BA888: A comprehensive introduction to BU's Shared Computing Cluster (SCC)

June 6, 2024

Resources/Where to Start?

Our website: rcs.bu.edu

Sections of note: System Usage, Python, ML

Cheatsheets (under System Usage)

Academic FAQ

Recorded Tutorials/Slides

Examples

Shared Computing Cluster

Why use the SCC?

Login Nodes

Compute Nodes

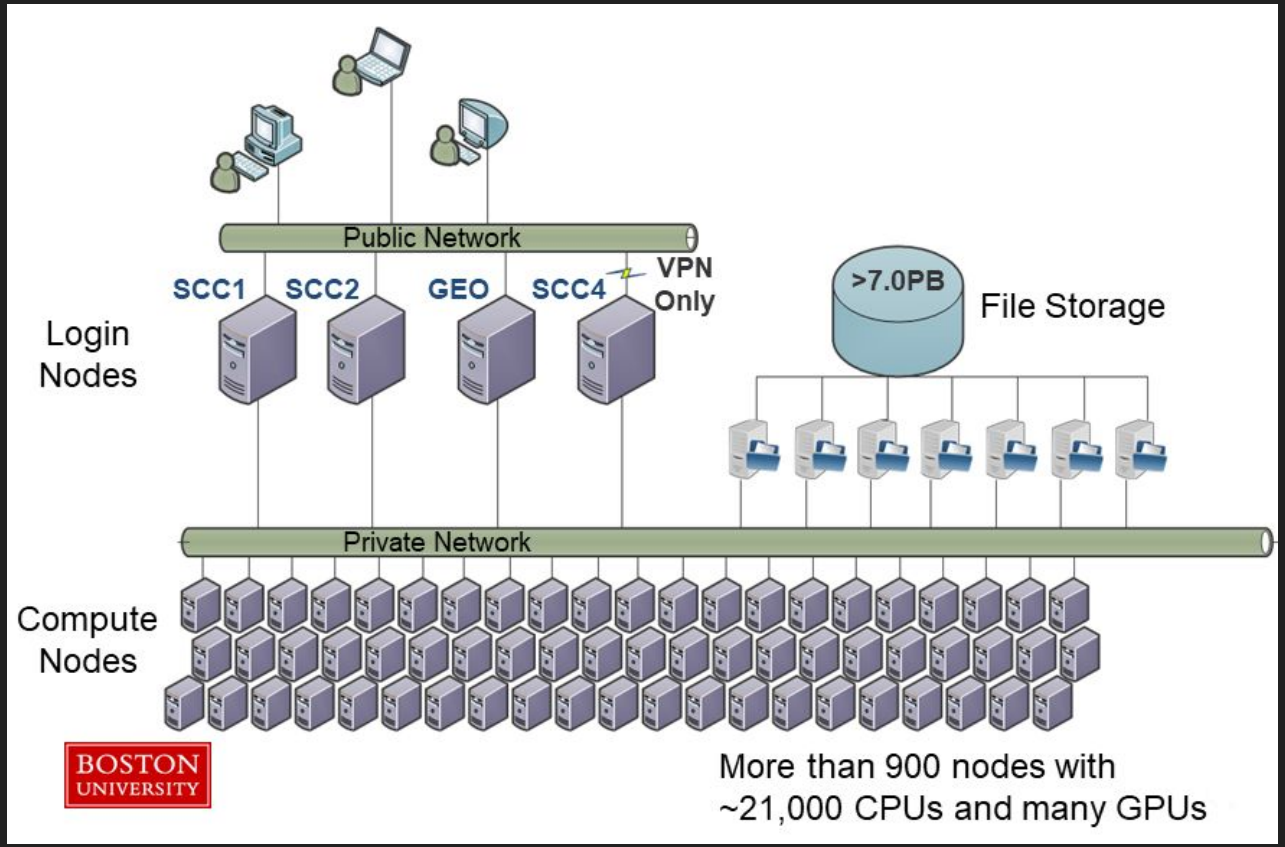
Interactive vs Batch

Terminal vs Graphical

ssh vs OnDemand

NFS

/scratch



OnDemand

Log into OnDemand: scc-ondemand.bu.edu

No account yet? scc-ondemand-tutorial.bu.edu

File browser

Built-in terminal: where is the running? what are the pitfalls?

Desktop session

Basic desktop activities

Filesystem

Linux basics (`man, pwd, cd, mkdir, cp, mv`)

Absolute/relative paths

Home/project space (`quota, pquota`)

File permissions (`chmod`)

snapshots

NFS vs /scratch (environment variables, `$TMPDIR`)

Modules: How to use software

Python as an example

Useful diagnostic tool (`which`)

`module avail`

`module load`

`module list`

Dependencies (for example: academic-ml)

Running a simple Python program

```
cd

cp -r /project/scv/classes/ba888me .

cd ba888me

ls

module load python3/3.10.12

python example.py
```

Batch System

Compute nodes for computation

Job resources: CPU, RAM, disk, time

Service Units (SUs)

Job killed

“error: no suitable queues”

GPU resources

Batch (non-interactive) Jobs

Why use a batch job?

Submit my_batch.qsub (`qsub`)

Get job status and delete (`qstat, qdel`)

Specify job resources

Looking at completed jobs (`qacct`)

Python Particulars

Installing packages (`pip`)

- Install to a project space (`--prefix`)
- PYTHONPATH

Virtual environments (`conda`)

- miniconda
- .condarc (use project space)
- Everything you need you should install in your conda environment

Notebooks (desktop vs server)

Machine Learning on the SCC

GPU usage (and `qgpus`)

Equivalent of `top` for GPUs (`nvidia-smi`)

Preconfigured conda env (academic-ml environment)

Custom ML conda env: verify correct install of GPU enabled PT/TF

PyTorch/Tensorflow resource usage

- `CUDA_VISIBLE_DEVICES`
- PT: num_workers for dataloader
- TF: intra_op_parallelism_threads vs. inter_op_parallelism_threads

Git on the SCC

git module

Using Github 2FA on the SCC

```
git clone https://github.com/userjjb/ProjectEulerMatlab

git remote -v

git status #modify file and git status again

git commit -m “Example commit”

git push origin master
```

Basic SQL usage on the SCC (sqlite terminal)

Simple example given in this tutorial:

```
module load python3/3.10.12
sqlite3
.mode csv
.import city.csv cities
SELECT
name
FROM
cities;
```

Basic sqlite usage on the SCC (Python)

Useful for reading in data and working with it programmatically (pre/post-processing)

```
less sql_example.py
```

```
python sql_example.py
```