Mass General Brigham

MLSC Compute Cluster

What it is and how to use it

Paul Raines

January 21, 2021

Athinoula A. Martinos Center for Biomedical Imaging

What is the MLSC Computer Cluster?

MLSC Cluster Overview

The MLSC Cluster consists of the equipment bought on the $2.2M computation infrastructure part of the MLSC Grant to the CTRU. It is located at the Marlborough Data Center.

- Computational Servers (4x A100 DGX, 8x EXXACT RTX, 32x Dell non-GPU)
- Network Infrastructure
  - 1x Mellanox 32 QSFP28 100GbE ports
  - 1x Mellanox 64 QSFP28 100GbE ports
  - 2x VAST 32 port Mellanox switches
  - 2x 100GbE uplinks to MGB corporate network
  - Out-of-band 1GbE management network
- Storage Infrastructure
  - VAST 1.35PB raw Universal Storage (about 1PB usable)
    - 8x CBOX servers with 2x 50GbE network
    - 2x DBOX servers w/ 676TB NVMe Flash, 18TB 3DXP, 4 x 100GbE
- Other Infrastructure
  - Racks, KVM, PDU, cables, support contracts
- Website: https://it.martinos.org/mlsc-cluster/

Computational Server Hardware

- Four NVIDIA A100 DGX servers
  - 8x A100 GPUs (40GB, NVLINK Switch)
  - 2x AMD EPYC 7742 64-core 3.40GHz CPUs
  - 1TB DDR RAM
  - 10x 100GbE fibre network ports
  - 2x 1.92TB NVMe SSD, 4x 3.84TB NVME U.2
- Eight EXXACT RTX GPU servers
  - Two servers with 8x RTX6000 (24GB)
  - Five servers with 10x RTX8000 (48GB)
  - One server with 4x RTX8000 (48GB)
  - 2x Xeon Gold 622R 2.90GHz 16-core CPUs
  - 1.5TB DDR RAM
  - 2x 100GbE fibre network ports
  - 2x 250GB SSD, 2x 3.84TB SSD
- Thirty Dell PowerEdge R440 servers
  - 2x Xeon Silver 4214R 12-core 2.40GHz CPUs
  - 384GB DDR RAM
  - 2x 25GbE fibre network ports
  - 480GB SSD, 1.6 TB NVME
- Two Dell PowerEdge R440 servers
  - 2x Xeon Silver 4214R 12-core 2.40GHz CPUs
  - 384GB DDR RAM
  - 2x 25GbE fibre network ports
  - 2x 480GB SSD, 4x 2TB SATA
  - Used for login and control, not jobs

Computational Server Software

- Operating System
  - CentOS8 on Dell and RTX servers
  - Ubuntu 18.04 on A100 DGX servers
- Job Management
  - SLURM 20.02
    - GPU resource allocation
    - Accounting tools
    - Fairshare priority system
- Systems Management
  - Configuration: Puppet 6
  - Monitoring: Nagios, nhc
  - Accounts: sssd ➔ Martinos LDAP ➔MGB/Partners AD
- Other notable software
  - MATLAB (users encouraged to compile)
  - Singularity for running docker containers
  - Freesurfer, FSL, R, python, etc.

How do I use the MLSC Compute Cluster?

Getting Access

- PI needs to establish a SLURM group account for their LAB with Martinos IT
  - PI sends email request to help@nmr.mgh.harvard.edu
  - Can also request volume on VAST storage at this time
  - PI will get a Martinos account if they don't already have one. If they will not need to submit jobs the account can be a "fee"less non-login account.
- PI requests Martinos Center full login Linux account for each user who will submit jobs
  - PI fills out https://www.nmr.mgh.harvard.edu/computer/requestAccount
  - Full login accounts have fee of $16.67/month
- Users need to ask for MLSC cluster access
  - User or PI sends email request to help@nmr.mgh.Harvard.edu
  - Request must state which SLURM group account (Lab) they will submit jobs under
  - Users can be in multiple SLURM group accounts (specified when job submitted)
  - Users will be added to **batch-users** mailing list (mandatory)
- Users must submit jobs from the MLSC login server
  - Connect via SSH at **mlsc.nmr.mgh.harvard.edu**
  - Users can ssh to nodes where they have running jobs for monitoring/debugging

SLURM terminology and defaults

- **SLURM group account** – group of users working on same projects
  - Possible for Lab to have multiple group accounts if need demonstrated
  - Fairshare priority system is currently implemented by these groups, not by users
- **job** – submission of analysis that defines resource allocations needed
- **partition** – a group of nodes with a certain set of resource limits and defaults
  - equivalent of queues in PBS/torque (In SLURM, queue is just the job queue)
- **node** – one of the physical servers (A100 DGX, EXXACT RTX, Dell R440)
- **resource** – memory, time, GPU, cores, etc required by job
- **task** – division of a job with multiple processes (ex: MPI). Typically just one per job.
  - Separate programs/processes you run at the same time in the same job
- **cpu/core** – physical CPUs are called sockets in SLURM. Hyperthreading is off.
  - number of cores usable by multi-threading capable programs like MATLAB
- **Defaults**:
  - Memory of 10GB: DGX's & RTX's have 120GB/GPU while Dell's have 15GB/core
  - Time of 4 hours
  - These small defaults are to force users to think about the resources they use so the system can manage priority and resources more efficiently

Resource Overview

Description	#Nodes	#Cores	RAM	#GPUs	Partition	Scratch	Name
Dell R440	30	24	384GB	none	basic	1.5TB	r440-XX
EXXACT RTX 6000 GPU	2	32	1.5TB	8	rtx6000	7.0TB	rtx-XX
EXXACT RTX 8000 GPU	5	32	1.5TB	10 (4), 4 (1)	rtx8000	7.0TB	rtx-XX
NVIDIA DGX A100	4	64	1.0TB	8	dgx-a100	14TB	A100-XX

Local scratch space is at **/scratch** on each node. TMPDIR is set to /scratch in the default environment for each node. Files are cleaned when over 7 days old. In the future we want to setup a scheme with a per job scratch directory under /scratch that is immediate erased on job exit.

Scratch space at **/vast/scratch** exists shared by all the nodes using the VAST storage.

Submitting Jobs via jobsubmit

- **jobsubmit** is a locally written wrapper around the native **sbatch** tool to make submission both simpler and help user avoid "default" mistakes
- Specifying memory and time limits are mandatory
- Specifying partition and SLURM group account are also mandatory
- Command to run on node is given *without* quotes after jobsubmit options are specified
- Full help found by running: **jobsubmit –h**
- If neither -o or -e arguments are specified, then
  - STDOUT will go to /cluster/batch/$USER/sjob_%n.out%j
  - STDERR will go to /cluster/batch/$USER/sjob_%n.err%j
  - %n is jobsubmits serial number and %j is SLURM's job ID
  - If only –o is giving, both STDOUT and STDERR go to the given file
- Example submission:

```
jobsubmit -p rtx6000,rtx8000 -A sysadm -m 120G -t 1-12:00:00 -c 3 -G 1 \
 -M END,FAIL python3 my_gpu_job.py -d /cluster/data/subject –epochs 100
```

Submitting Jobs via sbatch

- This method requires writing a script that you then submit with the **sbatch** command
- Options can be specified in the script itself or on the **sbatch** command line

```
#!/bin/bash
#SBATCH –account=sysadm
#SBATCH –partion=basic
#SBATCH –nodes=1
#SBATCH –ntasks-per-node=5
#SBATCH –mem=2G
cd /autofs/cluster/batch/raines/john-1.9.0-jumbo-1/run/
./john --test=300 --format=md5crypt
```

```
sbatch --time=2-05:00:00 --job-name=john001 --output=john001_%j.out \
--mail-type=END,FAIL testjob.sh
```

Submitting Interactive Jobs via srun

- Interactive jobs are submitted using the **srun** command with **--pty**
- It is important for users to actually use the shells on the nodes they get this way and not leave them idle overnight. Fairshare weighting applies actually clock time used, not CPU time, so idle time affects your future job priority just as much as time spent using 100% the CPU or GPU.
- To run X11 GUI apps from a job, include the **–x11=first** option
- Example submission asking for 2 RTX8000 GPUs for 1 day 10 hours with 6 CPU cores and 256 RAM:

```
srun -p rtx8000 -A sysadm -N 1 --ntasks-per-node=1 --gpus=2 --mem=256G \
--time=1-10:00:00 --cpus-per-task=6 --pty /bin/bash
```

Other SLURM tools

- **squeue** – show the job queue and job states
- **sinfo** – show overview of cluster node status and queues they are in
- **scontrol** – show and modify details for running job or node
  - **scontrol show job=88134**
- **sshare** – show fairshare stats
- **sacct** – show info on finished jobs
  - **sacct --starttime 2020-09-08 -o "JobID,Elapsed,CPUTime,MaxRSS,TotalCPU,State"**
- **scancel** – cancel submitted or running job
- **sattach** – view the stdout/stderr of running job
- **sprio** – show details of a job's priority calculation in the queue if waiting

All commands have man pages that can be read on the MLSC login node

Users should develop/compile programs on their own hardware but can use an interactive job in a pinch.

Some PBS/Torque commands exist for migration assistance but beware: they are missing many key options like the "-l" resource option for qsub

Fairshare priority system

- Each SLURM group account gets the same 200 "raw" shares
  - 200 is basically an arbitrary number big enough to give some granularity. What matters is relative value compared to other groups. But since right now everyone has the same, it is immaterial.
  - Still working on share policies though. May allow temporary increased shares for labs with well defined short term deadlines. Or "bigger" groups gets more share with what bigger means some combination of total funding, number of users, etc. Or maybe “bigger” groups get multiple accounts.
- Job priority in the submission queue is based on the group account's share weighted by a usage factor by the resources used by the group
  - Each resource (GPU, CPU, memory) has a weight with "better" resources having more weight than slower ones (i.e. a A100 GPU has more weight than a RTX8000 which has more than a RTX6000)
  - The older the usage, the less its weight with a half life factor of 1 month (a job you ran a month ago will "weight down" your fairshare half as much as an equivalent job you just ran today)
  - Final fairshare factor for a slurm group account is normalized to be between 1.0 (account has run no jobs recently so has highest priority) and 0 (no share left – account will only have jobs run when nodes are free and no accounts with non-zero share have jobs waiting)

Fairshare priority system monitoring

Use the **sshare** command to monitor the Fairshare system. Below results obtained for example using the following command on the MLSC login node.

```
sshare | egrep '(arnoldgp|lcn|roffmagp|qtim|visuo|Fair)' | cut -c1-15,34-
```

Account	RawShares	NormShares	RawUsage	EffectvUsage	FairShare
arnoldgp	200	0.066644	0	0.000000	1.000000
fsdev	200	0.066644	135029786	0.191098	0.137032
lcn	200	0.066644	334436023	0.473300	0.007280
qtim	200	0.066644	3830317	0.005421	0.945179
roffmagp	200	0.066644	89617483	0.126832	0.267367
visuo	200	0.066644	0	0.000000	1.000000

See your labs usage breakdown by user with for example: **sshare --account=lcn -a**

More on VAST storage

- 10TB limit per lab with administration allowing more on case by case basis
- Same price of $26.67/TB/month as our Dell storage
- Ultra High performance only over NFS to cluster nodes
- Better performance over NFS than Dell storage to non-cluster Martinos Linux systems
- SMB/CIFS access to VAST available by Dell gateway proxy server
- Each lab volume gets a weekly mirror backup to Dell storage
- Future features: snapshots, analytics

Mass General Brigham