Baobab Cluster Documentation

The cluster Baobab is a computing cluster at the University of Geneva.

Documentation and Resources

You can find documentation and resources for Baobab at the following locations:

- Baobab Website `https://baobab2.hpc.unige.ch`
- Interface Web `https://baobab2.hpc.unige.ch`
- Baobab Forum `https://baobab2.hpc.unige.ch/forum`

Cluster Usage

Connecting to the Cluster

To connect to the cluster, use the following SSH command:

```
ssh username@baobab2.hpc.unige.ch
```

Replace `username` with your actual username.

Using Graphical Applications

If you wish to use graphical applications on Baobab, you can connect with X11 forwarding using the following command:

```
ssh –Y username@baobab2.hpc.unige.ch
```

Loading Modules

To compile code with MPI and CUDA and have access to a recent compiler, load the ”foss” and ”CUDA” modules using the following commands:

```
module load foss
module load CUDA
```

You can add these module load commands to your `.bashrc` file in your user directory to load the modules automatically upon login.

Listing and Managing Modules

You can list available modules using the following command:

```
module list
```

To search for specific modules, you can use:

```
module spider ”appToLoad”
```

To remove modules, you can use:

```
module purge ”appToRemove”
```

Submitting a Job

Baobab uses the SLURM queuing system. To submit a job, you can use the `srun` command or create a job script specifying the execution configuration and submit it with `sbatch`.

Note: Never run a program on a shared machine without using the queuing system. Always use `sbatch` and avoid `mpirun` or `mpiexec` directly.

Example job script for `sbatch`:

```
#!/bin/sh
#SBATCH ——job—name=JobName
#SBATCH ——output=JobOutput.o%j
#SBATCH ——ntasks=20
#SBATCH ——partition=partition1,partition2,etc
#SBATCH ——time=01:00:00

echo $SLURM_NODELIST

srun ./YourProgram
```

Save this script in a file (e.g., `script.sh`) and submit it to the cluster using:

```
sbatch script.sh
```

Monitoring and Managing Jobs

To view pending and running jobs, you can use the `squeue` command. To limit the display to your jobs, use:

```
squeue —u username
```

You can also show details of a specific job using:

```
scontrol show <job_id>
```

To cancel a job, you can use `scancel` with either the job ID, job name, or your username:

```
scancel jobid
scancel jobname
scancel —u username
```

You can also use the web interface to view the cluster's status and job information.

Performance Measurement

When performing performance measurements, it's important to limit yourself to nodes of the same generation. Here's an example script to enforce this limitation:

```
#!/bin/sh
#SBATCH --job-name=JobName
#SBATCH --output=JobOutput.o%j
#SBATCH --ntasks=nProc
#SBATCH -p shared,cui
#SBATCH -t 01:00:00
#SBATCH --constraint=E5-2630V4

echo $SLURM_NODELIST

srun ./YourProgram
```

In this example, the result is written to the local /scratch/ partition, allowing you to measure write times without relying on shared storage.

DEBUG Partition

There is a debug partition with two nodes and a limited execution time of 15 minutes. It is useful for quick code tests before submitting to the shared partition.

Example usage of the debug partition:

```
#!/bin/sh
#SBATCH --job-name=JobName
#SBATCH --output=JobOutput.o%j
#SBATCH --ntasks=nProc
#SBATCH -p debug
#SBATCH -t 00:15:00

echo $SLURM_NODELIST

srun ./YourProgram
```