BTU AI CLUSTER START GUIDE

RUNNING JOBS ON THE HPC CLUSTER

New User Warnings

If you have never used a Cluster, or are not familiar with this cluster, YOU WILL WANT to read and follow the examples below to become familiar with how to run jobs on HPC. It is a common practice by new users to ignore this manual and simply try to run jobs without understanding what they are doing. Such carelessness can and WILL easily impact hundreds/thousands of critical jobs and users currently running on the cluster. If your actions compromise the health of the HPC cluster, your account will be LOCKED so please make sure you run through the examples below before you embark on running jobs.

Do NOT use the login nodes for work. If everyone does this, the login nodes will crash keeping HPC users from being able to login to the cluster.

Never submit large number of jobs (greater than 2) without first running a small test case to make sure all works as expected. Start slow and then ramp up with jobs once you are familiar with how things work.

If you have a question, in the first instance please carefully check this User Guide.

How to use HPC

Using a High Performance Computing Cluster such as the HPC Cluster requires at a minimum some basic understanding of the Linux Operating System.

It is outside the scope of this manual to explain Linux commands and/or how parallel programs such as MPI work. This manual simply explains how to run jobs on the HPC cluster.

When you login to HPC you are connected to what is called a login node. The HPC Cluster has several major components:

- Login nodes
- Compute nodes

The **head node** runs all of the cluster critical services.

The login nodes are meant for simple tasks such as submitting jobs, checking on job status, editing *(emacs, vi)* and performing simple tasks.

The **compute nodes** are the workhorse of the cluster. For computational work both Serial or Parallel, in Batch mode or Interactive mode, you will be using the compute nodes.

Getting started

You will need to have a basic understanding of Linux and Linux commands in order to use the system, unless you use an application that has a web or GUI interface to the compute resources.

HPC are shared resources that need to allocate user job requests amongst the available processors and memory in an equitable manner.

How do I login to the HPC Machines?

Windows

Use a secure shell client, e.g. MobaXterm

1) Here is the direct link to download the mobaxterm program
2) Once you have mobaxterm installed follow this guide

Note if you have cygwin installed, you can open a cygwin-terminal and then use ssh the same as for Linux and Mac below.

If you aren't sure what cygwin is, you can safely ignore the above line.

Linux and Mac

Use ssh on the command line

```
ssh username@IP
```

Note: Username is your AI username

How do I copy files/data to the HPC Machines?

Windows

Use MobaXterm. This is a GUI-based scp client for MS Windows-based computers that has a drag-and-drop facility and an inbuilt file editor. If you have cygwin installed, you can open a cygwin-terminal and then use ssh the same as Linux and Mac below.

- Download MobaXterm

Linux and Mac

Use the scp on the command line

```
scp file-to-name USERNAME@IP:~/HOME_DIR/SUB_FOLDER/new-filename
```

This will copy the file to a SUB_FOLDER and renaming it to new-filename

Now that I have connected to the HPC System - What do I do now

You need to use the Linux command line. Try this cheat sheet to get you started.

How do I edit my files on the HPC system?

Use a command line editor such as ‘vi’ or ‘emacs’. Here are some cheat notes to get you started:

- Vi cheat sheet
- UNIX tutorial for beginners

Home Folder

Home Folder is located at **/cta/users/<username>/**. This folder is for long term storage to keep your job files. It is a general purpose distributed file system.

Application software

To see a list of application software that is installed on the HPC, type the the **module avail** command from your ssh console once you have logged in. This will give a list of applications, along with their version numbers.

If you want any other software installed, or a different version, send a request email support@compecta.com. Please include a detailed information about the software like web page, version, license etc. Note that if the software requires a licence, the users of the software will need to purchase the licence.

Some software will only run on a single processor, in which case you will probably not see any speedup from running the software on the supercomputer compared to running it on your desktop, unless it requires large memory. However, many standard software packages can take advantage of multiple processors, or have parallel versions (e.g. using MPI) that can. Check the user guide for the software.

SLURM queuing system

HPC that are shared among many users typically use a job management system such as Torque, SLURM, SGE or Moab to manage submission and execution of jobs.

Jobs are not run directly from the command line, the user needs to create a job script which specifies both the required compute resources, libraries and the job’s application that is to be run.

The script is submitted to the job management system (queueing system) and if the requested resources (processors, memory, etc) are available on the system, the job will by run.

If not, it will be placed in a queue until such time as the resources do become available. In order to provide a fair share of the resources among users, the priority of jobs in the queue may be varied based on how much resources someone has used, so it is possible that jobs may not run in the order in which they have been submitted to the queue.

Users wanting to use the HPC need to understand how to use the queueing system, how to create the job submission script, as well as check its progress or delete a job from the queueing system.

Running jobs

Jobs are managed by a Resource Manager on the HPC Cluster. BTU AI Cluster uses SLURM Resource Manager for this purpose.

You need to login (using ssh) to the HPC Cluster and submit jobs using **`sbatch`** command.

SLURM Job Submission Scripts

You will find SLURM submission script templates in a the folder: **`/cta/share/`**

Copy the one you need to your work folder and modify it as required:

```
mkdir /cta/users/<username>/workfolder

cd /cta/users/<username>/workfolder/

cp /cta/share/example_submit.sh /cta/users/<username>/workfolder/my_experiment.sh

vim my_experiment.sh
```

Submitting jobs to the queue

Jobs are submitted to the system with the command below:

sbatch myscript.sh

See the page about SLURM Queueing System Commands for more information on creating job submission scripts.

SLURM Partitions (Job Queues)

SLURM Resource Manager has partitions which are job queues. These partitions has different limits and member nodes.

BTU AI HPC Cluster's Partitions are listed below:

These partitions and limits are subject to change in near future. Please check back here again. You can also see the active partitions and their limits with **`sinfo`** command on the cluster.

The SLURM Cheat Sheet

Essential SLURM Commands

Command	Description
`sbatch` sbatch [script]	Submit a batch job Example: $ sbatch job.sub
`scancel` scancel [job_id]	Kill a running job or cancel queued one Example: $ scancel 123456
`squeue` squeue	List running or pending jobs Example: $ squeue
`squeue -u userid` squeue -u [userid]	List running or pending jobs Example: $ squeue -u john

Submitting a SLURM Job Script

The job flags are used with SBATCH command. The syntax for the SLURM directive in a script is "#SBATCH <flag>". Some of the flags are used with the srun and salloc commands, as well for interactive jobs.

Resource	Flag Syntax	Description	Notes
partition	--partition=short	Partition is a queue for jobs.	default on
time	--time=01:00:00	Time limit for the job.	1 hour; default is 2 hours
nodes	--nodes=1	Number of compute nodes for the job.	default is 1
cpus/cores	--ntasks=4	Corresponds to number of cores on the compute node.	default is 1

resource feature	--gres=gpu:1	Request use of GPUs on compute nodes	default is no feature
memory	--mem=6000	Memory limit per compute node for the job. Do not use with mem-per-cpu flag.	default limit is 6000 MB per core in beegfs[101-108] nodes
memory	--mem-per-cpu=6000	Per core memory limit. Do not use the mem flag,	default limit is 6000 MB per core in beegfs[101-108] nodes
account	--account=users	Users may belong to groups or accounts.	default is the user's primary group.
job name	--job-name="hello_test"	Name of job.	default is the JobID
constraint	--constraint=gpu	compute-nodes	AVAIL_FEATURES
output file	--output=test.out	Name of file for stdout.	default is the JobID
email address	--mail-user=username@foo.com	User's email address	required
email notification	--mail-type=ALL --mail-type=END	When email is sent to user.	omit for no email

Job Reason Codes

These codes identify the reason that a job is waiting for execution. A job may be waiting for more than one reason, in which case only one of those reasons is displayed.

State	Code	Meaning
PENDING	PD	Job is awaiting resource allocation.
RUNNING	R	Job currently has an allocation.
SUSPENDED	S	Job has an allocation, but execution has been suspended.
COMPLETING	CG	Job is in the process of completing. Some processes on some nodes may still be active.
COMPLETED	CD	Job has terminated all processes on all nodes.
CONFIGURING	CF	Job has been allocated resources, but are waiting for them to become ready for use
CANCELED	CA	Job was explicitly cancelled by the user or system administrator. The job may or may not have been initiated.

FAILED	F	Job terminated with non-zero exit code or other failure condition.
TIMEOUT	TO	Job terminated upon reaching its time limit.
PREEMPTED	PR	Job has been suspended by a higher priority job on the same resource.
NODE_FAIL	NF	Job terminated due to failure of one or more allocated nodes.
InvalidQOS		The job's QOS is invalid.
PartitionNodeLimit		The number of nodes required by this job is outside of it's partitions current limits. Can also indicate that required nodes are DOWN or DRAINED.
PartitionTimeLimit		The job's time limit exceeds it's partition's current time limit.
QOSJobLimit		The job's QOS has reached its maximum job count.
QOSResourceLimit		The job's QOS has reached some resource limit.
QOSTimeLimit		The job's QOS has reached its time limit.

Please follow the links for more...

JOB REASON CODES
JOB STATE CODES

Software

System software

- Ubuntu 20.04 LTS - operating system
- SLURM resource manager

Compilers and parallel programming libraries

- GNU Compiler (GCC, GFortran)
- Java, Python, Perl, Ruby
- OpenMPI - library for MPI message passing for use in parallel programming over Infiniband and Ethernet
- ...and more. We will be updating this document in future.

Libraries

- Please run the **module avail** command from your ssh console to view a list of available applications.

Application software

- Gaussian, Python, Matlab, Anaconda and many more.
- Please run the **module avail** command from your ssh console to view a list of available applications.

Contacts and Help

For more information on HPC facilities, systems support, assistance with parallel programming and performance optimisation and to report any problems, contact the CompecTA Service Desk.

Need Additional Help?

Alternatively you can send an email to support@compecta.com this will create a support ticket automatically.

When reporting problems, please give as much information as you can to help us in diagnosis, for example:

- **Your username**
- **Queue/partition name**
- **Job ID(s)**
- **A copy of any error messages**
- **Command used to submit the job(s)**
- **Path(s) to scripts called by the submission command**
- **Path(s) to output files from your jobs**
- When the problem occurred
- What commands or programs you were trying to execute at the time
- A pointer to the program you were trying to run or compile