HPC data handling

Sabry Razick
Topics

• Data staging
  ○ Using work
  ○ Using $SCRATCH
  ○ Using $TMP

• Check whether your job can run in parallel
  ○ Simple way using top

• Ask for the correct resources

• Make sure environment can be reproduced

• Investigate during the job run
Different locations

● To find out the mounts
  ○ `df -h`

● `/cluster` is slow
  ○ `$HOME` is on cluster

● `/work` is fast
  ○ `/work/users/<USERNAME>`
  ○ `$SCRATCH` is on work
  ○ Temp directory during a job is on work

● `$LOCALTMP` is the fastest
  ○ IT is a local disk on a compute node.
  ○ Not shared between nodes
  ○ Very good when there are millions of very small files to be processed.
Check the available directories

● Test job
  ○ mkdir $HOME/HPC_NOV2018
  ○ cp /cluster/teaching/abel_tutorial/NOV2018/location_test.slurm $HOME/HPC_NOV2018
  ○ cd $HOME/HPC_NOV2018
  ○ chmod +w location_test.slurm
  ○ nano location_test.slurm
    ■ Edit the project account
  ○ sbatch location_test.slurm
When there are hundreds of thousands of files

● Test job
  ○ mkdir /work/users/$USER/HPC_NOV2018
    ■ If you are a guest then mkdir $HOME/HPC_NOV2018
  ○ cp /cluster/teaching/abel_tutorial/NOV2018/millionfiles.slurm /work/users/$USER/HPC_NOV2018
  ○ cd /work/users/$USER/HPC_NOV2018
  ○ chmod +w millionfiles.slurm
  ○ nano millionfiles.slurm
    ■ Edit the project account
  ○ sbatch millionfiles.slurm
When there are hundreds of thousands jobs

- Create an archive (no compression)
  - `tar -cvf <ARCHIVE_NM> <FILES>`
  - `tar -cvf file.tar *txt`

- List content
  - `tar -tvf <ARCHIVE_NM>`

- Append a file
  - `tar --append --file <ARCHIVE_NM> <NEW_FILES>`
  - `tar --append --file files.tar.gz 1.txt`

- Extract all
  - `tar -xvzf <ARCHIVE_NM>`

- Extract one
  - `tar -xvf <ARCHIVE_NM> <FILES>`
Big file and lsof

● Test job
  ○ mkdir /work/users/$USER/HPC_NOV2018
  ○ cp /cluster/teaching/abel_tutorial/NOV2018/file_IO*/
    /work/users/$USER/HPC_NOV2018
  ○ cd /work/users/$USER/HPC_NOV2018
  ○ chmod +w file_IO.slurm
  ○ nano file_IO.slurm
    ■ Edit the project account
Big file and lsof

- **Test job**
  - sbatch file_IO.slurm
  - squeue -u $USER
    - When the job starts get the compute node address
    - Login to the compute node
      - ssh cx.xx
      - top -u $USER
      - Find PI and investigate with lsof
sacct

- Investigate resource usage after the job is finished
- `sacct -j <JOB_ID>`
- `sacct -j <JOB_ID> -o all`
Qlogin

- `qlogin --account=<ACCOUNT>  --ntasks=4 --mem-per-cpu=2G --time=1:00:00`
- `qlogin --account=<ACCOUNT> --partition=accel --ntasks=8 --mem-per-cpu=6G --gres=gpu:2 --time=02:00:00`