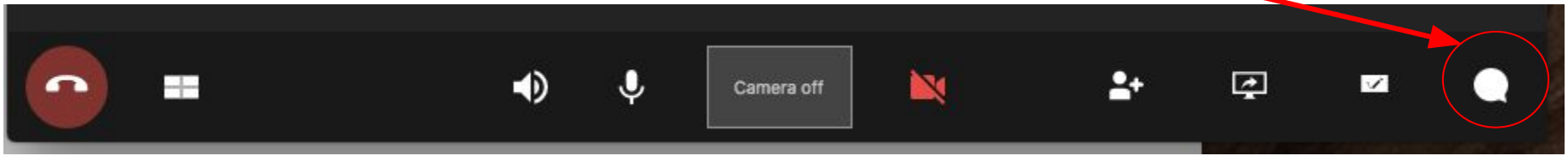
# WestGrid Town Hall: March 2019

Patrick Mann, WG Director of Operations
Lance Couture, HPC Sr Systems Administrator @SFU
Alex Razoumov, WG Training and Vis Specialist

# Admin

## To ask questions:

- **Websteam:** Email info@westgrid.ca
- **Vidyo:** Use the **GROUP CHAT** to ask questions.




Please mute your mic unless you have a question.

# Outline

1. Managing System Instability
   a. Causes of instability
   b. Best practices for dealing with instability
2. WestGrid & Compute Canada Updates
   a. Outages and maintenance.
   b. 2019 RAC Updates
   c. Updates: Arbutus upgrade, Béluga, ownCloud
   d. Reminders: Renewals, Orcinus
3. 2019 User Training

# Managing System Instability

Lance Couture
Lead HPC Systems Administrator for Cedar

1. Causes of instability
2. Best practices for dealing with instability

# Causes of Instability

1. Environmental
   - Power outages are a big one for SFU due to its location
   - We have large UPS, and generator for storage and network, not compute
2. Mechanical
   - In a system with > 1,600 compute nodes, > 60e3 cores , and > 30PB of disk - things fail!
   - How reliable is your car? Imagine using a Formula 1 car as your daily driver...
3. Software
   - Design of the system - optimized or general purpose
   - System software - e.g., Lustre (Storage), Omnipath (Fabric), SLURM (Scheduler), CVMFS
     - Lustre upgrades are sometimes untested b/c the possibility of new bugs outweigh current situations
     - Lustre versions, Omnipath versions, and combinations thereof
   - User land software - core heavy? Network heavy? Storage heavy? All have an impact

# Best Practices For Dealing With Instability

- Check point your data!
  - If you have large wall times, **checkpoint**
- Put data in proper locations
  - /home is for basic things - backed up
  - /scratch is for ephemeral calculations - not backed up, purged semi-monthly
  - /project is for data at rest - backed up
- If you want help with your job scripts, ask for our help
  - We can help optimize
    - Data locality
    - Script performance
    - Application best practices
    - General work flow

# User Perspective

Alex Razoumov
Training and Visualization Coordinator

Alex's slides

- What do you see when there are problems on Cedar?
- What can you do about these instabilities?
- Problematic workflows
- Other best practices

- Current maintenance
  - Cedar is down for system software updates (today only)
  - `/project` expansion March 1st - 4th, Lustre file metadata will be copied over to new SSDs, `/project` unavailable at this time, you can still use `/scratch` for running jobs during this time

- Cedar is a very complex system: lots of components, latest hardware and highly-optimized software (with small install base), shared filesystems, very broad user mix
  - $\sim 1,600$ nodes on Cedar, each with 24-48 cores, local storage
  - $\sim 66,000$ cores
  - 100Gb/s Omnipath interconnect linking all nodes and storage
  - three (Lustre) parallel file systems with $\sim 30$ PB combined storage
    - `/home`, `/scratch`, `/project`
    - each with its own policies, 2/3 backed up
  - 584 NVIDIA P100 Pascal GPUs
  - $\sim 60$ Slurm partitions, for long / short / GPU / large-memory / interactive jobs / CPU architecture
    - 3h, 12h, 1d, 3d, 7d, 28d maximum runtimes
    - trying to accommodate a large variety of job types
    - at the cost of efficiency and simplicity

- Our goals are to:
  - provide as much uptime as possible
    - we constantly monitor our clusters
    - work as quickly as possible to repair problems and return nodes to production
    - in case of downtime or other problems, provide frequent system status updates
  - accommodate a wide spectrum of jobs
  - maximize resource (CPUs, GPUs, memory, to smaller extent storage) utilization
  - minimize turnaround for your jobs

1. When hardware/etc problems occur, we want you to know how *in some cases* you can work around them
2. We want to show you how certain workflows can lead to problems on HPC clusters
   - and share with you best practices for working on these systems

# Major causes of system instability

## Node failures: a node needs rebooting or other work

- $\sim 1,600$ nodes on Cedar
- Of these, $\sim 30$ nodes have actual hardware failures at any one time
  - these get gradually replaced through a rather onerous return merchandise authorization (RMA) process
  - we are working with the vendor to simplify this process
- Marked offline by Slurm, for any number of reasons: not communicating, incorrect reports, low memory, cannot terminate the job, etc.
  - requires manual intervention
- Over-subscription of nodes, GPUs
  - e.g., too many threads
- Does not pass other checks and taken offline
  - GPUs get stuck in a strange state: *"Only EGL 1.4 and greater allows OpenGL as client API"*, requires reboot

# Major causes of system instability (cont.)

## File system problems

- Lustre object storage servers (OSS) can get overloaded with lots of small requests
  - example: this past Tuesday a user was running 90 jobs, all with high I/O in `/project` bringing it to a halt
  - putting these jobs on hold *did not* fix the system
  - one of the OSS servers had to be rebooted due to thread exhaustion (very heavy load requesting too many threads and eventually dead-locking)
  - end result: `/project` was not available to all users for $\sim$ 3 hours
- On Cedar we have:
  - 4 object storage servers handling `/home` (slow) and `/scratch` (fast)
  - 10 object storage servers handling `/project`
- These are paired into groups of two
  - one in a pair goes down $\Rightarrow$ the other one will take over, but high I/O jobs might take much longer than expected
  - both go down $\Rightarrow$ the entire filesystem will hang
  - any downed server will have to be rebooted

more on high I/O later ☞

# Major causes of system instability (cont.)

## Scheduler (Slurm) failures

- Can get overloaded with too many requests
- Bugs ...

more on scheduler later ☞

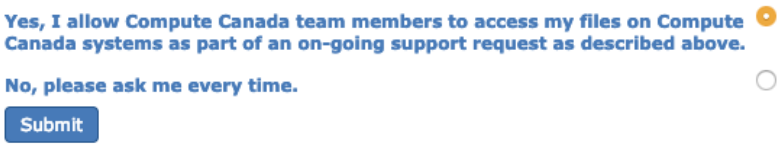
- No software stack synchronization between login and compute nodes
- Networking problems (within or outside our control)

# What do you see?

- Sluggish jobs
  - file system problem? node over-subscription? low memory? saturated network?
- Jobs not starting / taking unusually long to start
  - also valid reasons why your job's estimated start time could be pushed into the future
- Slurm not responding, or producing unusual output
  - e.g. last year's infamous Slurm bug leading to jobs stuck in 'Prolog' R (running) state for a long time, not producing any output
- Shell not responding to simple commands or very slow
  - could be per individual filesystem/command
- Output files missing from your working directory
- Inside running jobs see *"module not found"*
  - typically requires manual intervention
- Cannot log in

# What can you do about these instabilities?

- Pay attention to login messages (system's MOTD = message of the day)
  - terminal output from anything in your `~/.bash_profile` or `~/.bashrc` (e.g. when loading a module or activating a virtual environment ) might force important system messages scroll past the top of the terminal
  - these may contain both general system notices and `/scratch` purge notifications specifically for you
- Check `http://status.computecanada.ca` for updates and recent incidents
- Report problems to `support@computecanada.ca` with details:
  - system you are using
  - job IDs of affected jobs
  - detailed description of the problem, time/date it was first encountered
  - full path to one of the directories with the script and error files
    - check if you signed the consent that allow analysts to check your files (this will help resolve problems quickly instead of exchanging many emails), by logging in to `http://ccdb.computecanada.ca` and selecting My Account ➪ Agreements



# What can you do? (cont.)

- Sometimes you could work around a temporary filesystem problem by submitting jobs from another filesystem
  - on Cedar `/home`,`/scratch` files are handled by different servers than `/project` (may not be always possible: performance, input data)
- Do *not* delete and resubmit jobs that have been waiting in a queue for a long time until confirming with `support@computecanada.ca`
  - otherwise we can't analyze why a job is waiting
  - priority may be lost (grows slowly with the waiting time for each job)
- Expect a backlog of jobs after a system problem
  - do *not* swamp the system with a bunch of new jobs – be selective about what is most important to you
  - make sure that job parameters are chosen carefully to match the needs of particular jobs

# These workflows will create problems

- Running anything CPU-intensive on the head node

- Submitting large number of jobs
- Issuing too many requests to the scheduler
  - classical example: running `watch squeue ...` (never do this!)
  - using a script to submit thousands of jobs and then cancelling them
- Complex/unrealistic job dependencies can make Slurm unstable

- Not testing first on a small scale, and not scaling up gradually
  - large parallel jobs
  - many serial jobs and large job arrays
  - large computational problems in general
- Assuming perfect parallel scaling
  - your 64-core job may be slower than 32-core ...

# Problematic workflows (cont.)

- Excessive and/or "bad" I/O, i.e. anything resulting in high load on Lustre object storage servers
  - avoid lots of small reads/writes: many small files, frequent read/write in chunks smaller than 1MB, reading multiple small blocks from large files
- Storing a large number of small files
  - Lustre is very different from your laptop's drive
  - organize your code's output
  - use tar, or even better dar (`http://dar.linux.free.fr`, supports indexing, differential archives, encryption)

- Using nested parallelism in black-box pipelines
  - e.g. submitting serial jobs each of which launches multiple threads, sometimes asking for all cores on a node
  - your pipeline should be adapted to the cluster; if not sure, please talk to us

# Problematic workflows (cont.)

- Using `mv` command to move files `/home,/scratch` → `/project` will result in an overquota error message in the middle of moving
  - this is expected behaviour!
  - not so much a problem for the cluster, but certainly will be a problem for you ...
  - in `/project` the 1TB (or higher) quota is applied to all files with the group ID `def-group`
    - so that all your group members are able to write there
    - any new file you write to `/project` will have `def-group` group ID
    - you can find this group ID by running `id` and looking for 'def-...'
  - by default, all files in `/home,/scratch` have group ID `username`
  - `mv` command preserves group ID, i.e. effectively `mv` acts as `cp -a`
  - the quota for group ID `username` is almost zero in `/scratch`
  - solution: use `cp` instead, followed by `rm`

# Other best practices

- Implement/use checkpointing to be prepared for system failures
- Break your job into pieces, if possible (time-wise, processor-wise)
- Read the documentation about scheduling, running jobs, using modules, other topics ✍ `https://docs.computecanada.ca`
- Know as much as possible about your application (serial vs. parallel), and how it was parallelized (threaded vs. MPI)
  - very important for creating the correct job submission script!
- Start with some tests before running extensive simulations
  - estimate the resources (especially memory, wall time)
  - use `sacct` or `seff` to estimate your completed code's memory usage
  - test parallel scaling, scaling with problem size
- Only request resources (memory, running time) needed
  - with a bit of a cushion, maybe 115-120% of the measured values
  - otherwise your job will be queued much longer

# Other best practices (cont.)

- If you still need to do lots of small I/O from inside your job:
  - *use on-node SSD:* Slurm-generated directory `$SLURM_TMPDIR` points to `/localscratch/${USER}.${SLURM_JOBID}.0`
    - for both input and output
    - don't forget to move files out before your job terminates: everything in `$SLURM_TMPDIR` will be deleted
  - *use RAM disk:* `$TMPDIR` points to `/tmp`
    - don't forget to allocate additional memory to your job
    - don't forget to move the results before your job terminates
- Port your workflow to another CC's general-purpose cluster, to run it there in case of failures
  - data management part may not be so easy, but Globus should help
  - also try to port your workflows (have accounts, appropriate input data, programs installed) to local clusters where available (Grex, Orcinus, Plato)
- If you received a `/scratch` purge warning, do *not* wait until the last minute to transfer data to local systems or other clusters
  - always pay attention to `/scratch` purge notices (email, system's MOTD)
  - exercise care when transferring data close to quota in destination
  - when moving to `/project`, replace `mv` with `cp + rm`

# Other best practices (cont.)

- Be aware that some filesystems are not backed up (e.g. `/scratch`), and some have a purge policy (`/scratch`) ✍ have a backup plan
- If a file's path changes, our backup system will interpret it as a new file ⇒ unnecessary load on the filesystems
  - be careful with renaming large directories in `/home` and `/project`
- In general, do *not* run jobs in `/home`
  - slow, not designed for high performance (unlike `/scratch`)
  - small quota (50GB/user)
  - lots of I/O makes difficult to do backups
- After your job finishes:
  - clean up (remove files that are no longer needed)
  - compress large files to reduce the disk space usage
  - archive (tar) the directories with many small files to reduce the file count
  - eventually move your data from `/scratch` to `/project`, `~/nearline` (will be available on Cedar soon), your own storage

# WestGrid and Compute Canada Updates

**Patrick Mann**
**Director of Operations, WestGrid**

# Outages and Maintenance

| | | |
|---|---|---|
| **Graham** | Feb.11, 2019 | `/project` filesystem. Metadata server crashed and was rebooted. |
| **Cedar** | Mar 1, 2019 | Major outage for Lustre Metadata server upgrade (*see next slide*) |
| | Feb 15, 2019 | More filesystem issues. Very high load on Lustre. |
| | Feb 8, 2019 | Globus Data Transfer Node (DTN) was out for a few hours for planned maintenance. |
| | Jan 27, 2019 | Power outage at SFU. All nodes went down and jobs lost. |
| | Jan 17, 2019 | 10 minute network outage for network hardware maintenance by upstream provider. |
| | Jan 2, 2019 | `/project` metadata crash and error condition prevented metadata device from remounting. |
| **Niagara** | Jan 15-16 | 2 day scheduled shutdown to prepare for emergency power generator and larger UPS. |
| **Arbutus** | Feb 4, 2019 | (almost) all remaining west.cloud instances were moved to the new arbutus cloud. West.cloud was deactivated. (*details in next slides*) |

# Cedar Mar.1 Outage

- Moving `/project` Lustre Metadata servers from old SAS drives to new SSD's.
  - Have to copy the metadata from one to the other using specialized "tar" approaches.
- Upgrade Lustre from 2.10.1 to 2.10.6
- Upgrade to CentOS 7.6 with corresponding OPA drivers

| | |
|---|---|
| Friday Mar 1, 2019 | OS and Lustre updates. Begin Lustre metadata migration. |
| Saturday Mar 2, 2019 | Bring cedar up **without `/project`** (Lustre metadata migration still in progress)<br>● Jobs can be submitted from `/home` or `/scratch` |
| Wednesday Mar 6, 2019 | ● Complete metadata migration and acceptance testing.<br>● Remount `/project` (live), and back to normal operations. |

Jobs submitted before outage will stay in the queue and will be started after the downtime.

- Since the system will **not have `/project`**, make sure that such jobs can run without `/project`.
  - Otherwise those jobs would fail (resubmit when `/project` is remounted).

# Arbutus Upgrade & Migration

Feb 4/2019: **ALMOST DONE**

- All projects have been moved to Arbutus (except for a few special external projects).
- **west.cloud hardware is being deactivated and moved to Arbutus.**

**Massive upgrade!**

- Additional ~1,400 compute cores and ~3.5PB useable storage, 2 new DBaaS nodes
- Updated OpenStack with advanced provisioning capabilities.
- New monitoring/alerting infrastructure
- **Increased performance, scalability and stability**

Upgrade/Migration information

- https://docs.computecanada.ca/wiki/Arbutus_Migration_Guide
- https://docs.computecanada.ca/wiki/Arbutus_West_Cloud_Upgrade

New URL for cloud platform access:

- https://arbutus.cloud.computecanada.ca

# Reminder: Orcinus Defunding

## Usual reminder: **Orcinus defunding date: Mar 31/2019**

- UBC is planning to keep Orcinus going
  a. Until the WestGrid Network and LDAP are decommissioned (June 2019???).
  b. Opportunistic use (no allocations) to all existing WestGrid users.
  c. Preference will be given to UBC-based users
  d. No new users added
- Contact (roman@chem.ubc.ca) if you need access to orcinus post WestGrid era.

## Best Effort: Users responsible for their data

- No software updates
- Users should keep their own backup copies
- No plans beyond the summer so users should consider migrating
- Data can be moved to other CC sites (cedar, graham, beluga) `/project`
- Most users have an allocation. Check with "support@westgrid.ca".

# Béluga Update

- Power fluctuation issue traced to faulty power supplies. All power supplies replaced!
- Handed over to CQ Tuesday Feb.12.
  - CQ currently running acceptance tests.
- So significantly delayed (almost 2 months now)
  - After formal acceptance will be configure to CC standards.
  - Then test runs, application install, etc.
- **So April and RAC 2019 schedule is at risk.**
  - Further announcements re RAC 2019 allocation implementation will be forthcoming as soon as we know more.

# Resource Allocation Competition 2019

## RAC 2019 deliberations have been completed.

- Science and Technical reviews completed.
- Science committees have met and scored the proposals.
- Final allocations have been made.

**Some preliminary (unofficial) stats:**

**Notifications scheduled to go out Mar.15.**
*Stats will be available on CC web site sometime after that.*
*Implementation early in April.*

| Allocation | Ask vs Available | Comments |
|---|---|---|
| Storage | ~1.65x | New storage resources are being installed.<br>● 10 PB for Cedar - another outage in the next couple of months! |
| CPU | ~2.6x | Scaling function. But overall only able to satisfy <50% of requests. (includes Béluga) |
| GPU | ~5.25x | Drastic cuts. Only requests with very well-justified and specific asks. |
| Cloud CPU | ~1.16x | Some scaling. |
| Cloud storage | ~0.84x | Under. |

# Renewals 2019

As usual annual renewals will be requested next month.

- Every user will be asked to renew their accounts.
- PIs will be asked to update their research contributions (CCV)
- Expected opening Apr 8 with May 6 deadline

Detailed instructions for renewals 2019 should come out shortly.

Note: this is an important exercise both to clean out our accounts database and for our stats reporting to funders.

# WestGrid OwnCloud

Owncloud is currently run as a WestGrid service.

The plan is to move the WestGrid service to a Compute Canada service on Cedar

- Will make use of the new Cedar cloud partition (currently being installed).
- Some of the cloud nodes will be on redundant power with diesel backup (cloud storage is already there).

No dates yet - need the cloud partition to be in production. Probably later in the summer.

- Data migration may cause issues with some users. *We'll be in touch when we're ready.*

# Cedar Nearline

Still working on Nearline functionality.

- Recall: copy files to `/nearline/..` and then the files are auto-migrated to tape.
- Currently working on network issues for the replication between SFU, Waterloo and Toronto.
- Still aiming for RAC 2019 (April)

# FPGA's on Graham

Graham now has some FPGA's available for use

- Provided by Huawei as a value-add in the original contract.

Ask support@computecanada.ca if you're interested.

# Upcoming User Training

Alex Razoumov, WestGrid Visualization & Training Coordinator

# WestGrid Online Sessions

## WestGrid's bi-weekly webinars

- Every second Wednesday 10am Pacific / 11am Mountain / 12pm Central
- Additional, alternate-week webinars may be announced as we go along

**Up-to-date schedule and links**
**http://bit.ly/wg2019a**

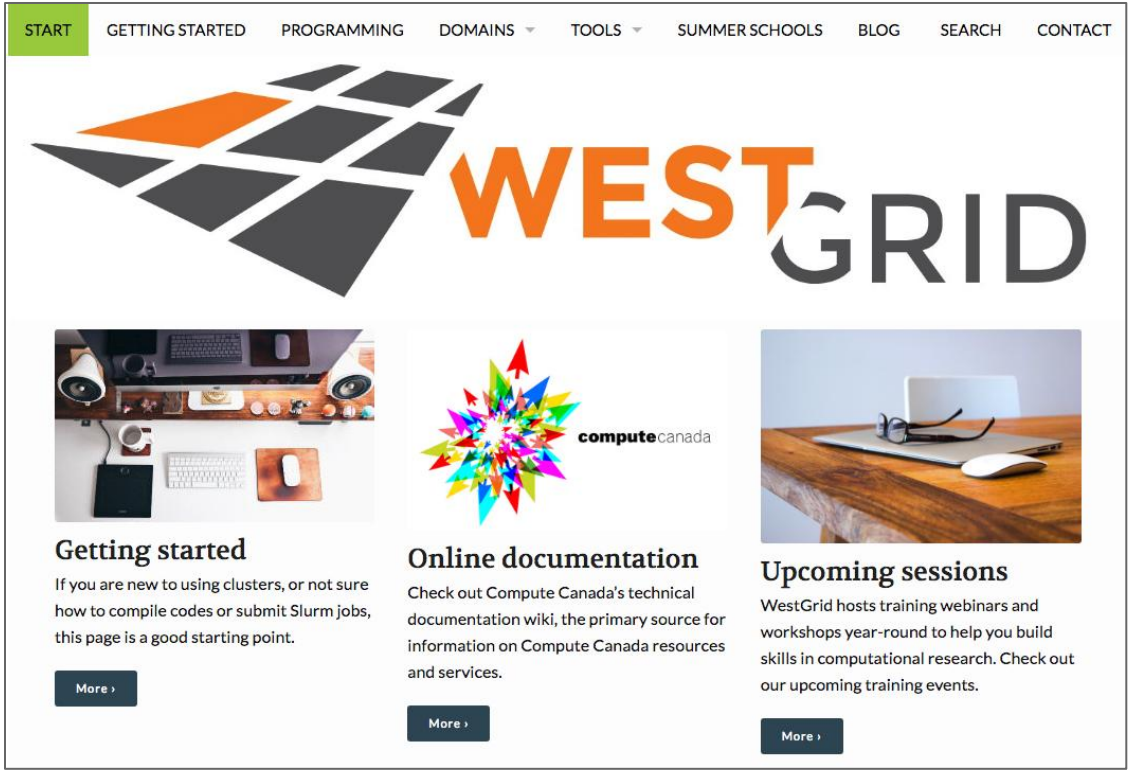
| | | |
|---|---|---|
| Mar-06 | *Molecular visualization with VMD* | Dmitri Rozmanov |
| Mar-20 | *Research Data Management Tools, Platforms, and Best Practices for Canadian Researchers* | Alex Garnett (SFU) and Adam McKenzie |
| Apr-03 | *Next-generation sequencing* (exact topic TBA) | Phillip Richmond (UBC) |
| Apr-17 | *Distributed unstructured meshes and graphs in Chapel* | Alex Razoumov |
| May-01 | *Programming best practices* (exact topic TBA) | Chris Want |
| May-15 | *Julia language for data analysis* | John Simpson |

# In-person Training

| | | |
|---|---|---|
| Mar-18 to Mar-21 | **Research Computing Workshops:** ***Introduction to Scientific Computing with Linux, High-Performance Computing, Parallel Programming, and Scientific Visualization*** | SFU |
| Apr-30 | **Using yt for analysis and visualization of volumetric data in Python** | SFU |
| May-27 to May-30 | Research Computing Summer School | University of Calgary |
| June (date TBA shortly) | Research Computing Summer School | University of British Columbia |

**Up-to-date schedule and links**
**http://bit.ly/wg2019a**

# User Training Archive




**https://westgrid.github.io/trainingMaterials**

- Recently added:
  - *Memory debugging with Valgrind* (slides)
  - *Data analysis with YT* (slides and video)
  - *Text parsing and matching in HPC* (slides)
- Links to other guides, documentation & upcoming events

# Questions?

# Support

## Contact us anytime:

support@westgrid.ca

www.westgrid.ca

docs.computecanada.ca

Any issues or problems? We can advocate for WG member and user concerns within Compute Canada.

# Extra Slides

System Instability extra slides.

# System Instability

## Complex systems with lots of components.

- Per-component fault rates are small
- with 100's or 1000's of nodes there is a high probability that some components will be out.
- Plus the usual software bugs and issues.

## (B)leading-edge

- Latest versions of hardware (CPU's, GPU's, FPGA's, Omnipath interconnect, ..)
- Highly-optimized software (emphasis on performance not reliability)
- Small installed base so bugs are not identified quickly.

## Petabyte-scale Shared Filesystems

- Single point-of-failure - everyone and everything needs the filesystem!
- Drives fail regularly - need complex redundancy schemes.
- And again performance is paramount - all nodes access the shared filesystem.
- And users can hammer the filesystem
- And Cedar in particular has a huge storage system running on the very new Omnipath interconnect.

## Very Broad User Mix

- GPUs, fat nodes, runtimes from short to very long, extremely large number of apps
- very experienced to very inexperienced users
- Very open (shared) system - users can actually effect system stability!
- Big job for scheduler - things like watching "squeue" can push the scheduler over the edge.

**External Effects**

- Power outages in particular
- Only key infrastructure is on UPS
- Building maintenance.

# Example: Cedar State

*Cedar node state reported by slurm (`sinfo`) as of Feb.28, 08:16*

| #nodes | Error | Comment |
|---|---|---|
| 3 | Local disk errors | |
| 4 | Bad OPA or OPA showing errors | Cable issues, dropped connections, .. |
| 15 | Dead or reporting hardware issue | Firmware, didn't boot, epilog, ... |
| 8 | GPU reporting errors | |
| 7 | Memory errors | |
| 3 | Miscellaneous | |
| 9 | "Kill task failed" | Unexplained Slurm issue. Cedar team working with SchedMD. |
| 14 | "Unable to establish IPMI connection" | Management Interface. |
| 31 | "Reserve island g1-10-2 for cloud deployment" | Currently reserved for cloud deployment |

| | |
|---|---|
| Total nodes | 1,612 |
| Reserved for cloud | 31 |
| Available | 1,581 |
| **Down** | **63 (4%)** |

# Summary

Nodes usually replaced by vendor. Software issues can be cleaned up.

- Can take a while.
  - Very tedious process: Cedar team working with Dell to streamline the process.
- But a few nodes will always be failing or otherwise not responding.

**Continous: issues replaced/fixed and other nodes go down or software explodes.**

- **Users should be prepared: minimize the effect of such issues.**

"Kill task failed" needs time-consuming manual cleanup by sysadmins.

- Nothing wrong with the node
- Does not effect jobs as the auto-clean occurs after a job has finished.
- Cedar team working with SchedMD (Slurm).

# Complexity of a Supercomputer

**(See patrick's backup slides at the end of the presentation).**

- reliability is a big problem, lots of parts, very general-purpose cluster (single-purpose setup would make it much easier); we are all to all people
- car analogy: pressing on accelerator of a new car non-stop for 5 years
- show users the extent of modifications and optimization that some people do in extreme cases to make their codes faster
- policies on Niagara: short runtimes => lots of problems will go away
- lots of technologies we regularly use are still in their infancy; very small installation base (how many do run with OPA and 60e3 cores); lots of bugs; Lustre is very telling (Lance's example)
- our explanations should not sound like excuses