

## Shaheen II Get Started

### Shaheen II Spec:

<b>System</b>	<b>Cray XC40 with 36 cabinets</b>
<b>Processor type</b>	Intel Haswell 2.3GHz, 2 CPU sockets per node, 16 processors cores per CPU
<b>Total Nodes</b>	6174 nodes
<b>Total Cores</b>	197,568 cores
<b>Memory</b>	128 GB of memory per node, over 790 TB total memory
<b>Interconnect</b>	Cray Aries with Dragonfly topology
<b>Scheduler</b>	SLURM
<b>Storage</b>	Lustre parallel file system with 17.4 PB

### To login:

```
$ ssh <username>@shaheen.hpc.kaust.edu.sa
```

Login with “-X” or “-Y” to enable X11 forwarding.

### To compile:

Three programming environments are supported. i.e. PrgEnv-cray (default), PrgEnv-intel, and PrgEnv-gnu. Use module swap to change PrgEnv, e.g.

```
$ module swap PrgEnv-cray PrgEnv-intel
```

Use the compiler driver wrappers cc, CC, ftn to compile and link C, C++, and Fortran codes, respectively. The wrappers are the same for all programming environments. For example

<b>C</b>	cc -c <any_other_flags> prog.c
<b>C++</b>	CC -c <any_other_flags> prog.cpp
<b>Fortran</b>	ftn -c <any_other_flags> prog.f90

Within a programming environment a user can switch between different compiler versions.

```
$ module swap cce cce/8.4.0
```

## Scheduler and Queues

### To run:

- SLURM is the batch scheduler. The following is a basic example of a batch script:

```
#!/bin/bash
#SBATCH --account=k##
#SBATCH --job-name=job_name
#SBATCH --output=job_name.out
#SBATCH --error=job_name.err
#SBATCH --nodes=4
#SBATCH --time=00:30:00

srun --ntasks=128 --hint=nomultithread --ntasks-per-node=32 --ntasks-per-socket=16 ./exe
```

- Hyperthreading is enabled by default and might improve the performance of some codes. Use “--hint=nomultithread” option in the execution line to disable it. For binding tasks to cores you can use “--cpu\_bind=rank”.
- Launch/Cancel jobs with:

```
$ sbatch job_script.sh
$ scancel job_id
```

### Queues:

- Use “**sinfo**” for the queue status and “**squeue**” to observe your job status.
- workq**: Default queue with a maximum 24 hours wall clock time.
- 72hours**: Queue with a maximum of 72 hours wall clock time. Mainly intended for pre/post processing and running one node job that cannot be check-pointed to finish within a stipulated 24 hours wall clock time. Add the following in your job script to use the 72 hours queue.

```
#!/bin/bash
#SBATCH --partition=72hours
#SBATCH --qos=72hours
```

## Storage, Quotas, Allocations

### To store:

- Storage Compute nodes can access only **/scratch** and **/project** directories. Jobs submitted from /home will fail.
- /home/<username>**: Home directory, designed for development, quota of 200GB. Previous versions of files can be recovered from /home/<username>/.snapshot directory.
- /scratch/<username>**: Temporary individual storage for data needed for execution. Files not accessed in the last 60 days will be deleted.
- /scratch/project/k##**: Temporary storage for the project. Files not accessed in the last 60 days will be deleted.
- /project/k##**: 20 TB per project. All files are copied to tape. Once a project has used 20 TB of disk storage, files will be automatically deleted from disk with a weighting based on date of last access. Stub files will remain on disk that link to the tape copy.
- /scratch/tmp**: temporary folder that will be cleaned every 3 days.

### To check:

- Your Group information, use the “**groups**” command
- Your allocation information use the “**sb**” command, e.g.

```
username@cdl3:~> sb k##
Project k##: MEMBER_INFO
PI: PI_NAME
Allocations  Core hours
-----
2015-06-25    50000000
2014-09-02         50
-----
Expiry on 2020-01-01
-----
Allocated    50000050
```

## Compiler Flags

- Following table displays some advanced flags

Feature	Cray	Intel	GNU
Recommended compiler optimization level	default (-O3)	default (-O2)	-O3 -ffast-math
Aggressive Optimization	-O3 -hfp3	-Ofast -fp-model fast=2	-Ofast -maxv -funroll-loops
Activate OpenMP directives and pragmas in the code	-homp (default)	-openmp	-fopenmp
Desactivate OpenMP	-hnoomp		
Read and write Fortran unformatted data files as big-endian	-h byteswapio	-convert big_endian	-fconvert=swap
Process Fortran source using fixed form specifications.	-f fixed	-fixed	-ffixed-form
Process Fortran source using free form specifications.	-f free	-free	-ffree-form
Show version number of the compiler.	-V	--version	--version
Zero fill all uninitialized variables.	-h zero	not implemented	-finit-local-zero
Creates .mod files to hold Fortran90 module information for future compiles	-e m		
Specifies the directory to which file.mod files are written when the -e m option is specified	-J dir_name		
Listing compiler feedbacks, produces .lst files	-hlist=a	-opt-report3	-fdump-tree-all

- For more information on individual compilers

PrgEnv	C	C++	Fortran
PrgEnv-cray	man craycc	man crayCC	man crayftn
PrgEnv-intel	man icc	man icpc	man ifort
PrgEnv-gnu	man gcc	man g++	man gfortran

## Software & Libraries

- Before requesting the installation of new packages or libraries, please check if the desired package is already installed on the system.

- To find the list of all the packages installed:

```
$ module avail
```

- To find a specific package:

```
$ module avail --long 2>&1 | grep xxxx
```

- To get information on the package usage:

```
$ module help xxxx
$ module show xxxx
```

- To display Cray Scientific Libraries execute the following line:

```
$ module avail -L
```

Here is a selection of libraries and applications already installed on Shaheen II:

- I/O Libraries
  - HDF5, NetCDF
- Numerical Libraries
  - LIBSCI, PETCS, FFTW, MKL, ...
- Visualization Tools
  - Gnuplot, Paraview
- Debugging Tools
  - lgdb, atp, Allinea, Totalview, STAT
- Performance tools
  - Craypat, Allinea, PAPI
- Some Third party Software
  - VASP, CP2K, NAMD, LAMMPS, ...

## General Tips

- Currently, static linking is the default. To switch between different link types you can either set CRAYPE\_LINK\_TYPE to "static" or "dynamic" or pass the "-static" or "-dynamic" option to the linking wrapper (cc, CC or ftn).
- LIBSCI is the collection of numerical routines optimized for best performance on Cray systems. It gathers BLAS, LAPACK, SCALAPACK and is highly recommended to be used instead of your own versions.
- When calling libraries installed by Cray, such as LIBSCI, HDF5, NetCDF you do not need to add -I, -L and -l flags. Instead, you will have to remove these paths from your Makefiles.
- Default I/O striping is 1, optimal for many cases especially when every MPI process writes to its own file resulting in as many files as number of processes used.
- Increase the stripe count when multiple processes write to a single shared file as with MPI-IO and HDF5 or NetCDF. Use the following command with a maximum stripe count of 144:

```
$ lfs setstripe --count [stripe-count] filename/directory
```

### To get an account:

- KAUST members should fill-in the Individual Access Application (IAA) and the Project Proposal (PP) forms. Forms are available at: <http://hpc.kaust.edu.sa/account-applications>

### For more information:

- Please visit the user guide and training materials at: <http://hpc.kaust.edu.sa/>
- Please email any issues/forms to [help@hpc.kaust.edu.sa](mailto:help@hpc.kaust.edu.sa) so that the KSL staff can get back to you immediately.
- Follow us on Twitter: [twitter.com/KAUST\\_HPC](https://twitter.com/KAUST_HPC)