.1 Gamma point ver.	/usr/local/app/VASP6/current/bin/vasp_gam *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.1 non-collinear ver.	/usr/local/app/VASP6/current /bin/vasp_ncl *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.2 Standard ver.	/usr/local/app/VASP6/vasp.6.1.2/bin/vasp_std *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.2 Gamma point ver.	/usr/local/app/VASP6/vasp.6.1.2/bin/vasp_gam *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.2 non-collinear ver.	/usr/local/app/VASP6/vasp.6.1.2/bin/vasp_ncl *Switch the module to "intel 19.1.0".	C_002 C_004

VASP 6.1.1 and VASP 6.1.2 executables linked with Wannier90 are also available.

Version	Path	Queue
VASP6.1.1 gpu (Wannier90) + Standard ver.	/usr/local/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_gpu /usr/local/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_gpu *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g

VASP6.1.1 gpu (Wannier90) + non-collinea rver.	/usr/local/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_gpu_ncl /usr/local/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_gpu_ncl *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP6.1.1 (Wannier90) Standard ver.	/usr/local/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_std /usr/local/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_std *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.1 (Wannier90) Gamma point ver.	/usr/local/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_gam /usr/local/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_gam *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.1 (Wannier90) non-collinear ver.	/usr/local/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_ncl /usr/local/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_ncl *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.2 gpu (Wannier90) + Standard ver.	/usr/local/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_gpu /usr/local/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_gpu *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP6.1.2 gpu (Wannier90) + non-collinea rver.	/usr/local/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_gpu_ncl /usr/local/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_gpu_ncl *Switch the module to "intel 18.0.3" and " CUDA 10.2.89".	A_004 CA_001 CA_001g
VASP6.1.2 (Wannier90) Standard ver.	/usr/local/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_std /usr/local/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_std *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.2 (Wannier90) Gamma point ver.	/usr/local/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_gam /usr/local/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_gam *Switch the module to "intel 19.1.0".	C_002 C_004
VASP6.1.2 (Wannier90) non-collinear ver.	/usr/local/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_ncl /usr/local/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_ncl *Switch the module to "intel 19.1.0".	C_002 C_004

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks][ -ppn MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/VASP5/current/bin/vasp_gpu > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) Accelerator Server

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N vasp

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 10 -ppn 10 -hostfile $PBS_NODEFILE
/usr/local/app/VASP5/current/bin/vasp_gpu > vasp.out 2> vasp.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server (excluding non-collinear ver.)

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N vasp

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE  /usr/local/app/VASP4/current/vasp
> vasp.out 2> vasp.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server (non-collinear ver.)

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N vasp

module switch intel intel/19.1.0

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE
/usr/local/app/VASP5/current/bin/vasp_ncl > vasp.out 2> vasp.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.3 QUANTUM ESPRESSO

Available executables

Version	Path	Queue
6.1 (gpu tag v1.0)	/usr/local/app/QuantumESPRESSO/current	A_004
6.2.1	/usr/local/app/QuantumESPRESSO/current	C_002 C_004
6.4.1 (gpu)	/usr/local/app/QuantumESPRESSO/qe-gpu-6.4.1	A_004
6.4.1	/usr/local/app/QuantumESPRESSO/qe-6.4.1	C_002 C_004
6.5(gpu)	/usr/local/app/QuantumESPRESSO/qe-gpu-6.5	A_004
6.6(gpu)	/usr/local/app/QuantumESPRESSO/qe-gpu-6.6 *Switch the module to “pgi 20.4” and “CUDA 10.1.243”.	A_004
6.7(gpu)	/usr/local/app/QuantumESPRESSO/qe-gpu-6.7 *Switch the module to “pgi 20.4” and “CUDA 10.1.243”. *Add "/opt/intel/mkl/lib/intel64" to the environment variable LD_LIBRARY_PATH	A_004

Create a script file in advance for Accelerator Server.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

module switch intel PrgEnv-pgi/18.5

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks ] [ -N MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/QuantumESPRESSO/current/bin/pw.x < input file > output file 2>
error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) Accelerator Server

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N espresso

module switch intel PrgEnv-pgi/18.5

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 10 -N 10 -hostfile $PBS_NODEFILE
/usr/local/app/QuantumESPRESSO/current/bin/pw.x < cluster4.in > qe.out 2>
qe.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

Specify MPI tasks as 10 or less.

(Example) Accelerator Server(6.7(gpu))

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N espresso

module switch intel PrgEnv-pgi/20.4
module switch cudatoolkit cudatoolkit/10.2.89

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/intel/mkl/lib/intel64

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 10 -N 10 -hostfile $PBS_NODEFILE /usr/local/app/QuantumESPRESSO/
qe-gpu-6.7/bin/pw.x < cluster4.in > qe.out 2> pe.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

Specify MPI tasks as 10 or less.

Create a script file in advance for parallel computing and informatics server.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks ][ -ppn MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/QuantumESPRESSO/current/bin/pw.x < input file > output file 2>
error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N espresso

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE
/usr/local/app/QuantumESPRESSO/current/bin/pw.x < cluster4.in > qe.out 2> qe.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.4 LAMMPS

Available executables

*Note: When using 7Aug19 or 3Mar2020 on the accelerator server, switch the module to "CUDA 10.1.243".

*Note: When using 29 Oct 20 on the accelerator server, switch the module to "CUDA 10.2.89".

Version	Path	Queue
31 Mar 17	/usr/local/app/LAMMPS/current	A_004 CA_001 CA_001g C_002 C_004
12 Dec 18	/usr/local/app/LAMMPS/lammps-12Dec18	A_004 CA_001 CA_001g
5 Jun 19	/usr/local/app/LAMMPS/lammps-5Jun19	A_004 CA_001 CA_001g C_002 C_004
5 Jun 19 -DFFT_SINGLE OFF	/usr/local/app/LAMMPS/lammps-5Jun19_wo_single	A_004 CA_001 CA_001g
7 Aug 19	/usr/local/app/LAMMPS/lammps-7Aug19 *Switch the module to "CUDA 10.1.243".	A_004 CA_001 CA_001g
3 Mar 20	/usr/local/app/LAMMPS/lammps-3Mar20 *Switch the module to "CUDA 10.1.243".	A_004 CA_001 CA_001g
29 Oct 20	/usr/local/app/LAMMPS/lammps-29Oct20 *Switch the module to "CUDA 10.2.89".	A_004 CA_001 CA_001g

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks ][ -ppn MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/LAMMPS/current/src/lmp_gpu -sf gpu -pk gpu GPUs per node < input
file > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) Accelerator Server

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N lammps

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 30 -ppn 30 -hostfile $PBS_NODEFILE
 /usr/local/app/LAMMPS/current/src/lmp_gpu -sf gpu -pk gpu 10 < in.ij > lammps.out
 2> lammps.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) Accelerator Server（7 Aug 19）

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N lammps

module switch cudatoolkit/9.0.176 cudatoolkit/10.1.243

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 30 -ppn 30 -hostfile $PBS_NODEFILE
 /usr/local/app/LAMMPS/lammps-7Aug19/src/lmp_gpu -sf gpu -pk gpu 10 < in.ij >
 lammps.out 2> lammps.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N lammps

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE
/usr/local/app/LAMMPS/current/src/lmp_intel_cpu_intelmpi  < in.ij > lammps.out
2> lammps.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.5 Gaussian16

Available executables

Version	Path	Queue
B.01	source /usr/local/app/Gaussian/g16.profile	C_002 C_004
C.01	source /usr/local/app/Gaussian/g16.profile	A_004
	source /usr/local/app/Gaussian/C.01/g16.profile	C_002 C_004

Prepare input files of Gaussian 16(***.com) in the working directory.
The following input file calculates geometry optimization for hexacarbonyl. It specifies Hartree-Fock method and 3-21G basis set.

(Example)

```
$ ls -l /usr/local/app/Gaussian/example.com
-rw-r--r-- 1 root root 420 Jul 12 16:33 /usr/local/app/Gaussian/example.com
```

・The number of threads

Specify ‘CPU=0-35’ in the input file for parallelization. In the case of ‘NProc=36’ is specified, the job cannot run in parallel normally.

・Using GPU

In order to use GPU, it is necessary to specify the number of cores and GPUs.
Below is an example using 36 cores and 10 GPUs.

Specification example in input file
% CPU = 0-35
% GPUCPU = 0-9 = 0-4,18-22

Example of environment variable specification
export GAUSS_CDEF = "0-35"
export GAUSS_GDEF = "0-9 = 0-4,18-22"

Not valid for all calculations.
Consider setting the upper limit of elapsed time.

See also the following page.

http://gaussian.com/gpu/

・The directory for temporary files

Create a directory on /work/scratch area for outputting Gaussian temporary files.

(Example) %Chk=example_app.chk

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=1
#PBS -q queue
#PBS -N jobname

source /usr/local/app/Gaussian/g16.profile
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

g16 input file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) Accelerator Server

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N g16

source /usr/local/app/Gaussian/g16.profile
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

g16 test0000.com 2> g16.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N g16

source /usr/local/app/Gaussian/g16.profile
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

g16 test0000.com 2> g16.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.6 CRYSTAL

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

source /usr/local/app/Crystal/current/utils17/cry17.bashrc
runmpi17 MPI tasks inputfile > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example)parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N crystal

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

source /usr/local/app/Crystal/current/utils17/cry17.bashrc
runmpi17 36 test11 > crystal.out 2> crystal.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.7 WIEN2k

WIEN2k is available for users who have a license. If you have the license and want to use WIEN2k, contact our center. After checking the license, WIEN2k is ready for you.

Available executables

Version	Path	Queue
17.1	/usr/local/app/WIEN2k/current	C_002 C_004
19.1	/usr/local/app/WIEN2k/WIEN2k_19.1	C_002 C_004
19.2	/usr/local/app/WIEN2k/WIEN2k_19.2 *Switch the module to “intel 19.0.1”.	C_002 C_004

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME
export SCRATCH=$WORKDIR/$DIRNAME
export TMPDIR=$WORKDIR/$DIRNAME
export WIENROOT=/usr/local/app/WIEN2k/current
export PATH=$WIENROOT:$PATH

wien2k Script Option > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N wien2k

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME
export SCRATCH=$WORKDIR/$DIRNAME
export TMPDIR=$WORKDIR/$DIRNAME
export WIENROOT=/usr/local/app/WIEN2k/current
export PATH=$WIENROOT:$PATH

run_lapw -p -cc 0.0001 -NI > wien2k.out 2> wien2k.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

You can use k-Point parallelization by using -p option in script file and set up .machines file in the current work directory.

```
$ cat .machines
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
1:localhost
granularity:1
extrafine:1
```

6.8 SIESTA

Available executables

Version	Path	Queue
4.0	/usr/local/app/SIESTA/current	C_002 C_004
4.1.5	/usr/local/app/SIESTA/siesta-4.1.5	C_002 C_004

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks ][ -ppn MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/SIESTA/current/Obj/siesta < input file > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N siesta

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE
/usr/local/app/SIESTA/current/Obj/siesta < input.fdf > siesta.out 2> siesta.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.9 ABINIT

Available executables

Version	Path	Queue
8.8.2	/usr/local/app/ABINIT/current/src/98_main/abinit	C_002 C_004
8.10.3	/usr/local/app/ABINIT/abinit-8.10.3/src/98_main/abinit	C_002 C_004

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks ][ -ppn MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/ABINIT/current/src/98_main/abinit < input file > output file 2>
error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N abinit

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE
/usr/local/app/ABINIT/current/src/98_main/abinit < input.files > abinit.out 2>
abinit.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.10 CPMD

CPMD is available for users who has own license.

To use CPMD, obtain the license on CPMD web site and contact our center.

After checking the license, CPMD is ready for you.

Available executables

Version	Path	Queue
4.1	/usr/local/app/CPMD/current	C_002 C_004
4.3	/usr/local/app/CPMD/CPMD4.3	C_002 C_004

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks ][ -ppn MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/CPMD/current/bin/cpmd.x  input file > output file 2> error file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -q C_002
#PBS -N cpmd

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE
/usr/local/app/CPMD/current/bin/cpmd.x inp-1 > cpmd.out  2> cpmd.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

6.11 MaterialsStudio

Please contact us if you want to use it on your PC.

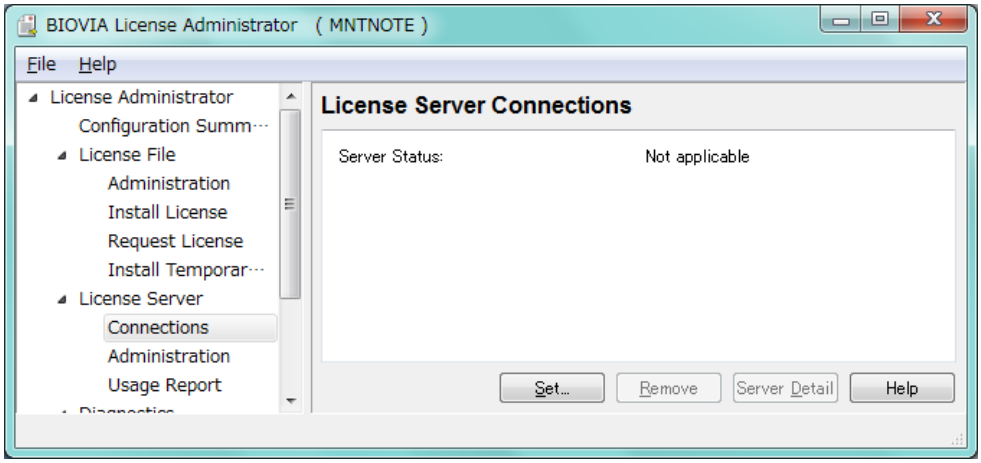
The following modules are available.

Module	Licenses	Functions
Visualizer	8	Construct structural models, creating and showing of input files for a simulation, calculation results, graphs, tables, etc.
CASTEP_Interface	2	Create input files for CASTEP execution, and analyze the result.
CASTEP	16	Simulate the wide range physical properties in ceramics, a semiconductor, the solid in the substance domain of science containing metal, an interface, and the surface.
DMol3_Interface	2	Create input files for DMol3 execution, and analyze the result.
DMol3-Solid_State	16	Predict physical properties in high reliability and at high speed by the ability of high precision computation based on quantum mechanics.
Forcite Plus	3	Predict a structure and molecule relations, an understanding of an intermolecular interaction, and the character of a solid, a liquid, and gas.
DFTB+	1	An improved implementation of the Density Functional based Tight Binding (DFTB) quantum simulation method for the study of electronic properties of materials and offers unique capabilities to study and understand systems containing hundreds of atoms.
Sorption	1	Predict fundamental properties, such as sorption isotherms (or loading curves) and Henry's constants needed for investigating separations phenomena.

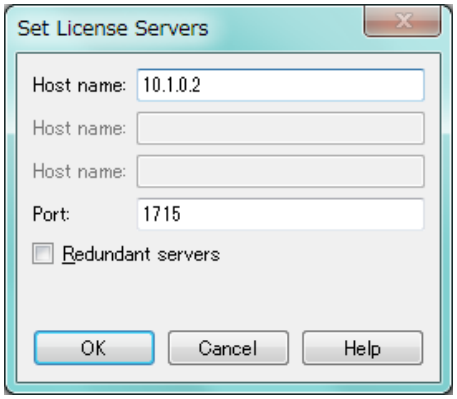
6.11.1 How to set License server

Select [BIOVIA] -[Licensing] - [License Administrator] from the start menu.

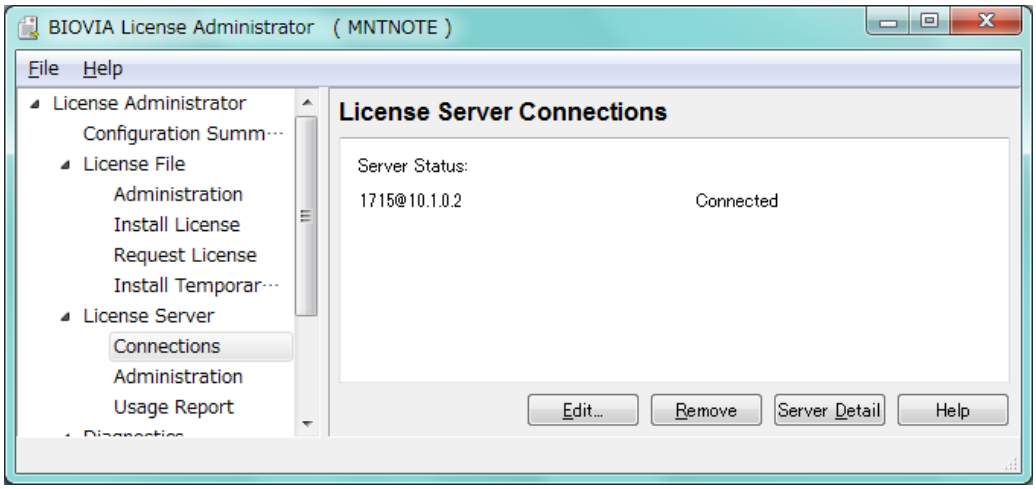
Select [License Server]-[Connections] and push [Set] or[Edit].



Set [Host name] to "10.1.0.2" and [Port] to" 1715", push [OK].

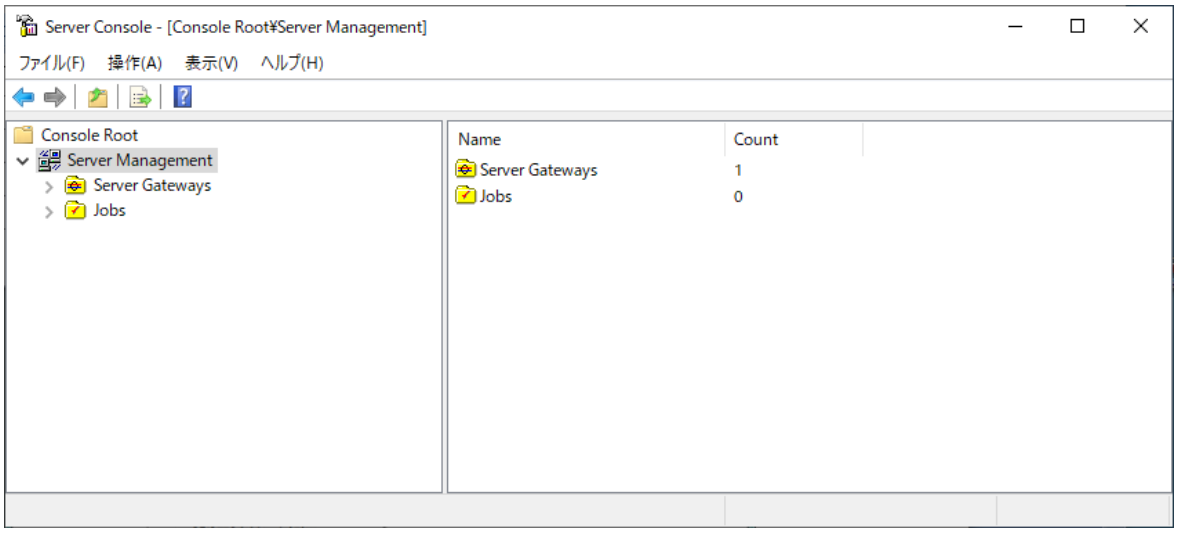


[Server Status] will be displayed as "Connected".

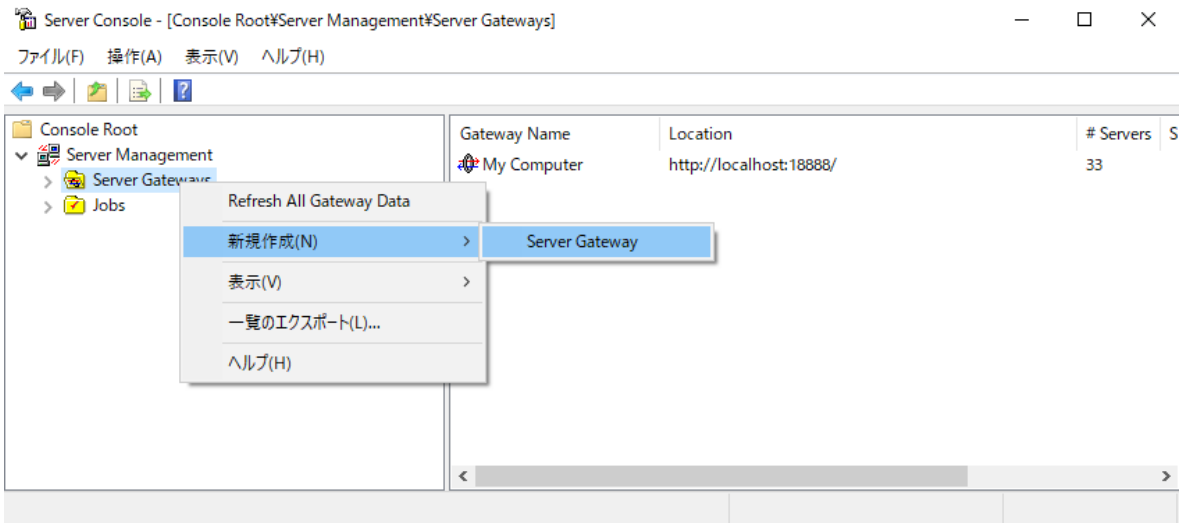


6.11.2 How to set Gateway

Select [BIOVIA] -[Server Console] from the start menu.

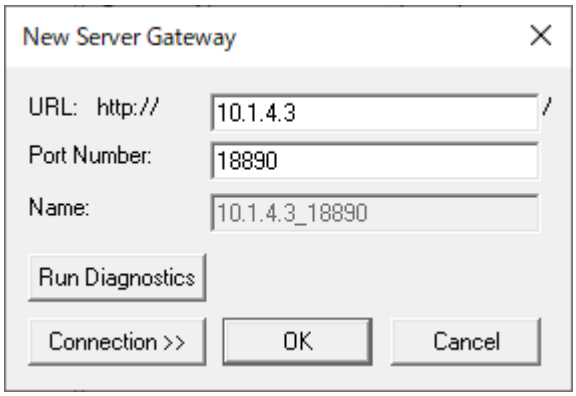


Right click [Server Gateways] and create [Server Gateway].

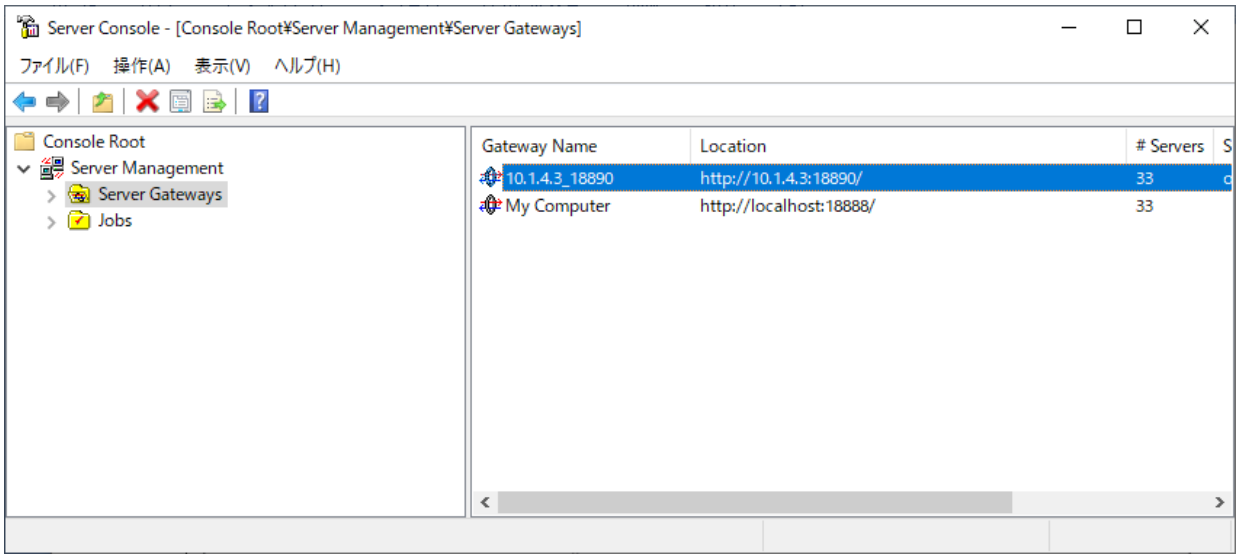


Set [URL] to "10.1.4.3", and [Port Number] to as bellow.

Version	Port Number
2019	18889
2020	18890

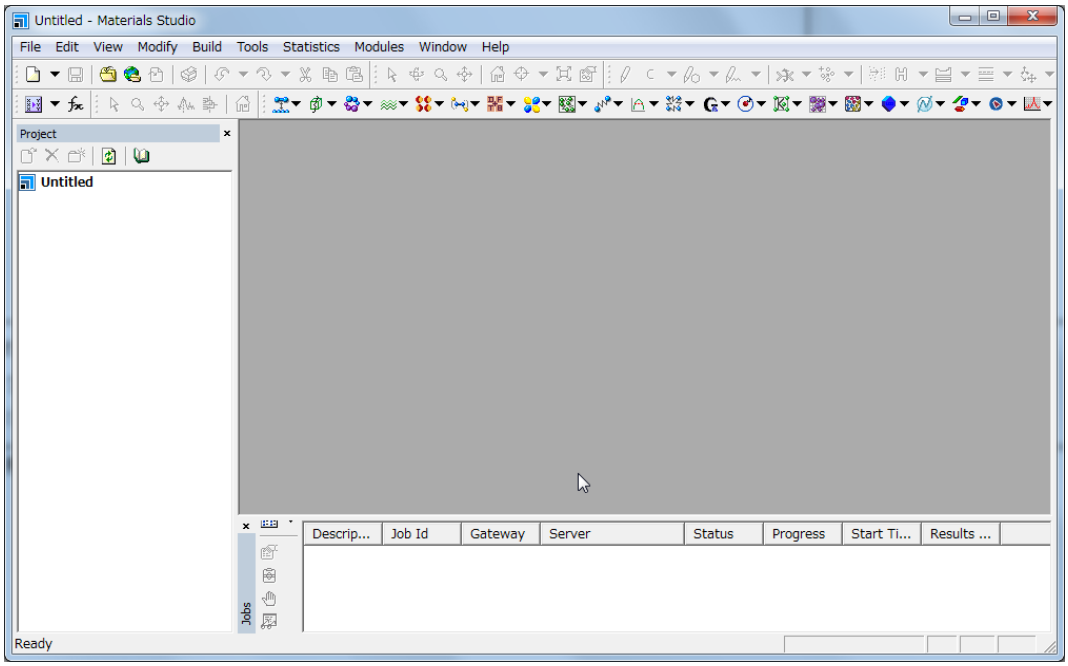


It will added to [Server Gateway].



6.11.3 How to execute

Select [BIOVIA] -Materials Studio from the start menu.

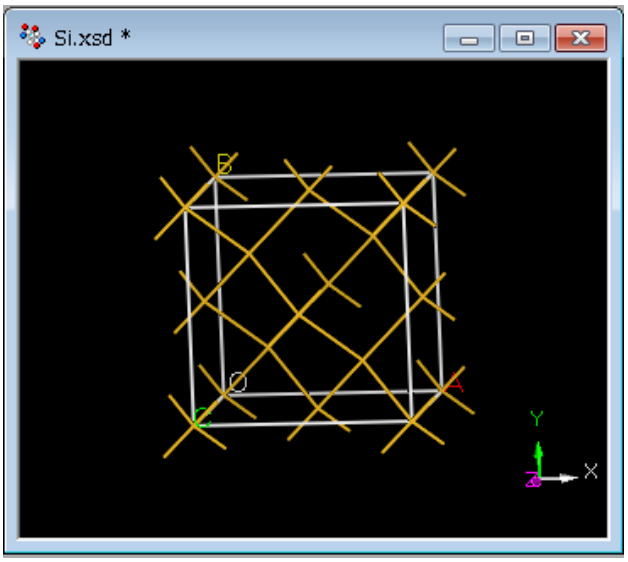


The followings are examples of CASTEP and DMol3.

6.11.4 How to execute CASTEP

This is an example of Si.

(1) Construction of a Model



(2) Setting Parameters

Select [Modules]-[CASTEP]-[Calculation] from the menu bar, and set parameters

(3) Submit Job

【 In the case of submitting job from Materials Studio 】

Select a machine you use on [Job Control] tab.

Gateway location	Machine
My Computer	Your PC where Materials Studio is installed
10.1.4.3_18889	Parallel Computing & Informatics Server(Materials Studio 2019)
10.1.4.3_18890	Parallel Computing & Informatics Server(Materials Studio 2020)

【 In the case of submitting job from command line 】

Select [Files]-[Save Files] in the [CASTEP Calculation] dialog to create input files. Since a job is submitted in the /work area, transfer them to the working directory in the super computing system.

- *.params
- *.cell

The *.cell is a hidden file.

Use text transfer mode to transfer these files, and don't include space and parentheses as a part of the filenames.

Copy a script file to the current directory.

```
$ cp /work/app/MaterialsStudio_cs/current/etc/CASTEP/bin/RunCASTEP.sh  ./
```

Make a script file

```
#!/bin/sh
#PBS -l select=1
#PBS -l castep= round up MPI tasks / 18
#PBS -q C_002
#PBS -N castep

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

./RunCASTEP.sh -np MPI tasks Si

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -l castep=2     * 34/18 =1.888··· -> 2
#PBS -q C_002
#PBS -N castep

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

./RunCASTEP.sh -np 34 Si

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

Option -l castep is mandatory.

The argument after MPI tasks is an input file name without extension.

. ex) Si.param/Si.cell -> Si

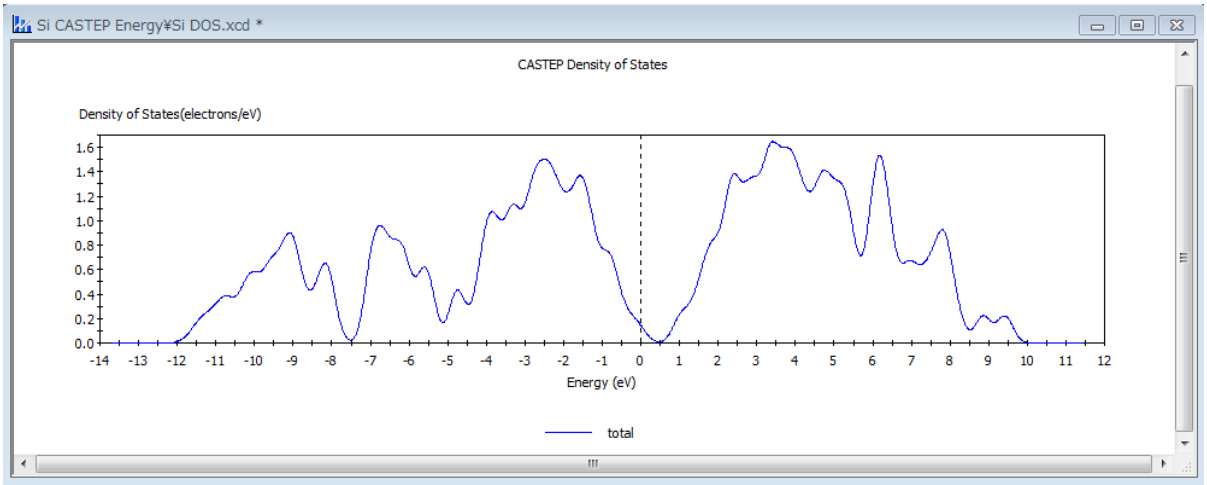
(4) Confirming Result

Copy output files to your PC.

Use text transfer mode if its format is text. Use binary transfer mode if not.

Select [Modules]-[CASTEP]-[Analysis] from the menu bar.

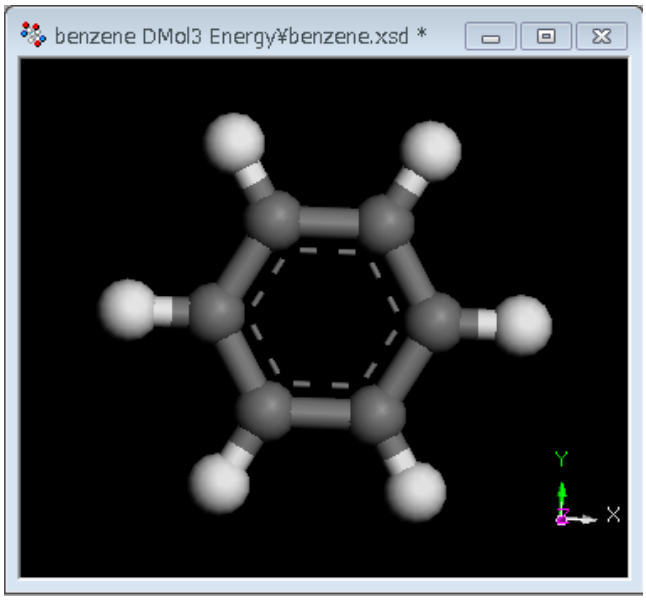
Density of States



6.11.5 How to execute DMol3

This is the example of benzene.

(1) Construction of a Model



(2) Setting Parameters

Select [Modules]-[DMol3]-[Calculation] from the menu bar, and set parameters.

(3) Submit Job

【 The case of submitting job from Materials Studio 】

Select a machine you use on [Job Control] tab.

Gateway location	Machine
My Computer	Your PC where Materials Studio is installed
10.1.4.3_18889	Parallel Computing & Informatics Server(Materials Studio 2019)
10.1.4.3_18890	Parallel Computing & Informatics Server(Materials Studio 2020)

【 The case of submitting job from command line 】

Select [Files]-[Save Files] in the [DMol3 Calculation] dialog to create input files. Since a job is submitted in the /work area, transfer them to the working directory in the super computing system.

- *.input
- *.car

The *.car is a hidden file.

Use text transfer mode to transfer these files, and don't include space and parentheses as a part of the filenames.

Copy a script file to the current directory.

```
$ cp /work/app/MaterialsStudio_cs/current/etc/DMol3/bin/RunDMol3.sh .
```

Make a script file

```
#!/bin/sh
#PBS -l select=1
#PBS -l dmol3= roundup MPI tasks / 18
#PBS -q C_002
#PBS -N dmol3

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

./RunDMol3.sh -np MPI tasks benzene

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) parallel computing and informatics server

```
#!/bin/sh
#PBS -l select=1
#PBS -l dmol3=2     * 32/18 =1.777··· -> 2
#PBS -q C_002
#PBS -N dmol3

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

./RunDMol3.sh -np 32 benzene

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

Option -l dmol3 is mandatory.

The argument after MPI tasks is an input file name without extension.

. ex) benzene.input/benzene.car -> benzene

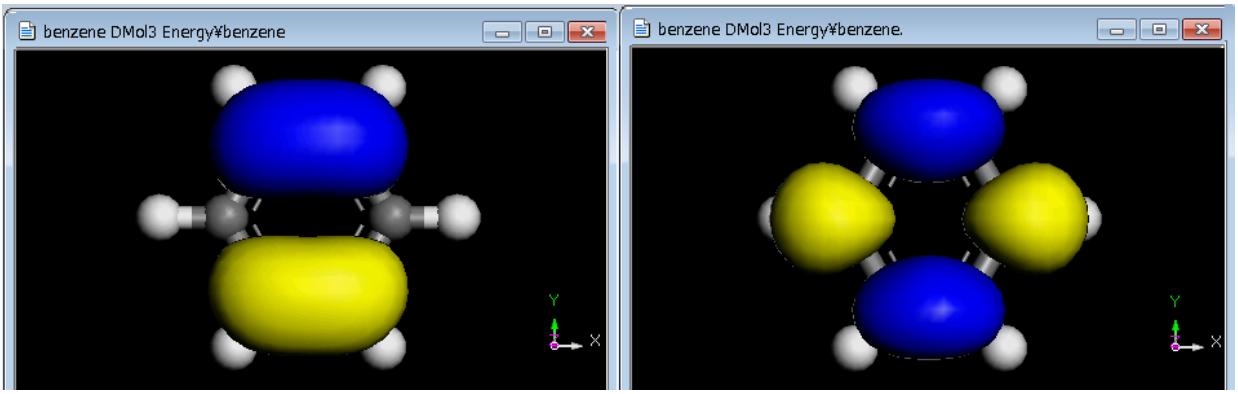
(4) Confirming Result

Copy output files to your PC.

Use text transfer mode if its format is text. Use binary transfer mode if not.

Select [Modules]-[DMol3]-[Analysis] from the menu bar.

Orbitals (HOMO：Right、LUMO：Left)



6.11.6 Confirming Job Status

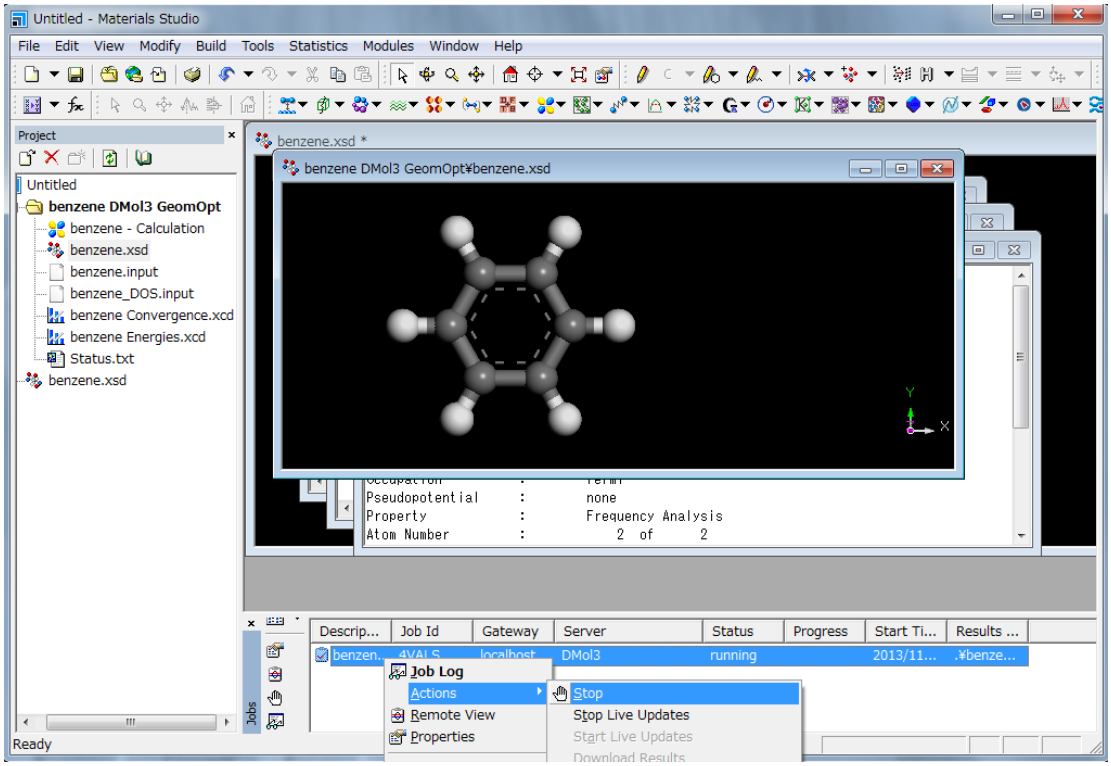
【 The case of submitting job from Materials Studio 】

See [Jobs] pane at the lower right of the screen on Materials Studio.

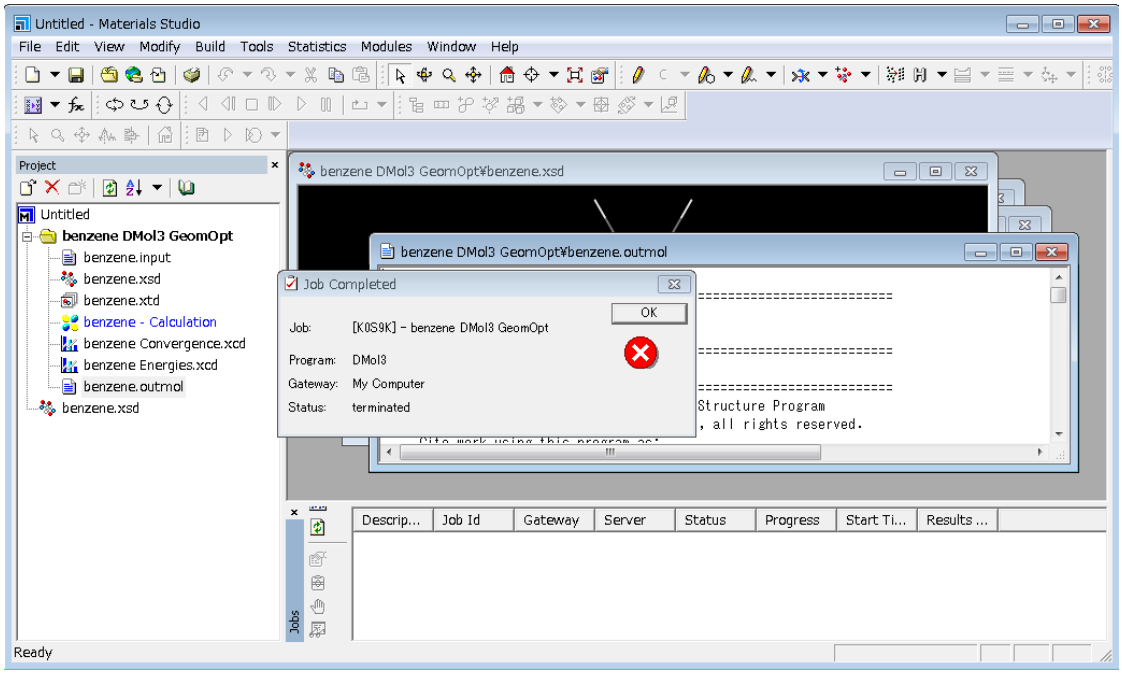
6.11.7 Canceling Job

【 The case of submitting job from Materials Studio 】

Right-click [Jobs] pane on Materials Studio, and select [Actions]-[Stop] from the pull-down menu.



[JobStatus] changes to "terminated" after a while, and your job stops with the following pop-up.



6.12 Wannier90

Available executables

Version	Path	Queue
1.2	/usr/local/app/Wannier90/wannier90-1.2	A_004
2.1.0	/usr/local/app/Wannier90/wannier90-2.1.0	A_004
3.1.0	/usr/local/app/Wannier90/current	A_004

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -q queue
#PBS -N jobname

module switch intel intel/19.1.0

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun [ -np MPI total tasks ][ -ppn MPI tasks per node ] -hostfile $PBS_NODEFILE
/usr/local/app/Wannier90/current/wannier90.x input file > output file 2> error
file

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

(Example) Accelerator Server

```
#!/bin/sh
#PBS -l select=1
#PBS -q A_004
#PBS -N wannier90

module switch intel intel/19.1.0

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

mpirun -np 36 -ppn 36 -hostfile $PBS_NODEFILE
/usr/local/app/Wannier90/current/wannier90.x wannier90 > wannier.out 2>
wannier.err

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm -rf $WORKDIR; fi
```

The argument after executable is an input file name without extension.

7

7 Usage of Machine Learning Environment

7.1 Machine Learning Environment

7.2 Chainer

7.3 Keras

7.4 Caffe

7.5 Jupyter Notebook

7.6 DIGITS

7.1 Machine Learning Environment

The following environment are available.

Use the queue CA_001g for the machine learning environment.

Please submit a job from /work_da area.

#	Application	image:tag
1	Chainer	conda3/mlenv:cuda10.1-007 (for CUDA10.1) conda3/mlenv:cuda9.1-006 (for CUDA9.1)
2	Keras	
3	Caffe	
4	Jupyter Notebook	
5	DIGITS	nvidia/digits:6.0

7.2 Chainer

The job script is as follows.

```
#!/bin/sh
#PBS -l select=1[:ncpus= number of CPU ][:ngpus=number of GPU ]
#PBS -q CA_001g
#PBS -N jobname
#PBS -v DOCKER_IMAGE= conda3/mlenv:tag

cd $PBS_O_WORKDIR

python input file > output file 2> error file
```

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q CA_001g
#PBS -N Chainer
#PBS -v DOCKER_IMAGE=conda3/mlenv:cuda10.1-007

cd $PBS_O_WORKDIR

python  train_cifar.py > train_cifar.out 2> train_cifar.err
```

The command for interactive mode is as follows.

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:tag
```

(Example)

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:cuda10.1-007
qsub: waiting for job 26269.gpu1 to start
qsub: job 26269.gpu1 ready

bash-4.2$ cd $PBS_O_WORKDIR
bash-4.2$ python  train_cifar.py > train_cifar.out 2> train_cifar.err
```

7.3 Keras

The job script is as follows.

```
#!/bin/sh
#PBS -l select=1[:ncpus= number of CPU ][:ngpus=number of GPU ]
#PBS -q CA_001g
#PBS -N jobname
#PBS -v DOCKER_IMAGE= conda3/mlenv:tag

cd $PBS_O_WORKDIR

python input file > output file 2> error file
```

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q CA_001g
#PBS -N Keras
#PBS -v DOCKER_IMAGE=conda3/mlenv:cuda10.1-007

cd $PBS_O_WORKDIR

python cifar100_resnet_multigpu.py > cifar100.out 2> cifar100.err
```

The command for interactive mode is as follows.

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:tag
```

(Example)

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:cuda10.1-007
qsub: waiting for job 26269.gpu1 to start
qsub: job 26269.gpu1 ready

bash-4.2$ cd $PBS_O_WORKDIR
bash-4.2$ python cifar100_resnet_multigpu.py > cifar100.out 2> cifar100.err
```

7.4 Caffe

The following commands are available.

For details, see the -help option.

```
caffe
classification
classify
compute_image_mean
convert_cifar_data
convert_imageset
convert_mnist_data
convert_mnist_siamese_da
ta
detect
device_query
draw_net
extract_features
finetune_net
net_speed_benchmark
test_net
train
train_net
upgrade_net_proto_binary
upgrade_net_proto_text
upgrade_solver_proto_tex
t
```

The job script is as follows.

```
#!/bin/sh
#PBS -l select=1[:ncpus= number of CPU ][:ngpus=number of GPU ]
#PBS -q CA_001g
#PBS -N jobname
#PBS -v DOCKER_IMAGE= conda3/mlenv:tag

cd $PBS_O_WORKDIR

command option
```

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q CA_001g
#PBS -N Caffe
#PBS -v DOCKER_IMAGE=conda3/mlenv:cuda10.1-007

cd $PBS_O_WORKDIR

caffe train --solver=examples/cifar10/cifar10_quick_solver.prototxt
```

The command for interactive mode is as follows.

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:tag
```

(Example)

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:cuda10.1-007
qsub: waiting for job 26269.gpu1 to start
qsub: job 26269.gpu1 ready

bash-4.2$ cd $PBS_O_WORKDIR
bash-4.2$ caffe train --solver=examples/cifar10/cifar10_quick_solver.prototxt
```

7.5 Jupyter Notebook

Use interactive mode for Jupyter Notebook.

The command is as follows.

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:tag
$ jupyter notebook  --ip=* --no-browser
```

(Example)

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=conda3/mlenv:cuda10.1-007
qsub: waiting for job 26322.gpu1 to start
qsub: job 26322.gpu1 ready

Access Port:
       <proto>://10.1.4.28:6037/ -> container port 8888
bash-4.2$ jupyter notebook  --ip=* --no-browser
...

    To access the notebook, open this file in a browser:
 file:///home/hitacse/.local/share/jupyter/runtime/nbserver-12-open.html
    Or copy and paste one of these URLs:

http://cgpu28-26322-cgpu28:8888/?token=3a4f5d232de317ef49f51630ba4038e47bdd51
f7d191ee2c
```

Check the port corresponding to "container port 8888".

In the example above, it is 6037. (hereinafter referred to as “port A”)

And check the login token. In the example above, it is

3a4f5d232de317ef49f51630ba4038e47bdd51f7d191ee2c.

Port forward any port of your PC to port 22 of gpu2 using new terminal. (hereinafter referred to as “port B”)

```
$ ssh -L port B:gpu2:22 username@cms-ssh.sc.imr.tohoku.ac.jp
```

(Example)

```
$ ssh -L 8022:gpu2:22 userA@cms-ssh.sc.imr.tohoku.ac.jp
```

Port forward any port of your PC to port of Jupyter Notebook using new terminal again. (hereinafter referred to as “port C”)

```
$ ssh -L port C:10.1.4.28:port A -p port B localhost
```

(Example)

```
$ ssh -L 15000:10.1.4.28:6037 -p 8022 localhost
```

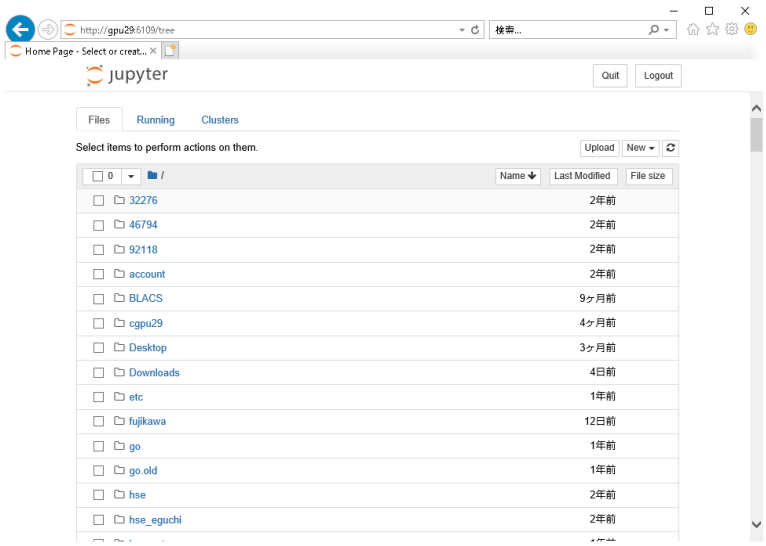
Access the following URL from your PC.

```
http://localhost:port C/?token=login token
```

(Example)

```
http://localhost:15000/?token=3a4f5d232de317ef49f51630ba4038e47bdd51f7d191ee2c
```

The Jupyter Notebook screen will appear.



7.6 DIGITS

Use interactive mode for DIGITS.

```
$ qsub -I -q CA_001g -v DOCKER_IMAGE=nvidia/digits:6.0
qsub: waiting for job 26322.gpu1 to start
qsub: job 26322.gpu1 ready

Access Port:
       <proto>://10.1.4.28:6037/ -> container port 5000
       <proto>://10.1.4.28:6161/ -> container port 6006
bash-4.2$
```

Check the port corresponding to "container port 5000".
In the example above, it is 6037. (hereinafter referred to as “port A”)

Port forward any port of your PC to port 22 of gpu2 using new terminal. (hereinafter referred to as “port B”)

```
$ ssh -L port B:gpu2:22 username@cms-ssh.sc.imr.tohoku.ac.jp
```

(Example)

```
$ ssh -L 8022:gpu2:22 userA@cms-ssh.sc.imr.tohoku.ac.jp
```

Port forward any port of your PC to port of DIGITS using new terminal again. (hereinafter referred to as “port C”)

```
$ ssh -L port C 10.1.4.28:port A -p port B localhost
```

(Example)

```
$ ssh -L 15000:10.1.4.28:6037 -p 8022 localhost
```

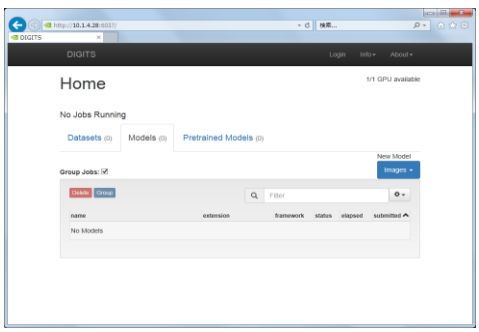
Access the following URL from your PC.

```
http://localhost:port C
```

(Example)

```
http://localhost:15000
```

The DIGITS screen will appear.



8 Usage of Python

8.1 About using Python

8.2 Install pyenv

8.3 Setting environment variables

8.4 Operational check

8.5 Basic usage

8.6 Execution method

8.1 About using Python

In this system, pyenv can be installed in the user's environment by executing the following script. Python version control is possible with pyenv. Please see below for details.

8.2 Install pyenv

Execute the following command to install pyenv on super and gpu.

```
$ bash /work/app/pyenv/pyenv-setup.bash
```

8.3 Setting environment variables

After executing the installation script, a file called bash_env will be generated in the current directory. To load pyenv by default, copy the contents to ~/.bash_profile with the following command.

```
$ cat bash_env >> ~/.bash_profile
```

8.4 Operational check

After installing pyenv, log out from the front end node once, log in again, and then execute the following command.

```
$ pyenv --version
pyenv 1.2.8-5-gec9fb549     ←The version may be different.
```

8.5 Basic usage

・Install Python using pyenv

```
$ pyenv install --list     ←View available Python versions
....
$ pyenv install 3.7.8      ←Install Python 3.7.8
```

・Switch python versions

```
$ pyenv versions           ←Check the installed version
* system (set by /home/userA/.pyenv/version)
  3.7.8
$ pyenv global 3.7.8       ←Switch to Python 3.7.8
$ python --version
```

You can also add packages using pip.
For more detailed usage, please check the pyenv documentation etc.

Simple Python version management
https://github.com/pyenv/pyenv

8.6 Execution method

Execute Python programs with high load on compute nodes instead of front end nodes.

(Example) Interactive mode

```
userA@gpu2:~> qsub -I -q CA_001
qsub: waiting for job 70568.gpu1 to start
qsub: job 70568.gpu1 ready

-bash-4.2$ python
Python 3.7.8 (default, Mar 25 2021, 09:54:46)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```

It is also possible to submit it as a job and execute it.

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q CA_001
#PBS -N sample
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

python program.py

cd; if cp -raf $WORKDIR/$DIRNAME $PBS_O_WORKDIR/.. ; then rm
-rf $WORKDIR; fi
```