جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

KAUST SUPERCOMPUTING CORE LAB

Shaheen III CPU Partition

Getting Started

System Specifications

System	HPE Cray EX with 18 cabinets
Processor	AMD EPYC 9654 (Genoa family) 2 sockets/node, 96 cores/socket
Total nodes	4608 **Total cores** 884,736
Memory	384 GB per node, 1.76 PB total
Interconnect	Slingshot-11 200 Gbps
Job scheduler	SLURM
Storage	2 Lustre parallel file-systems Scratch 32PB; Project 57 PB

Login

```
ssh portalUserName@shaheen.hpc.kaust.edu.sa
```

To enable X11 forwarding, use the following command:

```
ssh -XY portalUserName@shaheen.hpc.kaust.edu.sa
```

Compile Source Code

Four programming environments are supported: `PrgEnv-cray` (default), `PrgEnv-intel`, `PrgEnv-gnu`, and `PrgEnv-aocc`. Use `module swap` to change PrgEnv, e.g.,

```
module swap PrgEnv-cray PrgEnv-gnu
```

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

C	`cc prog.c`
C++	`CC prog.cpp`
Fortran	`ftn prog.f90`

Within a programming environment, another version of a compiler can be switched to:

```
module swap cce cce/20.0.0
```

The default version of cpe (Cray Programming Environment) is shown as follows:

```
module av -S cpe
----- /opt/cray/pe/modulefiles -----
cpe/24.11 cpe/25.03(default) cpe/25.09
```

A newer version of cpe can be loaded as follows:

```
module load cpe/25.09
```

Scheduler and Queues

Run

SLURM is the batch scheduler. Here is a basic script:

```
#!/bin/bash
#SBATCH --partition=workq
#SBATCH --nodes=2
#SBATCH --ntasks=384
#SBATCH --ntasks-per-node=192
#SBATCH --cpus-per-task=1
#SBATCH --hint=nomultithread
#SBATCH --account=k1####
#SBATCH --time=01:00:00
#SBATCH --mail-type=ALL
#SBATCH --mail-user=emailAddress
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}
srun --hint=nomultithread -n ${SLURM_NTASKS} -c
    ↪ ${OMP_NUM_THREADS} --cpu-bind=cores ./
    ↪ executable
```

Hyperthreading is enabled by default and might improve the performance of some codes. Use `--hint=nomultithread` option in the execution line to disable it.

Submit a job:

```
sbatch job_script.sh
```

List jobs:

```
squeue --me
```

Cancel a job:

```
scancel id_of_my_job
```

Queues

Use `sinfo` for the queue 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.

```
#SBATCH --partition=72hours
#SBATCH --qos=72hours
#SBATCH --time=72:00:00
```

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

KAUST SUPERCOMPUTING CORE LAB

Storage, Quotas, Allocations

Storage

	/home	/project	/scratch
Login nodes	RW	RW	RW
Compute nodes	No access	R	RW
DTN*	RW	RW**	RW

R:Read W:Write

* DTN: Data Transfer Nodes

** Accessible via this path: `/lustre2/project`

- Compute nodes can access only `/scratch` and `/project` directories. Jobs submitted from `/home` will fail.
- `/home/$USER`: Home directory
- `/scratch/$USER`: Temporary individual storage for data needed for execution. Files not accessed in the last 60 days will be deleted.
- `/scratch/project/k1####`: Temporary storage for the project. Files not accessed in the last 60 days will be deleted.
- `/project/k1####`: Persistent storage
- `/scratch/tmp`: temporary folder that will be cleaned every 3 days.

Default quotas, capacity and performance:

	#files	Size	Total	Read/Write Perf.
`/scratch/$USER` (capacity tier)	1 M	10 TB	25 PB	330/260 GB/s
`/scratch/$USER/bandwidth`		1 TB	7 PB	3750/2500 GB/s
`/scratch/$USER/iops`		50 GB	338 TB	10 M IOPS
`/project/k1####`	1 M	80 TB*	57 PB	200/120 GB/s

* Per Principal Investigator (PI)

Find out your project groups:

```
groups
```

Check remaining core hours for a project:

```
sb k1####
```

Check user storage quota:

```
kuq
```

Check project storage quota:

```
kpq k1####
```

Software & Libraries

Before requesting installation of software, please check if the desired software is already installed on the system:

```
module av
```

To find a specific software:

```
module av -S esso
```

To get information about the software usage:

```
module help espresso
```

To get information about the software:

```
module show espresso
```

Many modules provide example job scripts:

```
module show espresso
/sw/ex113genoa/modulefiles/espresso/7.5:

module-whatis
-------------------------------
Quantum ESPRESSO is an ...
-------------------------------
setenv  MY_EXAMPLE /sw/ex113genoa/espresso/7.5/
    ↪ intel2023/example
-------------------------------------
```

The examples folder includes all necessary files, along with a SLURM job script to run a sample job:

```
ls -1 /sw/ex113genoa/espresso/7.5/intel2023/
    ↪ example/01
qe.scf.in
Si.pbe-rrkj.UPF
z_jobs_shaheen
```

A sample job can be launched as follows:

```
cd $SCRATCH
cp /sw/ex113genoa/espresso/7.5/intel2023/example
    ↪ /01/* .
sbatch z_jobs_shaheen
```

Links

Supercomputing Lab: https://www.hpc.kaust.edu.sa/

Portal login: https://www.hpc.kaust.edu.sa/user/login

Apply for access:
https://www.hpc.kaust.edu.sa/content/apply-access

How to login to the CPU partition:
https://hpc.kaust.edu.sa/content/login-shaheen

User documentation: https://docs.hpc.kaust.edu.sa

Contact Us

help@hpc.kaust.edu.sa