COSMA 101:

Demystifying COSMA

cosmICConnections
2023-2024
Alastair Basden

News/Reminders

- Acknowledgements
- ga007: AMD MI300X GPU (x8)
  - ga008, MI300A (x4) system coming soon
- dine2 system
  - 8x 2TB nodes with GPU (A30)
- COSMA5 replacement
- cosma.readthedocs.io

COSMA refresher (October)

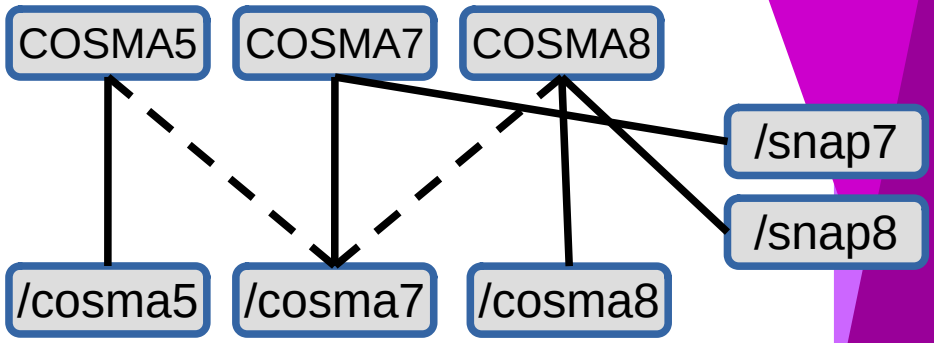
- Login nodes: login.cosma.dur.ac.uk, login5, login7, login8
  - Best to compile on the one you're submitting to
- COSMA5: ICC-only - submit to the cosma or cosma5 queue
  - cosma gives older nodes (16 cores/node)
  - cosma5 gives newer nodes (256 cores/node) - non-exclusive
- COSMA7: Submit to cosma7 or cosma7-rp
  - cosma7-rp will use the Rockport fabric, often shorter queues
- COSMA8: cosma8 (all 528 nodes), cosma8-rome (360 older nodes) or cosma8-milan (168 newer nodes)
- Other facilities: cosma7-shm, cosma7-shm2, cosma8-shm, cosma8-shm2, cosma8-shm3, bluefield1 (DINE), cosma8-serial, cordelia, dine2
- GPU nodes: gn001, cosma8-shm2, login8b
- /cosma/home, /cosma/apps
- /cosma5, /cosma7, /cosma8, /snap7, /snap8, /madfs

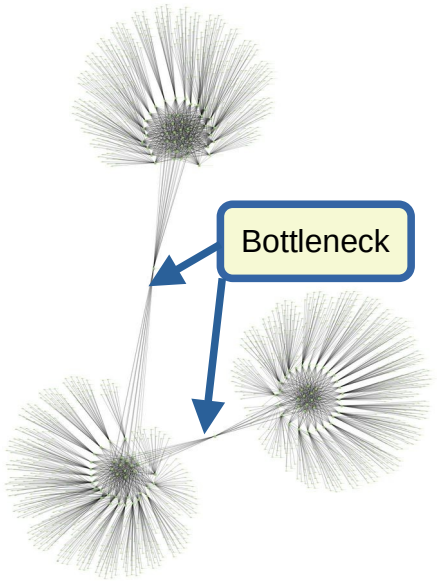
COSMA access tips (October)

- Making access easier: ssh connection sharing, memorising passphrases:
  - In your .ssh/config file (local, e.g. on your laptop):
    - Host login7.cosma.dur.ac.uk
      ControlPath ~/.ssh/controlmasters/%r@%h:%p
      ControlMaster auto
      ControlPersist yes
    - If your Internet connection changes, you'll need to kill this connection (it will time out)
  - Remembering your passphrase (enter it once per reboot):
    - Various options
      - Some OSs will do this automatically
      - Or, e.g. (in .bashrc): eval $(keychain --eval /home/ali/.ssh/id_rsa_cosma 2> /dev/null)
- x2go: a graphical desktop
- Jupyter ssh tunnel: ssh -N -L localhost:8443:login8b:443 USER@login8b.cosma.dur.ac.uk
  - Then visit https://localhost:8443

COSMA Filesystems (December)

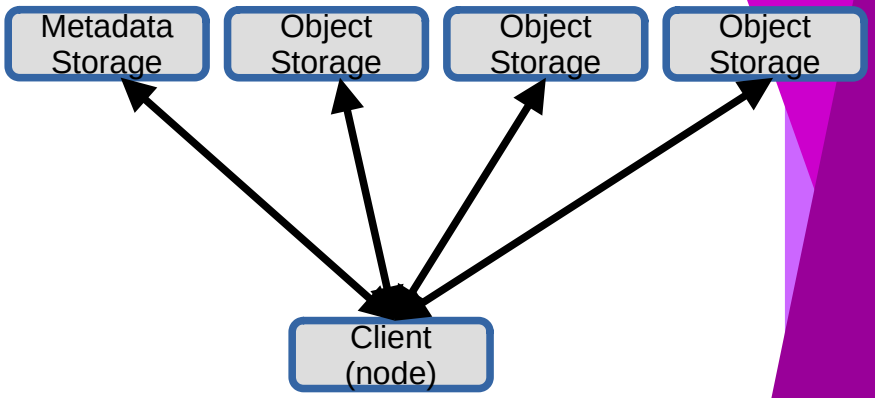
- /cosma/home: 10GB quota, backed up daily (38TB)
  - Not a parallel file system
  - Use for things like source code etc
  - Do not write to here from large parallel jobs
  - Hourly snapshots in a hidden .snapshot/ directory
- /cosma/apps: 100GB quota
  - Use for installing code, Python venvs, etc
- /cosma5,7,8: up to 23PB total
  - Best used from the appropriately numbered COSMA
  - Can often be read from other COSMAs reducing the need for copying data
    - But only recommended for infrequent use (if you need to read a lot, copy it across)
    - Quota can be increased upon request (default 5-10TB)
    - Not all mounted everywhere (e.g. /cosma7 not mounted on COSMA8 compute nodes)
- Which one to use, when
  - Use /cosma5 from COSMA5, use /cosma8 from COSMA8, etc
  - Login nodes mount everything (except /snap-specific)
  - If you try writing from a Slurm job and it doesn't return any output, probably writing to the wrong place





Advanced Lustre (January)

- COSMA uses Lustre for bulk file systems
  - A parallel file system: Designed to be used by many nodes in parallel
- Key command is “lfs” (Lustre File System?)
  - lfs quota /cosma8
  - lfs find (like find, but faster and with Lustre-specific options)
  - lfs df -h /cosma8 (like df, but shows the system components)
  - lfs getstripe /path/to/file
  - lfs setstripe --stripe-count N --stripe-size XM /path/to/newfile/or/dir
- Striping: Manually deciding how a file is split between disks
  - By default it will go to a single virtual disk
  - Can see performance improvements if splitting over several virtual disks on several nodes
  - Inherited from parent directory
  - Can also place small files on the metadata servers (for faster access)
- Lustre rsync
  - For copying data between file systems
  - module load rsynclustre
  - rsync -aAxvh --progress /from /to (take care with trailing /’s - might or might not be what you want)
    - Automatically stripes large files >1GB
    - Does not preserve existing striping



Slurm queues (February)

- Slurm is the system used for submitting jobs to the compute nodes
  - Once submitted, your job will sit in a queue (partition) until the requested nodes become available
    - Priority depends on several things: queue priority, job age, recent success, job size
- One main partition for each system (e.g. cosma8).
  - Note, cosma7 has two: cosma7 and cosma7-rp
    - which are the same size but have a different network fabric
  - Various other partitions for more bespoke work. e.g. the *-shm* queues (large memory, GPUs, etc)
  - Start using cosma5 (new nodes), rather than cosma (old nodes, decommissioned shortly)
- Useful commands:
  - squeue [-p cosma8] [-u username]
  - sinfo [-p cosma8] [-n NODE]
  - showq -f -l -p cosma8
  - scontrol show job=JOBID / scontrol update job=JOBID A=X B=Y
  - scontrol show partition=cosma8 - work out which -A and -p flags you need, max runtimes, etc
  - squota - wrapper script to show project hours remaining.
  - sprio - priority of jobs

squota example!

Usage for project dp203 on cosma7 for current quarter:
3769872 / 6202300 core hours used (60%), 2432427 hours remaining

Usage for project dp203 on cosma8 for current quarter:
18377941 / 27105970 core hours used (67%), 8728028 hours remaining

Usage for project dp004 on cosma7 for current quarter:
1529298 / 18248544 core hours used (8%), 16719245 hours remaining

Usage for project dp004 on cosma8 for current quarter:
15183782 / 70899836 core hours used (21%), 55716053 hours remaining

Modules (March)

- COSMA contains many different versions of different software libraries
  - Unlike a desktop, where an update will overwrite the previous one
  - To preserve past performance, work arounds, etc
- Specify which library to use with the “module” command (tab completion)
  - module avail (shows all the available modules)
  - module load MODULE/version
  - module purge, module unload
    - Note, Intel compiler modules don’t unload cleanly
  - module show MODULE/version, module help MODULE/version
  - module list (lists loaded modules)
- Some modules require combinations of others, and will usually give a hint
- Most large codes have a recommended set of modules known to work well
- Python: Load the correct module then use a virtual environment (see cosma.rtfd.io / cosma.readthedocs.io)
- Modules are periodically updated to newer versions
  - Please test these, and use if you don’t see any problems! They are usually(?) less buggy and offer better performance
- Some modules are specific for particular architectures (e.g. cosma7, cosma8)
- Newer intel modules: Once loaded, you then need to module load compiler mpi

Data curation (April)

Compiling and debugging tips (May)

- icx
- NUMA effects

Energy saving tips (June)

- Energy monitoring
  - Quarterly emails
  - energy.py script
- Code selection (e.g. SWIFT vs GADGET)
- Solar panels