Radiation Oncology HPC Cluster

User Guide

- How to access:
  - SSH
    - host: rohpc.swmed.org
    - port:22
  - VNC
    - SSH to rohpc.swmed.org
    - Configure xstartup
    - Vncserver –geometry 1280x768
    - Connect to vnc session using free tightvnc or turbovnc
  - Web
    - https://rohpc.swmed.org:8081/userportal/#/login
    - Cluster overview, workload etc…

- Computing resources:
  - "sinfo" to see the partitions(queue) and compute nodes available
    - defq : default queue, 1 compute node: rohpc100: 24C/128GB
    - dgxq: v100 queue: 3 V100 GPU nodes: rodgx101, rodgx102: 8GPU/32GB, 80C/500GB
    - k80q: k80 queue: 1 K80 GPU node: rok8001: 16GPU/16GB
    - titansq: titans queue: 2 Titans nodes: rotitans[009-010]: 4GPU/6GB
    - mem192: 3 compute nodes : rohpc101, rohpc102, rohpc103: 24C/192GB
    - mem384: 2 compute nodes : rohpc301, rohpc302: 24C/384GB
    - *future GPU nodes: titans1-8 (16GPU/16GB)
  - scontol show nodes to see further details

- Data storage:
  - **Avoid** storing any data in:
    - /home
      - 2 TB capacity, daily snapshot, no backup
  - When storing data you can use
    - /data, /apps
      - 46 TB capacity, daily snapshot, no backup
    - **(NOT READY YET)** /archive
      - TBD capacity, mount from Research Storage >400TB
  - When running GPU jobs, can temporarily use the storage available below to save models
    - /rodgx101fs
      - 7 TB capacity, no snapshot, no backup
    - /rodgx102fs
      - 7 TB capacity, no snapshot, no backup
  - In the /data, /archive, /rodgx101fs, and /rodgx101fs, there is a folder hierarchy, so please make your user folder inside the "maia" folder.
- Using slurm:
  - slurm is the scheduler used on the hpc
  - https://slurm.schedmd.com/documentation.html
  - Submit jobs with slurm:
    - Examples in /apps/slurm-test: defq-test, dgxq-test, mem192q-test, mem384q-test

        - cat test-jobs.sh
        - ./test-jobs.sh
    - GPU run example:
        - You have a “helloworld.py” python file you want to run. Let’s say it requires CUDA 10.1 and cudnn 7.6. You make a .sh file (let’s call it “helloworld.sh”), with the following lines:

```
#!/bin/bash
#
#SBATCH --job-name=helloworld
#SBATCH --output=helloworld.txt
#
#SBATCH --partition=dgxq
#SBATCH --gres=gpu:dgx:1
#
#SBATCH --ntasks=1
#SBATCH --time=20:00:00
#SBATCH --mem-per-cpu=64

module load cuda10.1/profiler/10.1.243
module load cuda10.1/toolkit/10.1.243
module load cuda10.1/blas/10.1.243
module load cuda10.1/fft/10.1.243
module load cuda10.1/nsight/10.1.243

module load cudnn7.6-cuda10.1/7.6.5.32

srun python helloworld.py
```

        -
            - This particular setup is asking for 1 GPU on the dgx partition. The maximum runtime here is 20 hours, with 64 GB max memory per cpu.
                - Note: the higher resources you request, the lower your priority
            - “module load” allows you to customize your environment as needed. You can run “module avail” to see what is available to use.
        - You can then queue the slurm scheduler with
            - sbatch helloworld.sh
            - Note: you need to be in a python/conda environment when you queue for it to recognize the .py file properly

- SBGRID : Mount from BioHPC
    - https://portal.biohpc.swmed.edu/content/cloud-services/sbgrid/
    - SBGrid Software Suite contains 400+ scientific software applications
    - https://sbgrid.org
    - add . /programs/sbgrid.shrc
    - require VNC session
    - example application tested : ccp4i, maestro

- Docker:
    - Docker is installed on all the CPU nodes: rohpc100-rohpc103, rohpc301-302
    - If you plan to run docker on these nodes, your id must be added to the docker group
    - Example:
        - ssh myid@rohpc.swmed.org
        - >ssh myid@rohpc301
        - >module load docker
        - >docker pull hello-world
        - >docker run hello-world