SciOps + ENP April 2022

Topics

- Tape Bandwidth Planning
- Lustre improvements
- Operational Status
- User Documentation
- Future Topics

Bryan Hess, *bhess@jlab.org*

Friday, April 8, 2022

Tape Bandwidth Planning

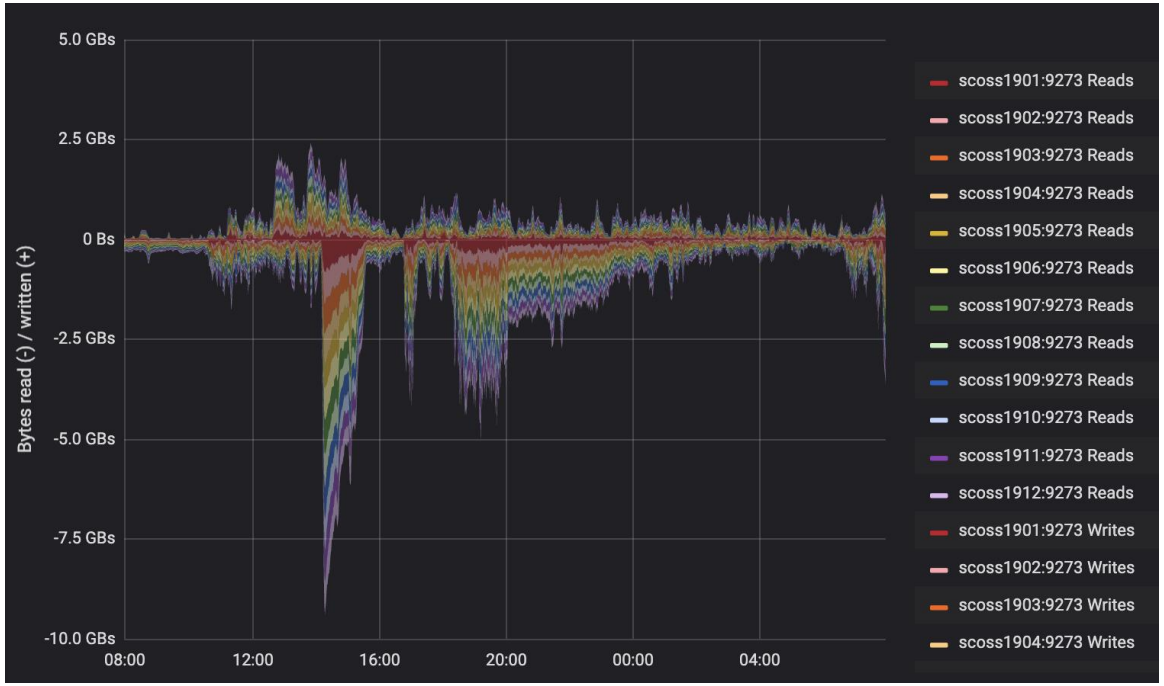
- We are adding four additional LTO8 drives to the tape library. The package is in procurement now. Lead times are long; I do not expect installation before mid-summer, but an outage is not needed.
- This will accommodate the expected data rate increases during the next run,and avoid starvation of the farm during data taking.
- This will fill every available drive bay, bringing us to
  - 24 LTO8 drives
  - 4 LTO7 drives
- We will fill all drive bays, but we have significant tape slot capacity, so data volume is not a concern at this point
- I will assemble a tape order sometime in the next month to prepare for the next run.

Tape Bandwidth Planning (2)

- To add drive capacity beyond this, we would need to either remove LTO7 drives, or add a drive frame.
- Case 1: Remove LTO7 drives
  - Requires that we migrate off LTO6 media (in progress)
  - Tape migration cannot be done during the run for bandwidth reasons, so we are finishing it now. (opportunistic background task)
  - This case can add 4 drives at most. 3 if we want to preserve the ability to read LTO5 and LTO6.
- Case 2: Add a drive frame
  - Higher cost (>$100K) because it is a library expansion and reconfiguration
  - 2-3 day library outage that cannot be risked during the run
  - Gives us significant head room (16 or 24 drive bays, depending on the configuration)
- We do not plan to move beyond LTO8 yet.

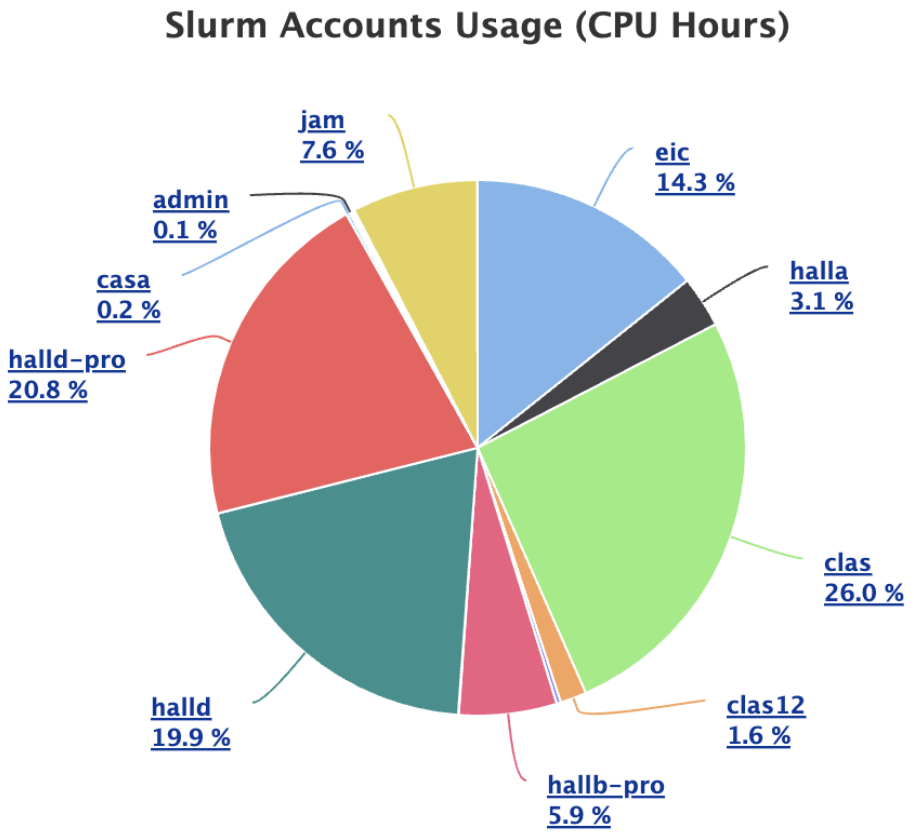
Lustre Update to latest long term stability release, 2.12.8

- Stable Since Upgrade – zero crashes
- Excellent Throughput
  - Typical 24 hour pattern below
  - 10GB/sec peaks are common.
  - Peaks nearing 20GB/sec observed



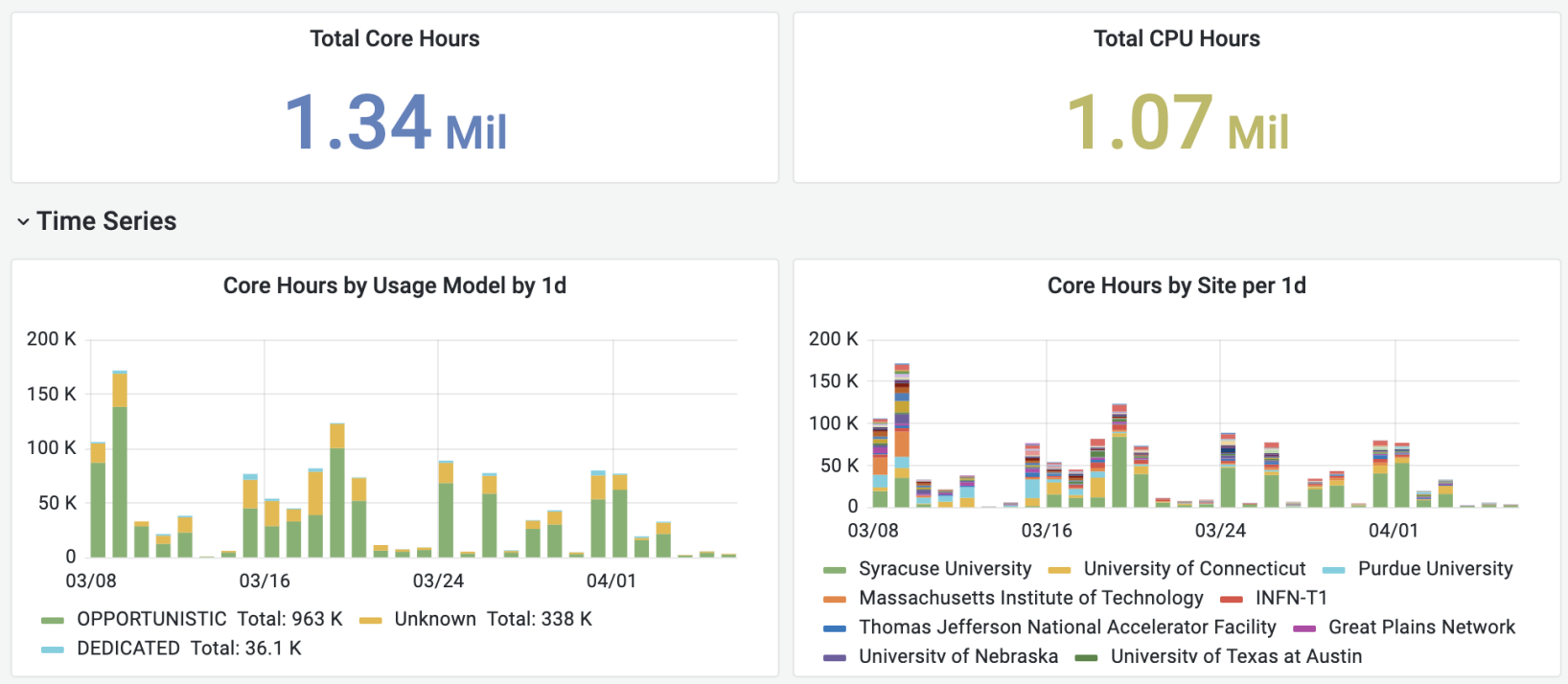
Farm Utilization this month

- Lustre stability has been good for the farm in recent weeks, and usage is high.
- Memory requests are the main reason for idle cpus, typically on older farm14 nodes.
- Swif2 and Slurm usage has gone well after the Auger and swif1 retirement
- Common causes of job inefficiency that we see
  - High Lustre metadata ops (more on this is a moment)
  - Not using /farm_out for log files (I/O wait for small IO)
  - Not using local /scratch for small I/O or high metadata ops



Open Science Grid Operations

- Operations have been smooth since the last round of upgrades
- We continue weekly operations meeting with OSG, so please let me know of any pain points
- Most recent month of data for GlueX + CLAS12 from gracc.opensciencegrid.org



Open Science Grid Development

- Two new submitter nodes are being installed: scosg220{1,2}
- Submitter nodes will be identically configured, but assigned to VOs by convention
  - One for CLAS12, one for GlueX, one for EIC.
  - This will avoid stepping on each other w.r.t. on-host resources
  - In the event of a host failure, moving to another is striaghtforward
  - scosg16 remains the development node.
- Upgrade to OSG 3.6 has been slowed a bit by OSG, but continues.
- Kick-off meeting with CILogon this week for the project to create a token issuer pilot.
- New configuration in slurm, OSG, HTCondor being setup for Moller.

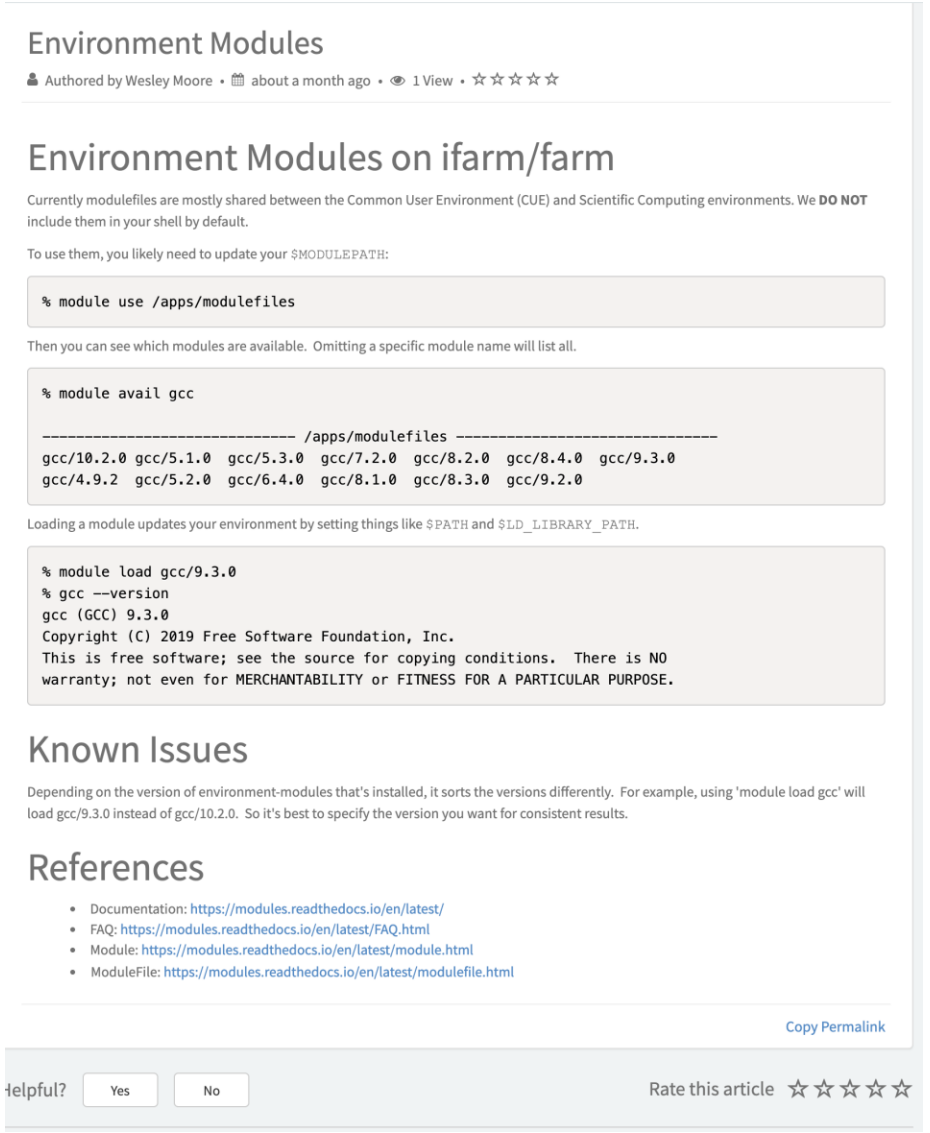
/work fileserver topics

- Snapshots
  - On smaller work areas, we sometimes get tickets about space not being freed up after large deletes because they are held in snapshots for some period of time (typically a week)
  - Snapshot schedules are configurable.
  - At some level they are good as a safety net, to save from accidental deletion
  - This does cause some confusion about space

- Storage Throughput
  - Newer ZFS based /work file servers have improved throughput, but do not match lustre in aggregate. We see workloads on Lustre that exceed /work capacity.
  - Because work is not a parallel file systems, at some point they can be overwhelmed by *n* farm nodes pointed at one fileserver (though they may handle fast metadata ops better)
  - Just for rough comparison: Theoretical network max 10GB/sec. Lustre19 Theoretical network max 120GB/sec.
    - The practical limit is (considering disk subsystems) is considerably less than that, but scales similary

Documentation in ServiceNow

- We are building Knowledge Base articles for common issues as they come up
- Emphasis is on focused articles rather than long manuals
- Writing KB articles for tickets that come up frequently
- KB articles support rating, feedback, which we see and can respond to
- Example
  - https://jlab.servicenowservices.com/kb?id=kb_article_view&sysparm_article=KB0014671



Environment Modules

Authored by Wesley Moore • about a month ago • 1 View • ☆☆☆☆☆

Environment Modules on ifarm/farm

Currently modulefiles are mostly shared between the Common User Environment (CUE) and Scientific Computing environments. We **DO NOT** include them in your shell by default.

To use them, you likely need to update your $MODULEPATH:

```
% module use /apps/modulefiles
```

Then you can see which modules are available. Omitting a specific module name will list all.

```
% module avail gcc

------------------------------ /apps/modulefiles ------------------------------
gcc/10.2.0 gcc/5.1.0  gcc/5.3.0  gcc/7.2.0  gcc/8.2.0  gcc/8.4.0  gcc/9.3.0
gcc/4.9.2  gcc/5.2.0  gcc/6.4.0  gcc/8.1.0  gcc/8.3.0  gcc/9.2.0
```

Loading a module updates your environment by setting things like $PATH and $LD_LIBRARY_PATH.

```
% module load gcc/9.3.0
% gcc --version
gcc (GCC) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Known Issues

Depending on the version of environment-modules that's installed, it sorts the versions differently. For example, using 'module load gcc' will load gcc/9.3.0 instead of gcc/10.2.0. So it's best to specify the version you want for consistent results.

References

- Documentation: https://modules.readthedocs.io/en/latest/
- FAQ: https://modules.readthedocs.io/en/latest/FAQ.html
- Module: https://modules.readthedocs.io/en/latest/module.html
- ModuleFile: https://modules.readthedocs.io/en/latest/modulefile.html

Copy Permalink

elpful? Yes No Rate this article ☆☆☆☆☆

More to Say: Topics for Future Meetings

- Two aspects of small files: Tape and Lustre.
  - Chris and I are developing a possibility for the former
  - We are investigating system improvements for the latter
- Multi-Factor Authentication
  - Separate meeting series to address DOE requriements
  - Planning for hallgw style access to ifarm
- Procurements
  - Farm Nodes (in progress)
  - Storage
- Rocky vs CentOS stream

Cutting Room Floor

Lustre Footnote: Metadata Example of Challenging Operations

- A recent example: a right loop that opens a file, writes 430 bytes, closes the file.
- This creates significant metadata ops for Lustre
- This slows the user job considerably
- In this case, it may have cost the user job ~30% of its CPU time
- An excellent case for local /scratch
  - State files
  - Database files
  - Debug files that can be copied off at the end of the job

```
open("./currentEvent.rndm", O_WRONLY|O_CREAT|O_TRUNC, 0666) =
7 fstat(7, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f078c7f0000

write(7, "mixmax state, file version 1.0\nN"..., 430) = 430
close(7)                                = 0

munmap(0x7f078c7f0000, 8192)            = 0

open("./currentEvent.rndm", O_WRONLY|O_CREAT|O_TRUNC, 0666) =
7

fstat(7, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f078c7f0000

write(7, "mixmax state, file version 1.0\nN"..., 426) = 426
close(7)                                = 0

munmap(0x7f078c7f0000, 8192)            = 0

open("./currentEvent.rndm", O_WRONLY|O_CREAT|O_TRUNC, 0666) =
7

fstat(7, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f078c7f0000

write(7, "mixmax state, file version 1.0\nN"..., 435) = 435
close(7)                                = 0

munmap(0x7f078c7f0000, 8192)            = 0
```