Institute for Materials Research, Tohoku University Large-Scale Parallel Computing Server Manual

May 17th, 2021

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

Contents

1

1 Outline of supercomputer

1.1 Configuration and specification

The specification of the supercomputer.

Server name	Supercomputer	Front end node
Model name	Cray XC50-LC	Cray XC50-LC
Number of servers	293 servers + 3 servers(spare)	2 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 : 2.1 GHz ・Number of CPU cores : 18 Core ・Number of socket : 1 Socket/Server
Total amount of main memories	768 GiB/Server	768 GiB/Server

1.2 Configuration of nodes

Nodes of the supercomputer are as follows.

Node type	Usage	Number of nodes	Installation site
Front end nodes	For submitting jobs	2 nodes	Center for computational materials science Room101
Computational node	For calculating	293 nodes	Center for computational materials science Room101
Computational node (Spare)	For changing node when failure is generated in a computational node	3 nodes	Center for computational materials science Room101

2

2 Login method

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 super that is front end node for the supercomputer from the ssh relay server.

```
$ ssh super
```

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]
```

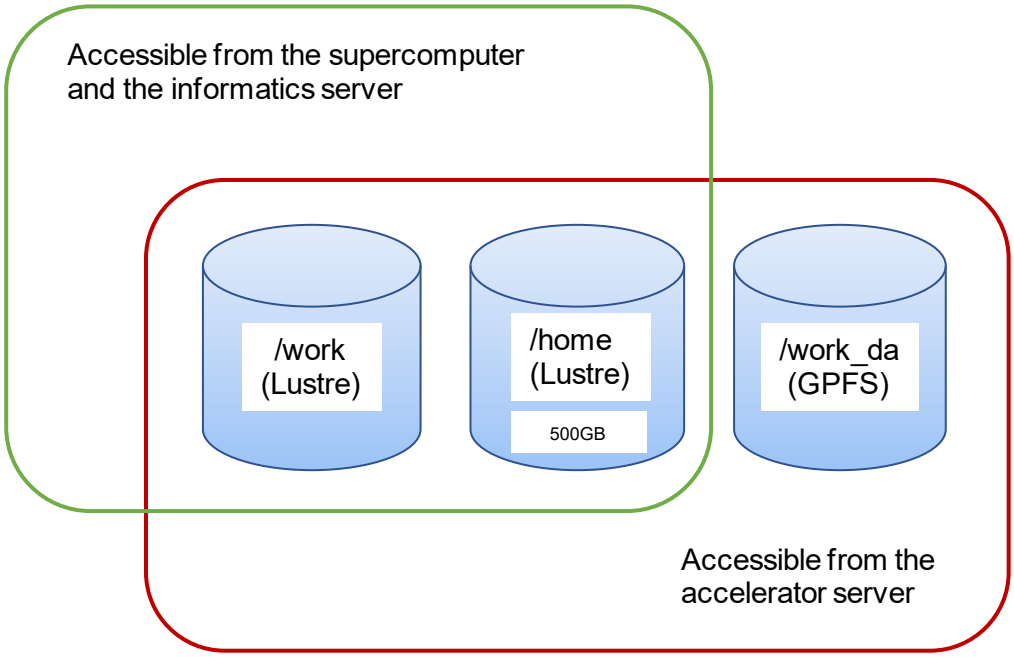
3

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	work_da	none	This is GPFS area.

server			To use the queue 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

4.1.1 Job submit command (qsub command)

Submit command places a job on the job queue of the supercomputer.

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] [-l license name=number of use licenses] [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 236characters. 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: DP_002, number of nodes: 2, limit of walltime: 10 minutes, script file:hello.sh

```
$ qsub -q DP_002 -l select=2 -l walltime=00:10:00 hello.sh
```

Description in the execution script.

```
#!/bin/sh
#PBS -q DP_002
#PBS -l select=2
#PBS -l walltime=00:10:00
 :
 :
 :
```

・queue: P_016, 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 P_016 -M userA@test.com -m be hello.sh
```

Description in the execution script.

```
#!/bin/sh
#PBS -q P_016
#PBS -M userA@test.com
#PBS -m be
 :
 :
 :
```

・queue: DP_002, application: QuantumATK, script file: atk.sh

```
$ qsub -q DP_002 -l atk=1 -l atkdp=35 atk.sh
```

Description in the execution script.

```
#!/bin/sh
#PBS -q DP_002
#PBS -l atk=1 -l atkdp=35
 :
 :
 :
```

4.1.2 Program execution command (aprun command)

To execute a program, use the aprun command.

* If you do not use aprun command, it is not executed on the calculation node. It may be canceled by the administrator because it has effects on other users.

(1) Usage

```
$ aprun  [-n MPI total tasks ] [-d OpenMP threads][-N MPI tasks per node] [-S MPI tasks per CPU socket] [-j 0|1|N] [--cc placement method] program
```

(2) List of options

Option	Description
-n *MPI total tasks*	Set the total number of MPI tasks.
-d *OpenMP threads*	Set the number of OpenMP threads. (Set OMP_NUM_THREADS together.)
-N *MPI tasks per node*	Set the number of MPI tasks per node.
-S *MPI tasks per CPU socket*	Set the number of MPI tasks per CPU socket.
-j 0|1|N	Set the number of thread per CPU core. 0: Use HyperThreading (default) 1: No use HyperThreading N: Use HyperThreading, place N threads per core.
--cc *placement method*	Set the tasks / threads placement method. depth: Bind the threads so that the process is close to the allocated CPU core (useful when executing OpenMP, MPI + OpenMP programs)

*To improve job performance, specify values so that:

[*MPI total tasks*(the value of "-n")] = [*Number of nodes*(the value of "#PBS -l select=")] × [*MPI tasks per node*(the value of "-N")]

4.1.3 Format of an execution script

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

aprun 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 P_016
#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

aprun ./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 using ESM mode of Cray XC

```
#!/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

aprun [ -n MPI total tasks ] [ -N MPI tasks per node ] 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 1 node and 2 MPI processes.

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N mpi1
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

aprun -n 2 -N 2 ./a.out > result.out 2> result.err

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

・Example To run a program on 2 node and 2 MPI processes.

```
#!/bin/sh
#PBS -l select=2
#PBS -q P_016
#PBS -N mpi2
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

aprun -n 2 -N 1 ./a.out > result.out 2> result.err

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

4.1.4 Interactive mode

Submit a job with interactive mode.

Add option -I (uppercase i) to the qsub command and specify IP_001 for the queue.

Be sure to use the aprun command to execute a program.

(1) Usage

```
$ qsub -I -q IP_001
```

(2) Example

```
$ qsub -I -q IP_001
qsub: waiting for job 220331.sdb to start
qsub: job 220331.sdb ready

Directory: /home/userA
Mon Sep 23 01:03:04 JST 2019

userA@mom1:~> cd $PBS_O_WORKDIR
userA@mom1: /work/userA/testdir> aprun -n 32 -j 1 ./a.out
```

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@super2:~> 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 supercomputer.

(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@super2:~> qstat -a
                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
3390.sdb        userA    P_016    abinit      193347   4 144 2760gb 72:00 R 47:28
3401.sdb        userA    P_016    prog9_1     121974   4 144 2760gb 72:00 R 47:26

userA@super2:~> qstat -p
Job id           Name            User              % done  S Queue
---------------  ----------------  ----------------  -------- - -----
3390.sdb         abinit          userA               2  R P_016
3401.sdb         prog9_1         userA               0  R P_016

userA@super2:~> qstat -t
Job id           Name            User             Time Use S Queue
---------------  ----------------  ----------------  -------- - -----
3390.sdb         abinit          userA          00:00:01 R P_016
3401.sdb         prog9_1         userA          00:00:01 R P_016

userA@super2:~ > qstat -x
Job id           Name            User             Time Use S Queue
---------------  ----------------  ----------------  -------- - -----
2235.sdb         prog9_2         userA          00:00:03 F P_016
2236.sdb         vasp4           userA          00:00:01 F P_016
2237.sdb         prog9_1         userA          00:00:01 F P_016
        The rest is omitted

userA@super2:~> qstat -f 3390.sdb
Job Id: 3390.sdb
    Job_Name = abinit
    Job_Owner = userA@nid00204
    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 supercomputer.

(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@super2:~> 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
DP_002       0     0  yes  yes     0     0     0     0     0     0 Exec
P_016        0     1  yes  yes     0     1     0     0     0     0 Exec
P_032        0     0  yes  yes     0     0     0     0     0     0 Exec
P_064        0     0  yes  yes     0     0     0     0     0     0 Exec
LP_032       0     0  yes  yes     0     0     0     0     0     0 Exec
LP_064       0     0  yes  yes     0     0     0     0     0     0 Exec
       The rest is omitted

userA@super2:~> 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 supercomputer.

(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@super2:~> qstat -B
Server            Max   Tot   Que   Run   Hld   Wat   Trn   Ext Status
---------------- ----- ----- ----- ----- ----- ----- ----- -----
sdb                0  1155     0     1     0     0     0     0 Active
userA@super2:~> 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 supercomputer.

(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@super2:~/work/20180712_sample> statj
                                                    Req'd   Req'd   Elap
Job ID     Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
---------- -------- -------- ---------- ------ --- --- ------ ----- - -----
3413.sdb   userA    P_016    abinit     3710   3   216 2304gb 72:00 R 00:00
3414.sdb   userA    DP_002   STDIN      13588  1    72  768gb 00:10 R 00:00
userA@super2:~/work/20180712_sample> qdel 3414.sdb
userA@super2:~/work/20180712_sample> statj
                                                    Req'd   Req'd   Elap
Job ID     Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
---------- -------- -------- ---------- ------ --- --- ------ ----- - -----
3413.sdb   userA    P_016    abinit     3710   3   216 2304gb 72:00 R 00:00
userA@super2:~/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@super2:~ >  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 Specify the parameters affecting performance of job

(1) Description

Specifying number of processes per core by apron command, performance of the job can be improved, because Hyper-Threading is enabled on the supercomputer.

(2) Usage

Specify 1 thread per physical core.

```
aprun -j 1 program
```

(3) Example

```
#!/bin/bash
#PBS -j oe
#PBS -l select=1
#PBS -q P_016

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

aprun -n 36 -N 36 -j 1 ./xhpl_skl_diag_cray_opt > 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 supercomputer are as follows.

Queue name	Limit of number of exclusive nodes (default)	Memory Limit [GiB]	Walltime limit [hours] (default)	Limit of concurrent execution	Limit of number of parallel execution*	Notes
IP_001	1(1)	768	24(1)	No limit	72	For interactive
DP_002	2(1)	1,536	0.5(0.5)	No limit	144	For debugging
P_016	16(1)	12,288	72(24)	No limit	1152	1 to 16 nodes used
P_032	32(32)	24,576	72(24)	4	2304	17 to 32 nodes used
P_064	64(64)	49,152	72(24)	2	4608	33 to 64 nodes used

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	Limit of number of parallel execution*	Notes
LP_064	64(64)	49,152	168(96)	2	4608	Executed during operation
MP_096	96(96)	73,728	72(24)	1	6912	
SP_064	64(64)	49,152	336(336)	1	4608	executed after maintenance operation only
SP_128	128(128)	98,304	168(168)	1	9216	
SP_293	293(293)	225,024	24(24)	1	21096	

5 Compilers and Libraries

5.1 Compiler

Following compilers are available on the supercomputer.

For more details, see each manual.

Compiler	Version	Notes
Cray Compiler Fortran/C/C++	8.7.10 8.6.5	default: 8.6.5
Intel Compiler Fortran/C/C++	19.1.3.304 19.1.0.166 19.0.2.187 18.0.2.199 17.0.4.196	default: 18.0.2.199
PGI Compiler Fortran/C/C++	19.1-0 18.5-0	default: 19.1-0

5.1.1 Programming environment

（1）Command

The command names of the Fortran, C, C ++ compiler are unified to ftn, cc, and CC.

The compiler called by the command will switch automatically by switching the programming environment.

Compiler	Command	Cray Compiler	Intel Compiler	PGI Compiler	gnu Compiler
Fortran	ftn	crayftn	ifort	pgf90	gfortran
C	cc	craycc	icc	pgcc	gcc
C++	CC	crayCC	icpc	pgc++	g++

Use ftn, cc, CC command in the case compiling MPI programs.

MPI libraries are linked automatically.

（2）Switching programming environment

The module files is as follows.

Cray compiler	Intel compiler	PGI compiler	gnu compiler
PrgEnv-cray	PrgEnv-intel	PrgEnv-pgi	PrgEnv-gnu

Switch the programming environment with the module switch command.

PrgEnv-cray is default.

To switch to the Intel compiler, execute as follows.

```
$ module switch PrgEnv-cray PrgEnv-intel
```

(3) Common options

The following options are common to all programming environments.

Option	Description
-craype-verbose	Display set options.
-static	Set static linking.
-dynamic	Set dynamic linking.
-shared	Make shared library.
-help	Display available options in the loaded programming environment.

5.1.2 Cray Compiler

(1) Setting of program environment

Cray compiler is the default one on super.

If you want to change the compiling environment to other, execute a following command.

・Example change from the environment of Intel compiler.

```
$ module switch PrgEnv-intel PrgEnv-cray
```

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

・Example change the compiler version

```
$ module avail cce
cce/8.6.5(default)
cce/8.7.10

$ module switch cce cce/8.7.10
```

(2) How to compile

Options

・Optimization Options

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.
-h autothread	Enables auto threading (default: disabled)
-h omp/noomp	Enable / disables OpenMP directives. (default: enabled)
-h thread[0-3]	Specify optimization level of OpenMP (3:Full optimization, default is 2)
-O [0-3]	Specify automatic threading level. (3:Highest level, default is 2)
-h ipa[0-5]	Specify the level of interprocedural optimization.

・Options for Fortran

Option	Description
-e0	Initialize undefined local stack variables to 0.
-ev	Assign variables to stack (There are exceptions.)
-ez	Initialize the memory allocated by allocate() statements to zero.
-f [free|fixed]	Set the format of source files.

・Options for C

Option	Description
-h c99	Compile with C99 standard manner.
-h noc99	Compile without C99 standard manner.
-h zero	Initialize undefined local stack variables to 0.

・Debugging options

Option	Description
-g	Output debugging information.
-G [0-2]	Manage debugging information. (0: Most detailed. -G0=-g)
-K trap=…	Trap exception handling. [fp | divz | denorm | …]

(3) Example

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

```
$ ftn -f fixed -o hello.out hello.f
```

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

```
$ ftn -f free -o hello.out hello.f90
```

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

```
$ ftn -h autothread -f fixed -o hello.out hello.f
```

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

```
$ ftn -h autothread -f free -o hello.out hello.f90
```

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

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

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

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

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

```
$ CC hello.cpp -o hello.out
```

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

```
$ CC -h autothread hello.cpp -o hello.out
```

5.1.3 Intel Compiler

（1）Setting of program environment

Change the program environment by the following line, because Cray compiler is default on super.

・Example　change to the environment of Intel compiler.

```
$ module switch PrgEnv-cray PrgEnv-intel
```

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

・Example　change the compiler version

```
$ module avail intel
intel/17.0.4.196
intel/18.0.2.199(default)
intel/19.0.2.187
intel/19.1.0.166
intel/19.1.3.304

$ module switch intel intel/19.0.2.187
```

（2）How to compile

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.

・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.

```
$ ftn -fixed -o hello.out hello.f
```

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

```
$ ftn -free -o hello.out hello.f90
```

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

```
$ ftn -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.

```
$ ftn -free -parallel -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' auto-threaded from C source file 'hello.c'.

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

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

```
$ CC -o hello.out hello.cpp
```

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

```
$ CC -parallel -o hello.out hello.cpp
```

5.1.4 PGI Compiler

(1) Setting of program environment

Change the environment by the following line, because Cray compiler is default on super.

・Example change from environment of Cray compiler

```
$ module switch PrgEnv-cray PrgEnv-pgi
```

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

・Example change the compiler version

```
$ module avail pgi
pgi/18.5
pgi/19.1(default)

$ module switch pgi pgi/18.5
```

(2) How to compile

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 Specify program format of free or fixed.

・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.

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

・make the execution module 'hello.out' which is autothreaded from the Fortran source file 'hello.f' in fixed form.

```
$ ftn -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.

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

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

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

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

```
$ ftn -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.

```
$ ftn -mp -Mfree -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' which is auto-threaded from C source file 'hello.c'.

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

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

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

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

```
$ CC -o hello.out hello.cpp
```

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

```
$ CC -Mconcur -o hello.out hello.cpp
```

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

```
$ CC -mp -o hello.out hello.cpp
```

5.1.5 GNU Compiler

(1) Setting of program environment

Change the program environment by the following line, because Cray compiler is default on super.

・Example　change to the environment of Intel compiler.

```
$ module switch PrgEnv-cray PrgEnv-gnu
```

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

・Example　change the compiler version

```
$ module avail gcc
gcc/4.9.3
gcc/5.3.0
gcc/6.1.0
gcc/7.3.0(default)
gcc/8.3.0

$ module switch gcc gcc/8.3.0
```

(2) How to compile

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.

```
$ ftn -fixed -o hello.out hello.f
```

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

```
$ ftn -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'.

```
$ CC -o hello.out hello.cpp
```

5.2 Libraries

Following libraries are available for compiling and linking.

For more details, see each manual.

Library	Version	Compiler	Notes
CSML(Cray Scientific and Math Libraries)	19.05.5 18.03.1	all compilers	
Intel MKL (Intel Math Kernel Library)	19.1.3.304 19.1.0.166 19.0.2.187 18.0.2.199 17.0.4.196	Intel compiler	
Third Party Products	-	all compilers	

5.2.1 CSML(Cray Scientific and Math Libraries)

CSML (Cray Scientific and Math Libraries) is a set of libraries provide by Cray.

Library	Description	Included libraries	Module	Option
Cray LibSci	Scientific library optimized for XC50 system Default	BLAS, LAPACK, BLACS, ScaLAPCK, IRT, etc	cray-libsci	
Cray PETSc (Portable, Extensible Toolkit for Scientific Computation)	Linear / nonlinear equation parallel solver library	MUMPS, SuperLU, SuperLU_dist, ParMETIS, HYPRE , etc	cray-petsc	
Cray Trilinos Packages	Object-oriented interface of scientific library Load cray-petsc in advance.	PETSc, Metis/ParMetis, SuperLU, Aztec, BLAS, LAPACK	cray-trilinos	
TPSL (Third Party Scientific Libraries)	Math library available with PETSc/Trilinos	MUMPS, Super_LU, Super_LU_dist, ParMetis, Hypre, Sumdials, Scotch, etc	cray-tpsl	

.

FFTW3.3 Library	FFTW version 3.3	FFTW3.3	cray-fftw	-lfftw3 (MPI parallel) -lfftw3_mpi (thread parallel) -lfftw3_threads
FFTW2.1 Library	FFTW version2.1 Library name depends on single precision and double precision.	FFTW2.1	fftw	(single precision) -lsrfftw_mpi -lsfftw_mpi -lsrfftw -lsfftw (double precision) -ldrfftw_mpi -ldfftw_mpi -ldrfftw -ldfftw

（1）Setting of program environment

Change the program environment by the following line, because Cray compiler is default on super.

・Example change to environment of Intel compiler.

```
$ module switch PrgEnv-cray PrgEnv-intel
```

（2）Example

ex1: Using FFTW 3.3 Library from MPI parallel program with Cray compiler

```
$ module load cray-fftw
$ cc main.c -L${FFTW_DIR} -lfftw3_mpi -lfftw3
```

ex2: Using FFTW 3.3 Library from MPI and thread parallel program with Intel compiler

```
$ module switch PrgEnv-cray PrgEnv-intel
$ module load cray-fftw
$   cc   -qopenmp   main.c   -L${FFTW_DIR}   -lfftw3_mpi
-lfftw3_threads -lfftw3
```

ex3: Using FFTW 2.1 Library with single precision

```
$ module load fftw
$ cc main.c -lsrfftw_mpi -lsfftw_mpi -lsrfftw -lsfftw
```

ex4: Using FFTW 2.1 Library with double precision

```
$ module load fftw
$ cc main.c -ldrfftw_mpi -ldfftw_mpi -ldrfftw -ldfftw
```

5.2.2 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 of program environment

In order to use Intel MKL, it is necessary to load the Intel program environment (PrgEnv-intel) and unload the Cray-libsci to avoid contention.

・Example change from environment of Intel compiler.

```
$ module switch PrgEnv-cray PrgEnv-intel
$ module unload cray-libsci
```

（2）Example

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

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

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

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

5.2.3 Third Party Products

The following libraries are available.

Library	Module	Note
NetCDF(Unidata's Network Common Data Format) Library	cray-netcdf cray-parallel-netcdf	serial version parallel version
HDF5(Hierarchical Data Format 5) Libraries & Utilities	cray-hdf5 cray-hdf5-parallel	serial version parallel version

6

6 Usage of applications

6.1 Applications

The following applications are available on the supercomputer.

#	Application	Version	Execution type
1	Gaussian 16	Rev B.01 Rev C.01	SMP
2	ADF	2017.113 2018.105 2019.102 2019.304 2020.101	MPI
3	QuantumATK	2019.03 2019.12 2020.09	MPI
4	CRYSTAL	17	MPI SMP
5	VASP	4.6.38 5.4.4 6.1.0 6.1.1 6.1.2 6.2.0	MPI
6	WIEN2k	17.1 18.2 19.1 19.2 21.1	SMP
7	SIESTA	4.0 4.1.5	MPI

8	ABINIT	8.8.2 8.10.2 8.10.3 9.2.2	MPI
9	CPMD	4.1 4.3	MPI
10	QUANTUM ESPRESSO	6.2.1 6.3 6.4.1 6.5 6.6 6.7	MPI
11	LAMMPS	31 Mar 17 22 Aug 18 12 Dec 18 5 Jun 19 7 Aug 19 3 Mar 20 29 Oct 20	MPI
12	OpenMX	3.8.5 3.9.1 3.9.2	MPI
13	SMASH	2.2.0	MPI
14	TOMBO	2	MPI
15	RSDFT	1.3.0	MPI
16	HPhi	3.1.2	MPI × SMP
17	mVMC	1.0.3	MPI
18	CP2K	7.0 8.1.0	MPI
19	Elk	6.3.2 6.8.4	MPI × SMP
20	ALAMODE	1.1.0	SMP

21	SALMON	1.2.1 2.0.0	MPI
22	OCTOPUS	9.1	MPI
23	Wannier90	1.2 2.1.0	serial
		3.1.0	MPI

6.2 Gaussian16

Available executables

Version	Settings
B.01	source /work/app/Gaussian/g16.profile
C.01	source /work/app/Gaussian/C.01/g16.profile

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 /work/app/Gaussian/example.com
-rw-r--r-- 1 root root 420 Jul 12 16:33 /work/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.
(Example) %CPU=0-35

・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.
Set '-d *parallel number*' to the same value as 'CPU' in the input file.

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

source /work/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

aprun -j 1 -d parallel number g16 input file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N g16

source /work/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

aprun -j 1 -d 36 g16 test0000.com 2> g16.err

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

6.3 ADF

Available executables

Version	Settings
2017.113	module load adf/2017.113
2018.105	module load adf/2018.105
2019.102	module load adf/2019.102
2019.304	module load adf/2019.304
2020.101	module load adf

Create a script file in advance.

・execute from input file. (Version：2017.113 - 2019.304)

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

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

adf -n MPI total tasks  < input file > output file 2> error file

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

(Example) 2019.304

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N adf

module load adf/2019.304
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

adf -n 36 < in > adf.out 2> adf.err

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

・execute from input file. (Version：2020.101)

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

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

ams -n MPI total tasks  < input file > output file 2> error file

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

(Example) 2020.101

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N adf

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

ams -n 36 < in > adf.out 2> adf.err

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

・execute from *.run-script.

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

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

export NSCM=MPI total tasks
./*.run-script > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N adf

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

export NSCM=36
./H2O_HF_freq.run > adf.out 2> adf.err

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

6.4 QuantumATK

Available executables

Version	Path
2019.03	/work/app/QuantumATK/QuantumATK-P-2019.03/bin/atkpython
2019.12	/work/app/QuantumATK/QuantumATK-Q-2019.12/bin/atkpython
2020.09	/work/app/QuantumATK/current/bin/atkpython

Create a script file in advance.

```
#!/bin/sh
#PBS -l select=nodes
#PBS -l atk=1 -l atkdp= MPI tasks -1
#PBS -q queue
#PBS -N jobname

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

aprun [ -n MPI total tasks ] [ -N MPI tasks per node ]  hostname  | grep -v ^Applicati
> hostfile
ccmrun /work/app/QuantumATK/current/libexec/mpiexec.hydra -n MPI tasks
-f ./hostfile -genv I_MPI_FABRICS=shm:tcp
/work/app/QuantumATK/current/bin/atkpython input file > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -l atk=1 -l atkdp=35
#PBS -q P_016
#PBS -N atk

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

aprun -n 36  hostname  | grep -v ^Applicati > hostfile
ccmrun /work/app/QuantumATK/current/libexec/mpiexec.hydra -n 36 -f ./hostfile
-genv I_MPI_FABRICS=shm:tcp /work/app/QuantumATK/current/bin/atkpython
input.py > atk.out 2> atk.err

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

Option -l atk and -l atkdp is mandatory.

6.5 CRYSTAL

Create a script file in advance.

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

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

source /work/app/Crystal/current/utils17/cry17.bashrc
runmpi17 MPI total tasks input file > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N crystal

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

source /work/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
```

6.6 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

Version	Path
VASP4.6.36	/work/app/VASP4/current/vasp
VASP4.6.36 Gamma point ver.	/work/app/VASP4/vasp.4.6_gamma/vasp
VASP5.4.4 Standard ver.	/work/app/VASP5/current/bin/vasp_std
VASP5.4.4 Gamma point ver.	/work/app/VASP5/current/bin/vasp_gam
VASP5.4.4 non-collinear ver.	/work/app/VASP5/current/bin/vasp_ncl
VASP6.1.0 Standard ver.	/work/app/VASP6/vasp.6.1.0/bin/vasp_std
VASP6.1.0 Gamma point ver.	/work/app/VASP6/vasp.6.1.0/bin/vasp_gam
VASP6.1.0 non-collinear ver.	/work/app/VASP6/vasp.6.1.0/bin/vasp_ncl
VASP6.1.1 Standard ver.	/work/app/VASP6/current/bin/vasp_std
VASP6.1.1 Gamma point ver.	/work/app/VASP6/current/bin/vasp_gam
VASP6.1.1 non-collinear ver.	/work/app/VASP6/current/bin/vasp_ncl
VASP6.1.2 Standard ver.	/work/app/VASP6/vasp.6.1.2/bin/vasp_std
VASP6.1.2 Gamma point ver.	/work/app/VASP6/vasp.6.1.2/bin/vasp_gam
VASP6.1.2 non-collinear ver.	/work/app/VASP6/vasp.6.1.2/bin/vasp_ncl
VASP6.2.0 Standard ver.	/work/app/VASP6/vasp.6.2.0/bin/vasp_std
VASP6.2.0 Gamma point ver.	/work/app/VASP6/vasp.6.2.0/bin/vasp_gam
VASP6.2.0 non-collinear ver.	/work/app/VASP6/vasp.6.2.0/bin/vasp_ncl

VASP 5.4.4 , VASP 6.1.1 and VASP 6.1.2 executables linked with Wannier90 are also available.

Version	Path
VASP5.4.4 (Wannier90) Standard ver.	/work/app/VASP5/vasp.5.4.4_wannier90/bin/vasp_std /work/app/VASP5/vasp.5.4.4_wannier90v2.1/bin/vasp_std
VASP5.4.4 (Wannier90) Gamma point ver.	/work/app/VASP5/vasp.5.4.4_wannier90/bin/vasp_gam /work/app/VASP5/vasp.5.4.4_wannier90v2.1/bin/vasp_gam
VASP5.4.4 (Wannier90) non-collinear ver.	/work/app/VASP5/vasp.5.4.4_wannier90/bin/vasp_ncl /work/app/VASP5/vasp.5.4.4_wannier90v2.1/bin/vasp_ncl
VASP6.1.1 (Wannier90) Standard ver.	/work/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_std /work/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_std
VASP6.1.1 (Wannier90)	/work/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_gam

Gamma point ver.	/work/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_gam
VASP6.1.1 (Wannier90) non-collinear ver.	/work/app/VASP6/vasp.6.1.1-wannier90v1.2/bin/vasp_ncl /work/app/VASP6/vasp.6.1.1-wannier90v2.1.0/bin/vasp_ncl
VASP6.1.2 (Wannier90) Standard ver.	/work/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_std /work/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_std
VASP6.1.2 (Wannier90) Gamma point ver.	/work/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_gam /work/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_gam
VASP6.1.2 (Wannier90) non-collinear ver.	/work/app/VASP6/vasp.6.1.2-wannier90v1.2/bin/vasp_ncl /work/app/VASP6/vasp.6.1.2-wannier90v2.1.0/bin/vasp_ncl

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

aprun [ -n MPI total tasks ][-N MPI tasks per node ] -j 1
/work/app/VASP5/current/bin/vasp_std > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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

aprun -n 36 -N 36 -j 1 /work/app/VASP5/current/bin/vasp_std > vasp.out 2> vasp.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
17.1	/work/app/WIEN2k/current
18.2	/work/app/WIEN2k/WIEN2k_18.2
19.1	/work/app/WIEN2k/WIEN2k_19.1
19.2	/work/app/WIEN2k/WIEN2k_19.2
21.1	/work/app/WIEN2k/WIEN2k_21.1

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=/work/app/WIEN2k/current
export PATH=$WIENROOT:$PATH
module load intel

aprun -b -d parallel number -j 1 --cc depth wien2k Script Option > output file
2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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=/work/app/WIEN2k/current
export PATH=$WIENROOT:$PATH
module load intel

aprun -b -d 36 -j 1 --cc depth 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.

(Example)

```
$ 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
4.0	/work/app/SIESTA/current
4.1.5	/work/app/SIESTA/siesta-4.1.5

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/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)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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

aprun -n 36 -N 36 -j 1 /work/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
8.8.2	/work/app/ABINIT/current/src/98_main/abinit
8.10.2	/work/app/ABINIT/abinit-8.10.2/src/98_main/abinit
8.10.3	/work/app/ABINIT/abinit-8.10.3/src/98_main/abinit
9.2.2	/work/app/ABINIT/abinit-9.2.2/src/98_main/abinit

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/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)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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

aprun -n 36 -N 36 -j 1 /work/app/ABINIT/current/src/98_main/abinit < input.file
> 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 have 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
4.1	/work/app/CPMD/current
4.3	/work/app/CPMD/CPMD4.3

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/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)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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

aprun -n 36 -N 36 -j 1/work/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 QUANTUM ESPRESSO

Available executables

Version	Path
6.2.1	/work/app/QuantumESPRESSO/current
6.3	/work/app/QuantumESPRESSO/qe-6.3
6.4.1	/work/app/QuantumESPRESSO/qe-6.4.1
6.5	/work/app/QuantumESPRESSO/qe-6.5
6.6	/work/app/QuantumESPRESSO/qe-6.6
6.7	/work/app/QuantumESPRESSO/qe-6.7

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ]  -j 1
/work/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)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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

aprun -n 36 -N 36 -j 1 /work/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
```

6.12 LAMMPS

Available executables

Version	Path
31 Mar 17	/work/app/LAMMPS/current
22 Aug 18	/work/app/LAMMPS/lammps-22Aug18
12 Dec 18	/work/app/LAMMPS/lammps-12Dec18
5 Jun 19	/work/app/LAMMPS/lammps-5Jun19
7 Aug 19	/work/app/LAMMPS/lammps-7Aug19
3 Mar 20	/work/app/LAMMPS/lammps-3Mar20
29 Oct 20	/work/app/LAMMPS/lammps-29Oct20

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/app/LAMMPS/current/src/lmp_intel_omp  < input file > output file 2> error
file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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

aprun -n 36 -N 36 -j 1 /work/app/LAMMPS/current/src/lmp_intel_omp < in.ij >
lammps.out 2> lammps.err

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

6.13 OpenMX

Available executables

Version	Path
3.8	/work/app/OpenMX/current
3.9.1	/work/app/OpenMX/openmx3.9.1
3.9.2	/work/app/OpenMX/openmx3.9.2

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ]  -j 1
/work/app/OpenMX/current/source/openmx  input file > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N openmx

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

aprun -n 36 -N 36 -j 1 /work/app/OpenMX/current/source/openmx C60.dat > C60.out
2> C60.err

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

6.14 SMASH

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ]  -j 1
/work/app/SMASH/current/bin/smash < input file > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N smash

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

aprun -n 36 -N 36 -j 1 /work/app/SMASH/current/bin/smash < mp2-energy.inp >
mp2-energy.out 2> mp2-energy.err

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

6.15 TOMBO

TOMBO is available for users who have own license.

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

After checking the license, TOMBO is ready for you.

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

mkdir tmp
aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/app/TOMBO/current/main > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N tombo

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

mkdir tmp
aprun -n 36 -N 36 -j 1 /work/app/TOMBO/current/main > tombo.out 2> tombo.err

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

6.16 RSDFT

Create a script file in advance.

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

export MPICH_NO_BUFFER_ALIAS_CHECK=1
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/app/RSDFT/current/src/rsdft.x > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N rsdft

export MPICH_NO_BUFFER_ALIAS_CHECK=1
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

aprun -n 16 -N 16 -j 1 /work/app/RSDFT/current/src/rsdft.x > rsdft.out 2> rsdft.err

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

Specify MPI tasks as the product of the value of PROCS in *fort.1*.

(Example)

```
$ grep PROCS fort.1
PROCS  2 2 4 1 1 1              / process partioning
```

-> 2*2*4*1*1*1=16

6.17 HPhi

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 OMP_NUM_THREADS= parallel number

aprun [ -n nodes ][ -d parallel number ] -j 1 --cc depth
/work/app/HPhi/current/build/src/HPhi -s input file > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N hphi

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME
export OMP_NUM_THREADS=36

aprun -n 1 -d 36 -j 1 --cc depth /work/app/HPhi/current/build/src/HPhi -s  stan.in
> hphi.out 2> hphi.err

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

6.18 mVMC

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/app/mVMC/current/build/src/mVMC/vmc.out -s input file > output file 2>
error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N mvmc

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

aprun -n 36 -N 36 -j 1 /work/app/mVMC/current/build/src/mVMC/vmc.out -s
StdFace.def > mvmc.out 2> mvmc.err

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

6.19 CP2K

Available executables

Version	Path
7.0	/work/app/CP2K/current
8.1.0	/work/app/CP2K/cp2k-8.1.0

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 CP2K_DATA_DIR=/work/app/CP2K/current/data

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/app/CP2K/current/exe/CRAY-XC50-cce/cp2k.popt input file > output file 2>
error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N cp2k

DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME
export CP2K_DATA_DIR=/work/app/CP2K/current/data

aprun -n 36 -N 36 -j 1 /work/app/CP2K/current/exe/CRAY-XC50-cce/cp2k.popt
H2O-32.inp > cp2k.out 2> cp2k.err

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

6.20 Elk

Available executables

Version	Path
6.3.2	/work/app/Elk/current/src/elk
6.8.4	/work/app/Elk/elk-6.8.4/src/elk

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

module load intel
export OMP_NUM_THREADS= parallel number
aprun [ -n nodes ][ -d parallel number ] -j 1 --cc depth
/work/app/Elk/current/src/elk > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N elk

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

module load intel
export OMP_NUM_THREADS=36
aprun -n 1 -d 36 -j 1 --cc depth /work/app/Elk/current/src/elk > elk.out 2> elk.err

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

6.21 ALAMODE

Create a script file in advance.

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

module load intel
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/work/app/ALAMODE/current/spglib/¥
install_dir/lib
DIRNAME=`basename $PBS_O_WORKDIR`
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

export OMP_NUM_THREADS= parallel number
aprun [ -d parallel number ] -j 1 --cc depth
/work/app/ALAMODE/current/anphon/anphon input file> output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N alamode

module load intel
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/work/app/ALAMODE/current/spglib/¥
install_dir/lib
DIRNAME=`basename $PBS_O_WORKDIR`
DIRNAME=`basename $PBS_O_WORKDIR`
WORKDIR=/work/$USER/$PBS_JOBID
mkdir -p $WORKDIR
cp -raf  $PBS_O_WORKDIR $WORKDIR
cd $WORKDIR/$DIRNAME

export OMP_NUM_THREADS=36
aprun -d 36 -j 1 --cc depth /work/app/ALAMODE/current/anphon/anphon test.in >
anphon.out 2> anphon.err

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

6.22 SALMON

Available executables

Version	Path
1.2.1	/work/app/SALMON/current
2.0.0	/work/app/SALMON/SALMON2-v.2.0.0

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/app/SALMON/current/bin/salmon.cpu < input file > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N salmon

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

aprun -n 36 -N 36 -j 1 /work/app/SALMON/current/bin/salmon.cpu < test.inp >
salmon.out 2> salmon.err

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

6.23 OCTOPUS

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/app/OCTOPUS/current/bin/octopus > output file 2> error file

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N octopus

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

aprun -n 36 -N 36 -j 1 /work/app/OCTOPUS/current/bin/octopus > octopus.out 2>
octopus.err

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

6.24 Wannier90

Available executables

Version	Path
1.2	/work/app/Wannier90/wannier90-1.2
2.1.0	/work/app/Wannier90/wannier90-2.1.0
3.1.0	/work/app/Wannier90/current

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

aprun [ -n MPI total tasks ][ -N MPI tasks per node ] -j 1
/work/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)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#PBS -N wannier90

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

aprun -n 36 -N 36 -j 1 /work/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 Usage of Python

7.1 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.

7.2 Install pyenv

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

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

7.3 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
```

7.4 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.
```

7.5 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

7.6 Execution method

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

(Example)

```
#!/bin/sh
#PBS -l select=1
#PBS -q P_016
#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

aprun python program.py

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