Field Notes 7

How to make the most of Slurm, and avoid common issues

Jason Booth

Slurm User Group 2023

Field notes - tradition

Continuing to build on last year's field notes

https://slurm.schedmd.com/publications.html

Purpose

To show you how to get the most out of your support experience and point you in a better direction while working with Slurm.

Field Notes - Agenda

- Short History of SchedMD and Random notes, observations and configuration preferences
  - History
  - System requirements
  - Upgrading
    - We target current releases for bug fixes
  - Node Addition and Removal
  - Configless
  - Storing job scripts (improvements)
  - Slurmdbd purge, Log rotation, archiving

Field Notes - Agenda Continued

- Agenda continued
    - Scalability
        - Fast statesave
        - Separate thread for slurm commands
        - Munge threads
        - Identity management and nss_slurm
    - A note on cgroups

Field Notes - Agenda Continued

- Agenda continued
  - Random notes
    - Coredumps
    - Debugging
    - A word on support and training
      - Let us help you

History

A Brief History - Slurm

- The use of “Slurm”, not SLURM or any other variation is preferred.
- **S**imple **L**inux **U**tility for **R**esource **M**anagement (Historic)
- Slurm in all capitals describes earlier days of the software when Slurm was just a resource manager.

A Brief History - SchedMD

- 2010 - SchedMD opened up as a part time custom Slurm development shop
  - 100% Owned by Danny Auble and Morris Jette
- 2011 Moe and Danny went all in
- 2014 Hired the first non-developer
- 2016 SchedMD relocated from Livermore CA to Lehi UT

A Brief History - SchedMD

- 2017 11 Employees
- …
- 2023 31 Employees that support, train, develop and plan.

A Brief History - SchedMD Goals

- Provide amazing commercial support for Slurm
- Keep Slurm dominant on the Top 100 systems
- Keep Slurm open source

A Brief History - We are hiring!

Visit : https://www.schedmd.com/careers.php

or

Email resume to : jobs@schedmd.com

System Requirements

System Requirements

- Hardware
  - Fewer faster cores on the slurmctld host is preferred
  - Fast path to the StateSaveLocation
    - IOPS this file system can sustain is a major bottleneck to job throughput
      - At least 2 directories and two files created per job

System Requirements - Continued

- Hardware
  - Fast path to the StateSaveLocation
    - …
      - The corresponding unlink() calls will add to the load
    - Use of array jobs instead of individual job records will help significantly, as only one job script and environment file is saved for the entire job array.

System Requirements - Continued

- Hardware - example minimum system requirements ~ 100k jobs a day / 500 nodes.
  - 16 GB RAM
  - Dual core CPU with high clock frequency
  - Dedicated SSD or NVME (statesave)
- The amount of RAM required will increase with a larger workload / node count.

System Requirements - Continued HA

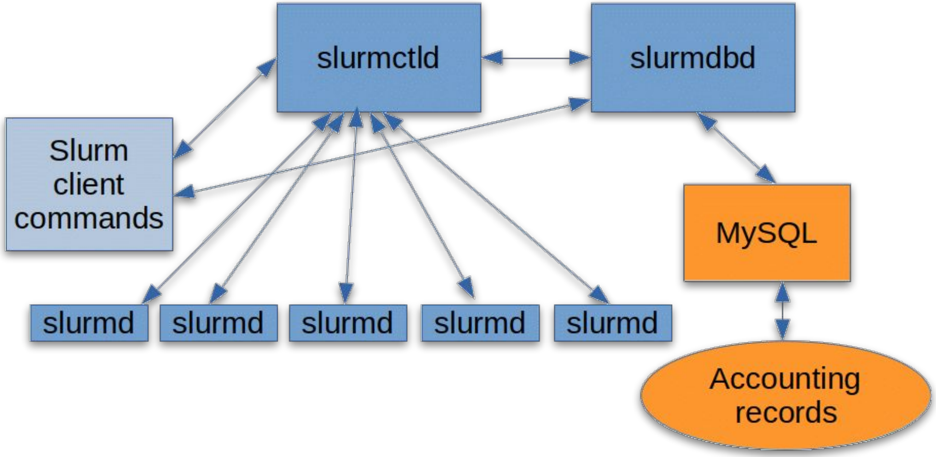
- Slurmctld
  - The state save directory must be mounted on both controllers.
    - The state save should not be local storage on either of the slurmctld hosts or local to one slurmctld and serviced out to the other slurmctld.
    - The state save should not share IOPS with user/job storage!

System Requirements - Continued Slurmdbd

- Slurmdbd - example requirements
    - Hardware - minimum system requirements
        - 16-32 GB RAM
            - The RAM requirement goes up in relation to the number of jobs you wish to store/query.
        - CPU requirements are not a picky as Slurmctld
        - Dedicated SSD or NVME for the database

System Requirements - Management network

- Slurmctld, slurmdbd and slurmd traffic should be on a dedicated network
- Should not share the same “twisted pair” network or bandwidth as user storage/data
- For most sites, a simple flat network is recommended
    - e.g. 192.168.0.1/24



System Requirements - Continued network

- By default, the slurmctld will listen for IPv4 traffic. IPv6 communication can be enabled by adding EnableIPv6 to the CommunicationParameters in your slurm.conf. With IPv6 enabled, you can disable IPv4 by adding DisableIPv4 to the CommunicationParameters. These settings must match in both slurmdbd.conf and slurm.conf

System Requirements - Continued network

- If you do have nodes that are in separate networks and are associated with unique switches in your topology.conf file, it's possible that you could get in a situation where a job isn't able to run. If a job requests nodes that are in the different networks, either by requesting the nodes directly or by requesting a feature, the job will fail because the requested nodes can't communicate with each other. We recommend placing nodes in separate network segments in disjoint partitions.

System Requirements - Continued network

- Further information can be found on our website
  - https://slurm.schedmd.com/network.html
  - https://slurm.schedmd.com/topology.html

Upgrading

Upgrading - What version are you on?

- By show of hands
  - 23.02
  - 22.05
  - 21.08
  - 20.11
  - No need to raise your hand.
    - Something older?

Upgrading - Reasons to Upgrade

- We target the most recent release for bug fixes
- For SchedMD supported customers, support contract requires staying on a current release
- You get to take advantage of performance improvements and new features

Upgrading - Continued

- There is a specific sequence to use when moving between major Slurm releases.

Upgrading - Continued

slurmdbd
>=
slurmctld
>=
slurmd
>=
slurmstepd
>=
client
commands

Must stay within 3 major releases.

E.g. {23.02, 22.05, 21.08} is okay,
but {23.02, 22.05, 21.08, 20.11} is not.

Upgrading - Continued

- Within each major release, you can mix the maintenance release versions without issue.
  - E.g. {23.02.0, 23.02.1, 23.02.2} is okay

Upgrading - Continued

- RPMs do make this process difficult to do with the system live.
- While we ship and support the slurm.spec file, we do not actually recommend using RPMs to install Slurm.
- We suggest structuring installs in version-specific directories, and using symlinks and/or module files to manage versions.
- This makes rolling upgrades much simpler.

Upgrading - Continued

```
# ./configure --prefix=/apps/slurm/23.02.5/ --sysconfdir=/apps/slurm/etc/
# ln -s /apps/slurm/23.02.5    /apps/slurm/dbd
# ln -s /apps/slurm/23.02.5    /apps/slurm/ctld
# ln -s /apps/slurm/23.02.5    /apps/slurm/d
# ln -s /apps/slurm/23.02.5    /apps/slurm/current
```

Use the appropriate symlink in each service file,
and add /apps/slurm/current symlink into $PATH
(through /etc/profile.d/ or a module file).

This makes a rolling upgrade much simpler, just
move the symlink when ready to move that component
forward onto the newer release.

Upgrading - Continued

- Backing up the MySQL database used by slurmdbd is strongly encouraged before upgrading.
    - You should probably be doing this already as part of a regular backup strategy, but this would be a good time to make sure it works.

Upgrading - Continued

- continued...
  - For larger databases, or more unusual systems, you may want to test the upgrade/conversion on a copy of the full production database on a separate machine.
  - Older MySQL versions (5.5 and before) have had problems with the conversion process.

Upgrading - Continued

- slurmdbd will automatically convert the MySQL schema.
    - This can take ~10-15 minutes or more, depending on the size of the database.
    - Taking a backup of StateSaveLocation is also recommended.
    - Once a daemon has been upgraded, you cannot roll back to a prior major version without loss of data and your job queue

Upgrading - Continued

- Recommended inode_db values during upgrading
  - Innodb_buffer_pool_size - somewhere between 5 and 50 percent of the available memory.
- A note on /tmp
  - Make sure /tmp has enough room and does not fill up!

Upgrading - Continued

- Keep in mind that…
    - With major OS releases, subtle things may report or behave differently.
        - kernel memory, libraries (pmix), database, cgroups.
- Examples
    - “free -m” may report different memory values between Kernel versions.

Upgrading - Continued

- Examples continued ...
  - Cgroups V2 or hybrid mode may be enabled.
    - Cgroup v2 hybrid mode is not supported
    - Though, you can have nodes on running v1 and others on v2
  - Differences between MySQL/MariaDB versions.

Upgrading - Continued

- MaraDB / MySQL warning
    - Before MariaDB 10.2.1, BLOB and TEXT columns could not be assigned a DEFAULT value. This restriction was lifted in MariaDB 10.2.1.
    - MariaDB note (MySQL 5.5 was introduced in 2012 and MariaDB10.5 in 2019) with a clear change in DEFAULT value handling)

Upgrading - Continued

- MaraDB / MySQL warning
  - In the past, we asked that you contact support if you are migrating from MySQL 5.5 to MariaDB or upgrading MariaDB >= 10.2.1 from an older version.
- Code to automatically fix these issue was introduced in 22.05.7 and 23.02

Upgrading - Continued

- If you are on an older version of Slurm then you might see symptoms like the following
  - sacctmgr: accounting_storage/slurmdbd: acct_storage_p_remove_assocs: No error Nothing deleted
  - Can not delete associations.
- Please upgrade to solve these problems or contact support.

Upgrading - Continued downgrading

- **There is no way to downgrade the state save files or the database schema once you upgrade!**

Upgrading - Continued downgrading

- It is possible to restore from a backup:
    - If there were no running jobs
    - You are restoring the state save files from the pre-upgrade backup
    - You are restoring a pre-upgrade snapshot of database

Upgrading with different slurmctld versions

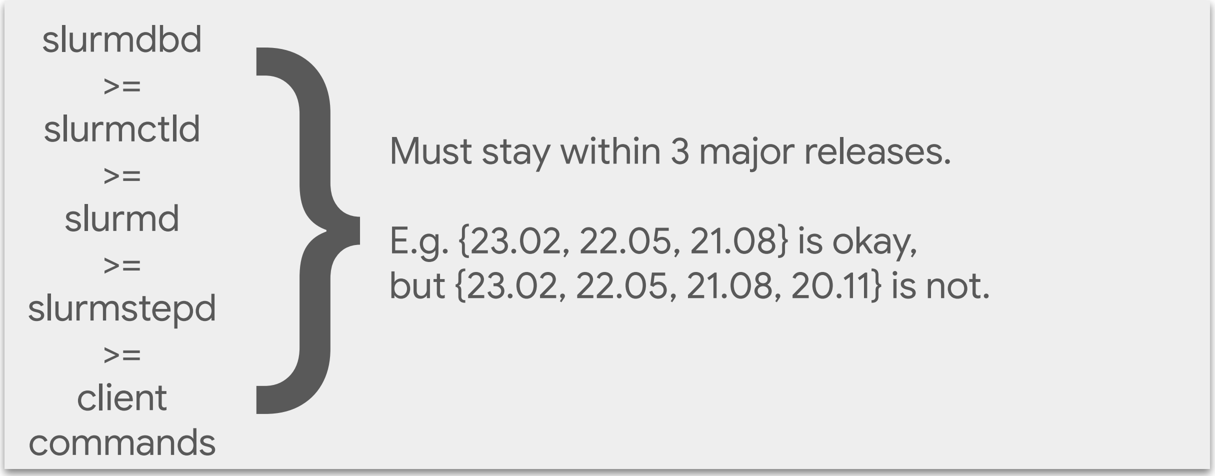
- And submitting jobs between two different clusters

Upgrading - Continued mixing slurmctld versions

- Worthy mentions regarding upgrades
  - Example:
  - Mixing 22.05 and 21.08 slurmctld's and submitting to multiple clusters sharing 1 slurmdbd
    - This is something new we are seeing and here are some guidelines

Upgrading - Continued mixing slurmctld versions

- The slurmdbd needs to be greater than or equal to the highest version slurmctld (slide 25)
- Stays within the supportability matrix



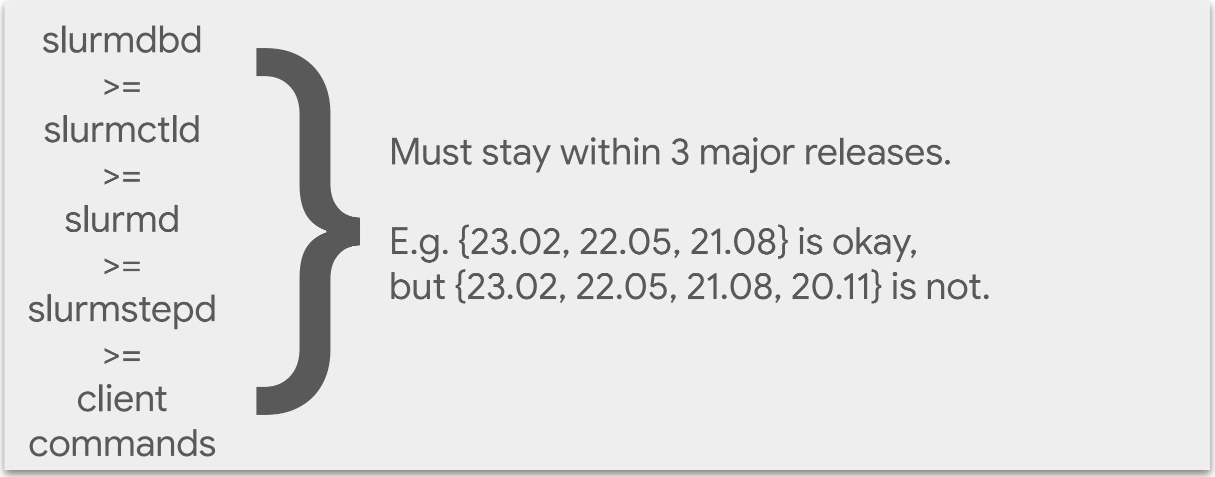
slurmdbd
>=
slurmctld
>=
slurmd
>=
slurmstepd
>=
client commands

Must stay within 3 major releases.

E.g. {23.02, 22.05, 21.08} is okay,
but {23.02, 22.05, 21.08, 20.11} is not.

Upgrading - Continued mixing slurmctld versions

- Furthermore, use older clients on both systems if you wish to submit between these two clusters that share a slurmdbd

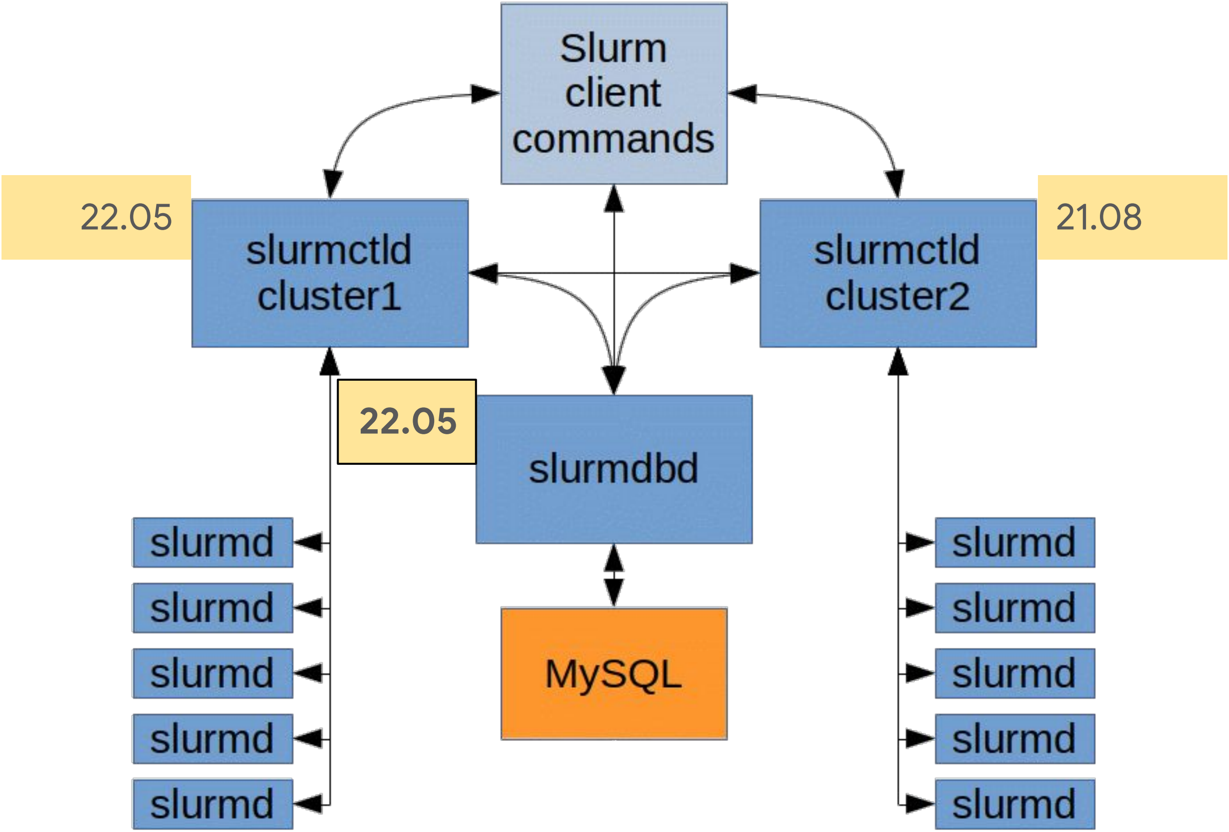


slurmdbd
>=
slurmctld
>=
slurmd
>=
slurmstepd
>=
client commands

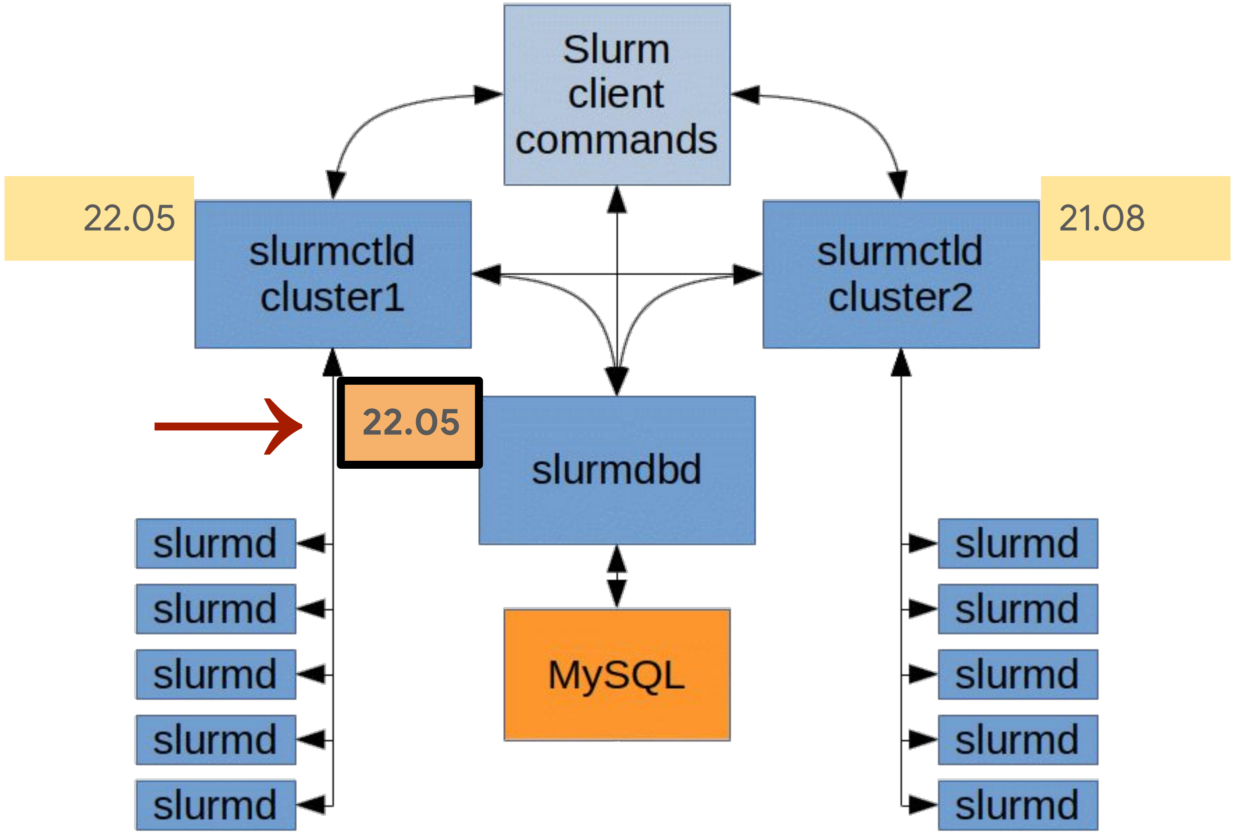
Must stay within 3 major releases.

E.g. {23.02, 22.05, 21.08} is okay,
but {23.02, 22.05, 21.08, 20.11} is not.

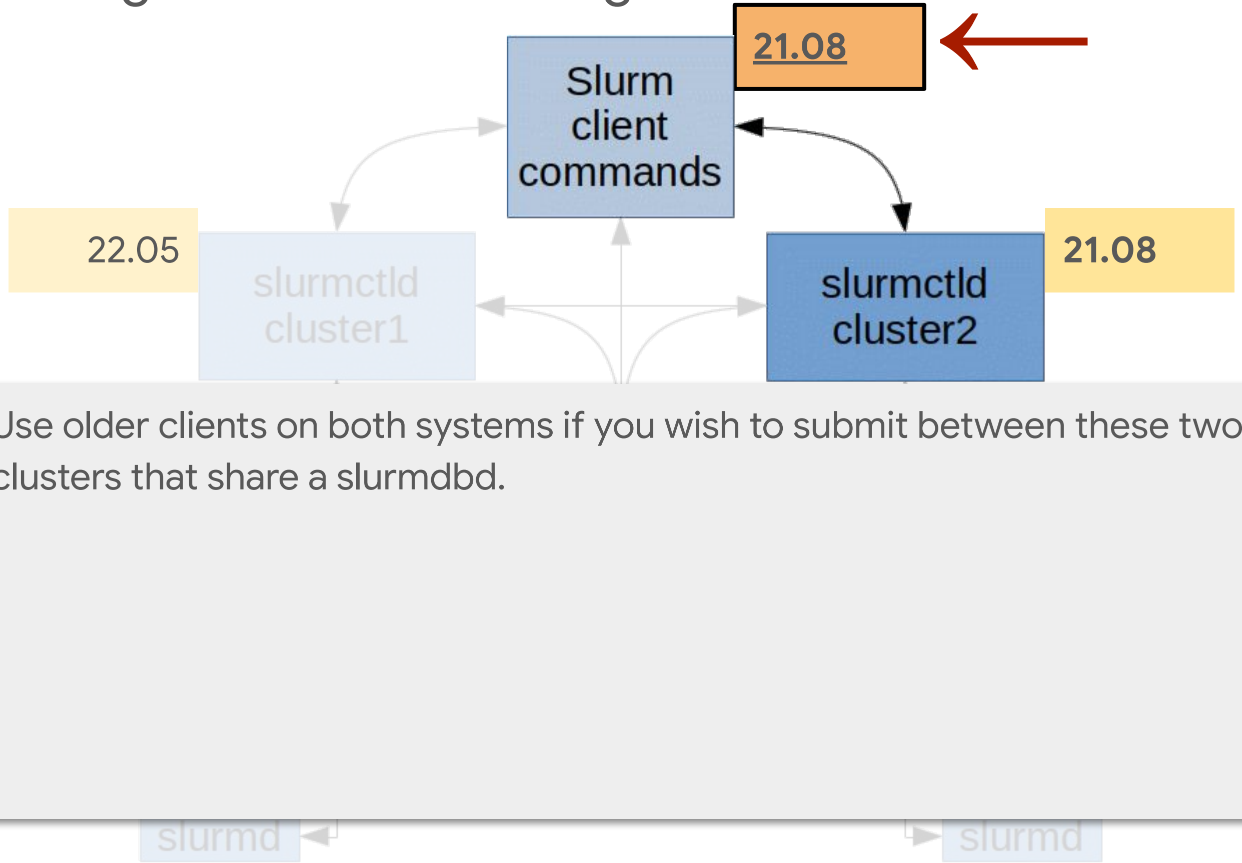
Upgrading - Continued mixing slurmctld versions



Upgrading - Continued mixing slurmctld versions

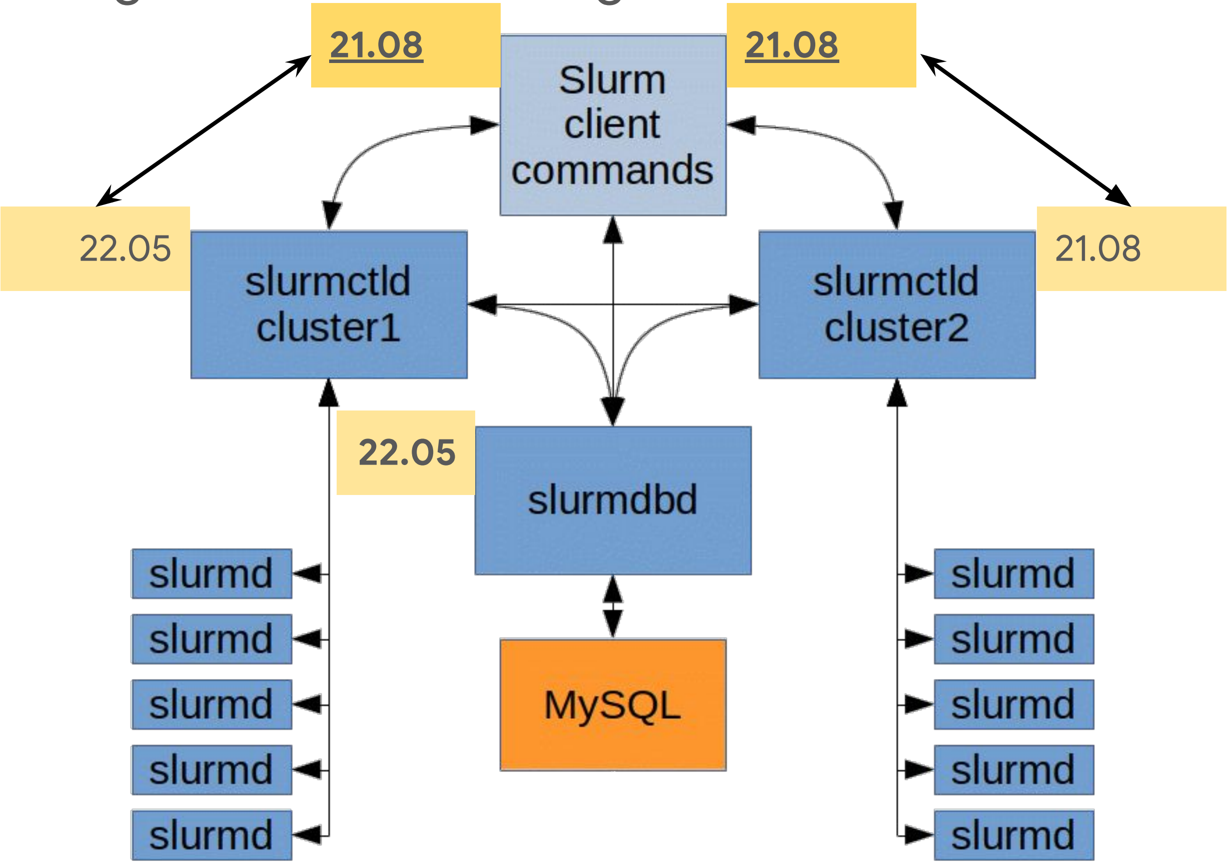


Upgrading - Continued mixing slurmctld versions



- Use older clients on both systems if you wish to submit between these two clusters that share a slurmdbd.

Upgrading - Continued mixing slurmctld versions



Node Addition and Removal

Node Addition and Removal

- Adding and removing nodes in Slurm is a sensitive operation.
  - This seems to cause problems for each site at least once early on.
- Certain internal data structures are built off the node list at startup, and are used within the communication subsystems.

Node Addition and Removal

- Changing the Node definitions, and restarting only the slurmctld, will usually lead to communication errors as messages are misrouted internally.

Node Addition and Removal

Safe procedure:

1. Stop slurmctld
2. Change configs
3. Restart all slurmd processes
4. Start slurmctld

Node Addition and Removal

Less-Safe, but usually okay, procedure:

1. Change configs
2. Restart slurmctld
3. Restart all slurmd processes really quickly

Node Addition and Removal

1. We do have plans to make this less painful long-term and some options today to help with this process (dynamic nodes).
2. The new cons_tres plugin has split some of these data structures apart, and will eventually let us change this.
3. This is blocked until cons_res is removed.

Node Addition and Removal - Dynamic nodes

- Available since 22.05
- Nodes can be created two ways

1. Through slurmd parameters
   a.
   ```
   #> slurmd -Z --conf "RealMemory=80000 Gres=gpu:2 Feature=f1"
   ```
2. Though scontrol
   a.
   ```
   #> scontrol create NodeName=d[1-100] CPUs=16 Boards=1
   SocketsPerBoard=1 CoresPerSocket=8 ThreadsPerCore=2
   RealMemory=31848 Gres=gpu:2 Feature=f1 State=cloud
   ```

- https://slurm.schedmd.com/dynamic_nodes.html

Node Addition and Removal - Dynamic nodes

- MaxNodeCount=#
- SelectType=select/cons_tres
- TreeWidth=65533

1. Set to the number of possible nodes that can be active in a system at a time. See the slurm.conf man page for more details.
2. Dynamic nodes are only supported with cons_tres.
3. Fanning out of controller pings and application launches through slurmds are not supported with dynamic nodes. TreeWidth must be disabled (i.e. set to 65533) for dynamic environments. However, the reverse fanout of step completions through slurmds does happen due to the job's alias list.

Node Addition and Removal - WARNING

- ⚠ A word of warning for **NO_CONF_HASH** and **config_overrides** ⚠
- config_overrides - Should not be used to add or remove nodes
- Any node with less than the configured resources will **not** be set DRAIN
- NO_CONF_HASH - Is dangerous since sites need to know when nodes fall out of sync.

Partition Addition and Removal

- Do not remove a partition with running jobs on it!!!!
  - Recommended draining the partition first.
- Adding and removing node from a partition with running jobs
  - Fine, so long as the node is not removed from the slurm.conf until all jobs complete on that node.
  - Moving nodes between partitions is also fine.

Configless

Configless

"Configless" Slurm is a feature that allows the compute nodes — specifically the slurmd process — and user commands running on login nodes to pull configuration information directly from the slurmctld instead of from a pre-distributed local file.

Note: slurmdbd does not need or use/need these config files and so it does not make sense to have this service use configless

Configless

- There are no extra steps required to install this feature. It is built in by default starting with Slurm 20.02.
- SlurmctldParameters=enable_configless in slurm.conf and restart slurmctld.

Configless - Continued

- Enabled with one of the following (slurmd's):
  - Start slurmd with "--conf-server <srv_name>"
  - By setting a DNS SRV record and ensuring there is no local configuration file on the compute node.

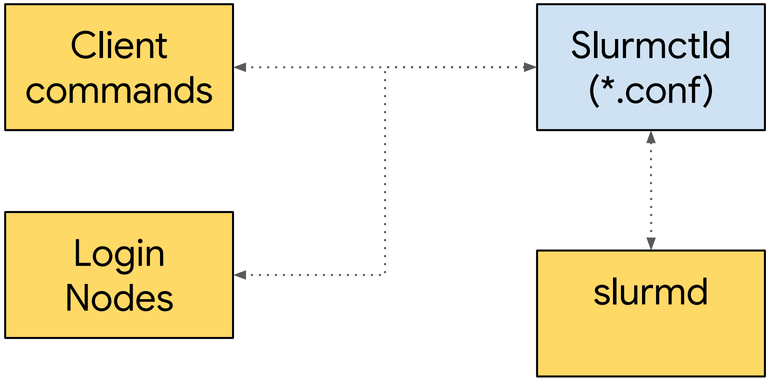
Configless - Continued - example “--conf-server”

- --conf-server
  - slurmd --conf-server slurmctl-primary:6817
- DNS SRV record
  - _slurmctld._tcp 3600 IN SRV 10 0 6817 slurmctl-backup
  - _slurmctld._tcp 3600 IN SRV 0 0 6817 slurmctl-primary

Configless - Continued

Key

- Configless 🟨
- Config server 🟦



Configless - Continued - Client Commands

- On nodes with slurmd running
    - It is assumed that you won’t have default conf files
    - The commands will check the synced location for config files
        - Config files will be in SlurmdSpoolDir under the /conf-cache/, and a symlink to this location will be created automatically in /run/slurm/conf

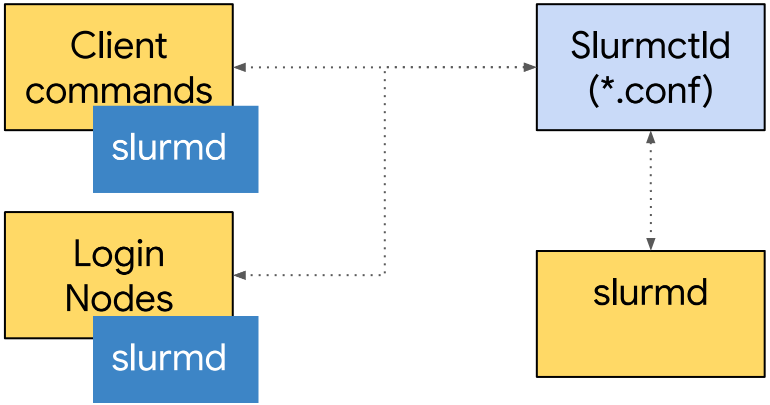
Configless - Continued - Client Commands

- We generally suggest that you run a slurmd to manage the configs on those nodes that run client commands, including submit or login nodes
- Without a slurmd to cache configs it can cause a bit of an RPC storm if the site has client commands request configs directly from the server.

Configless - Continued

Key

- Configless 🟨
- Config server 🟦
- Slurmd (conf only) 🟦



Configless - Continued

Verifying functionality

- Config files will be in SlurmdSpoolDir under the /conf-cache/
- A symlink to this location will be created automatically in /run/slurm/conf
- You can confirm that reloading is working by adding a comment to your slurm.conf on the slurmctld node and running scontrol reconfig and checking that the config was updated.

Configless - Continued - Order of precedence

1. The slurmd --conf-server $host[:$port] option
2. The -f $config_file option
3. The SLURM_CONF environment variable (if set)
4. The default slurm config file (likely /etc/slurm.conf)
5. Any DNS SRV records (from lowest priority value to highest)

Configless - Continued - supported configs

- slurm.conf
- acct_gather.conf
- cgroup.conf
- cgroup_allowed_devices_file.conf
- ext_sensors.conf
- gres.conf
- job_container.conf
- knl_cray.conf
- knl_generic.conf
- plugstack.conf
- topology.conf
- cli_filter.conf
- helpers.conf
- oci.conf
- mpi.conf

Configless - Continued

- Known issues / limitations
    - As of 22.05 the included configs will be shipped to the slurmds.
        - With prior versions, any additional config files will need to be shared a different way or added to the parent config
    - Adding / Removing Nodes is still not supported even with configless
    - 23.02 #include works. Older versions will not work with #include
- Documentation
    - https://slurm.schedmd.com/configless_slurm.html

Configless - Continued - Traditional Configuration

- Worthy mention - traditional distributed confs
  - If you see "conf mismatch" errors, that should be a priority to fix.

Storing Job Scripts

Archiving Job Scripts

- New feature
    - The ability to store job scripts in the slurmdbd
    - Added in 21.08 and improved caching of script in 22.05
        - Storing batch scripts and env vars are now in indexed tables using substantially less disk space. Those stored scripts in 21.08 will all be moved and indexed automatically when upgrading to 22.05.

Archiving Job Scripts - Continued

- AccountingStoreFlags
    - New / Revised options:
        - Job_script
            - Include the job's batch script in the job start message sent to the Accounting Storage database.

Archiving Job Scripts - Continued

- AccountingStoreFlags
    - New / Revised options:
        - Job_comment (Previously AccountingStorageJobComment) which will store the job's comment.
            - Note: the AdminComment and SystemComment are always recorded in the database.

Archiving Job Scripts - Continued

- SchedulerParameters
  - max_script_size=#
    - Specify the maximum size of a batch script, in bytes. The default value is 4 megabytes.
      - Larger values may adversely impact system performance

Archiving Job Scripts - Continued

- Querying archived job scripts
    - **sacct -j <jobid> --batch-script**
        - Returns the archived jobs script if archiving is enabled
- Other semi-related useful information
    - sacct -o SubmitLine
    - sacct -o Comment
    - sacct --helpformat
        - Run to query these and other options that sacct provides

Archiving Job Scripts - Continued

- Example

```
$ sacct -j 5458 --batch-script

Batch Script for 5458
------------------------------------------------------------
#!/bin/bash
sleep 10
env
```

Slurmdbd purge and archiving

Slurmdbd purge and archiving

- The slurmdbd database can grow very large with time, and especially for sites that run large numbers of jobs.
  - Backing up and Truncating tables can help performance, especially when you no longer need to immediately access jobs from past years.
  - Slurmdbd has a rich set of options to Purge/Archive data

Slurmdbd purge and archiving

- For sites that need historical access, this information can be moved to a non-production slurmdbd/database for semi quick access.

Slurmdbd purge and archiving

- Purge/Archive - slurmdbd.conf
  - [##months|##days|##hours]
  - PurgeUsageAfter=”12**hours**”
  - PurgeJobAfter=”12**months**”
  - PurgeEventAfter=”12**days**”

Slurmdbd purge and archiving

- Side note about an archived slurmdbd instance
  - This slurmdbd instance is separate from the archive options mentioned previously
  - This is an isolated instance of slurmdbd which runs a copy or part of a copy of the production database.
  - Can be used to query historical information

Slurmdbd purge and archiving

- Pros:
  - Does not impede production
  - Allows admins to prune data from production, thus keeping only necessary data in production
- Cons:
  - You have to use a separate instance on a different machine
  - The associations need to match production
    - It is not enough to create these. The associations need to match exactly

Scalability

Scalability

- Fast statesave (system requirements)
    - Your maximum system throughput, and overall Slurm controller responsiveness under heavy load, will be governed by latency reading/writing from StateSaveLocation.
    - In high-throughput (~200k+ jobs/day) environments, you may be much better off with a local NVMe drive in a single controller.

Scalability

- Especially if the alternative is an NFS mount shared with users that gets hammered frequently.
- You’re more likely to see performance issues related to this than an outage from the controller dying.

Scalability - Continued

- Munge threads
  - By default, the Munge daemon runs with two threads, but a higher thread count can improve its throughput. We suggest starting the Munge daemon with ten threads for high throughput support (e.g. "munged --num-threads 10").
  - Other sites may also benefit from this change, and increasing the thread count to 10 will not have negative effects.

Scalability - Continued

- Nss_slurm
  - nss_slurm is an optional NSS plugin that can permit passwd and group resolution for a job on the compute node to be serviced through the local slurmstepd process, rather than through some alternate network-based service such as LDAP, SSSD, or NSLCD.
  - Cloud nodes - no need to set up / maintain identity management on short-lived cloud systems

Scalability - Continued

- nss_slurm is not meant as a full replacement for network directory services such as LDAP, but as a way to remove load from those systems to improve the performance of large-scale job launches
- It accomplishes this by removing the "thundering-herd" issue should all tasks of a large job make simultaneous lookup requests
    - Generally for info related to the user themselves, which is the only information nss_slurm will be able to provide

Scalability - Continued

- Limitations
  - nss_slurm will only return results for processes within a given job step. It will not return any results for processes outside of these steps, such as system monitoring, node health checks, prolog or epilog scripts, and related node system processes.
- Documentation
  - https://slurm.schedmd.com/nss_slurm.html

Scalability - Continued

- Final thoughts
  - Other actions a site can take such as turning their Slurm instance

- SchedulerParameters
- Timeouts
- Previously mentioned: the use of array jobs
- Ulimits
- Disabling unnecessary plugins
- Limiting logging to “error” messages only
- nscd caching
- sssd caching

Scalability - Continued

- More help and resources found via
  - SchedMD support
  - https://slurm.schedmd.com/high_throughput.html

Cgroups

Cgroups

- Cgroups V1 support was rewritten for 21.08
    - Fixed a number of issues and bugs
    - Added more error logging
- Seamless transition when upgrading
    - No end user or configuration changes
- Cgroup v2 support was added in 22.05
    - Mixed environment are supported so long as
        - Nodes will need to have either v1 or v2 enabled
        - Hybrid nodes with v1 and v2 enabled are not supported

Cgroups - Continued

Plugin race on fini:

```
[2021-07-26T16:37:12.968] [18142.extern] debug2: _file_read_uint64s: unable to open
'/sys/fs/cgroup/memory/slurm_gamba1/uid_1000/job_18142/step_extern/memory.failcnt' for reading : No such file or directory
[2021-07-26T16:37:12.968] [18142.extern] debug2: xcgroup_get_uint64_param: unable to get parameter 'memory.failcnt' for
'/sys/fs/cgroup/memory/slurm_gamba1/uid_1000/job_18142/step_extern'
[2021-07-26T16:37:12.968] [18142.extern] debug2: unable to read 'memory.failcnt' from
'/sys/fs/cgroup/memory/slurm_gamba1/uid_1000/job_18142/step_extern'
```

Looking for incorrect files:

```
[2021-07-26T16:37:09.613] [18142.0] debug2: xcgroup_load: unable to get cgroup '/sys/fs/cgroup/cpuset'
entry '/sys/fs/cgroup/cpuset/slurm_gamba1/system' properties: No such file or directory
[2021-07-26T16:37:09.613] [18142.0] debug2: xcgroup_load: unable to get cgroup '/sys/fs/cgroup/memory'
entry '/sys/fs/cgroup/memory/slurm_gamba1/system' properties: No such file or directory
```

```
[2021-07-26T16:37:09.627] [18142.0] debug2: Sending SIGKILL to pgid 31137
[2021-07-26T16:37:09.627] [18142.0] debug2: xcgroup_delete: rmdir(/sys/fs/cgroup/cpuacct/slurm_gamba1/uid_1000/job_18142): Device or resource busy
[2021-07-26T16:37:09.627] [18142.0] debug2: jobacct_gather_cgroup_cpuacct_fini: failed to delete /sys/fs/cgroup/cpuacct/slurm_gamba1/uid_1000/job_18142 Device or resource busy
[2021-07-26T16:37:09.627] [18142.0] debug2: xcgroup_delete: rmdir(/sys/fs/cgroup/cpuacct/slurm_gamba1/uid_1000): Device or resource busy
[2021-07-26T16:37:09.627] [18142.0] debug2: jobacct_gather_cgroup_cpuacct_fini: failed to delete /sys/fs/cgroup/cpuacct/slurm_gamba1/uid_1000 Device or resource busy
[2021-07-26T16:37:11.288] [18142.0] debug2: xcgroup_delete: rmdir(/sys/fs/cgroup/memory/slurm_gamba1/uid_1000/job_18142): Device or resource busy
[2021-07-26T16:37:11.288] [18142.0] debug2: jobacct_gather_cgroup_memory_fini: failed to delete /sys/fs/cgroup/memory/slurm_gamba1/uid_1000/job_18142 Device or resource busy
[2021-07-26T16:37:11.288] [18142.0] debug2: xcgroup_delete: rmdir(/sys/fs/cgroup/memory/slurm_gamba1/uid_1000): Device or resource busy
```

Doing the same operation again and again, and *logging it*:

```
[2021-07-26T16:37:09.613] [18142.0] debug:  jobacct_gather_cgroup_cpuacct_attach_task: jobid 18142 stepid 0 taskid 0 max_task_id 0
[2021-07-26T16:37:09.613] [18142.0] debug:  xcgroup_instantiate: cgroup '/sys/fs/cgroup/cpuacct/slurm_gamba1' already exists
[2021-07-26T16:37:09.613] [18142.0] debug:  xcgroup_instantiate: cgroup '/sys/fs/cgroup/cpuacct/slurm_gamba1/uid_1000' already exists
[2021-07-26T16:37:09.613] [18142.0] debug:  xcgroup_instantiate: cgroup '/sys/fs/cgroup/cpuacct/slurm_gamba1/uid_1000/job_18142' already exists
```

slurmd shutdown:

```
slurmd: debug2: _file_read_uint32s: unable to open '(null)/tasks' for reading : No such file or directory
slurmd: debug2: xcgroup_get_pids: unable to get pids of '(null)'
```

Cgroups - A Quick Note On TaskPlugin

- NOTE: It is recommended to stack task/affinity,task/cgroup together when configuring TaskPlugin, and ConstrainCores=yes in cgroup.conf.
    - Setting “task/affinity,task/cgroup” plugins combining the best of both resource containment pieces and enables the srun “--cpu-bind and/or --mem-bind”.
        - task/affinity - **sched_setaffinity()**.
        - task/cgroup - **linux control cgroups**

Cgroups - A Quick Note On TaskPlugin

- TaskAffinity was removed from the cgroups.conf as of 21.08.
    - Sites that upgrade will need to remove this parameter from their cgroup.conf or config validation will not pass resulting in your services not starting.
    - Superseded by "TaskPlugin=task/affinity,task/cgroup".

Cgroups - Continued

- We encourage you to upgrade if you have run into issues

Random Notes

A few random suggestions - coredumps

- Core Dumps
    - Make sure you know the location of your core dumps.
    - If slurmctld is started with the -D option, then the core file will be written to the current working directory.
    - If SlurmctldLogFile is an absolute path, the core file will be written to this directory.
    - Otherwise the core file will be written to the StateSaveLocation, or "/var/tmp/" as a last resort.

A few random suggestions - coredumps

- Core Dumps
  - SlurmUser must have write permission for the directories. If none of the above directories have write permission for SlurmUser, no core file will be produced.
  - For testing purposes the command "scontrol abort" can be used to abort the slurmctld daemon and generate a core file.

A few random suggestions - coredumps

- Core Dumps
  - Please do not send us your core files
- Instead gather a backtrace
    - Core files:
      - gdb -ex 't a a bt' -batch slurmctld core.12345
    - If a crash happens at startup, the following can be used.
      - gdb --args slurmctld -Dvvvv
      - (gdb)“t a a bt full”

A few random suggestions - coredumps

- Core Dumps
  - There are other considerations. Please see the FAQ for further details
  - https://slurm.schedmd.com/faq.html#core_dump

A few random suggestions - Debugging

- Debugging Slurm services
  - slurmd -D
  - slurmctld -D
  - slurmdbd -D
- Useful debug commands options
  - scontrol setdebug debug
  - -vvv options work with most commands

A few random suggestions - Debugging

- sbatch --test-only
  - Validate the batch script and return an estimate of when a job would be scheduled to run given the current job queue and all the other arguments specifying the job requirements.
  - No job is actually submitted.

Commercial Support

Commercial Support

- SchedMD offers “Level-3” support only.
    - Our definition of these levels are online at https://www.schedmd.com/support.php
- Customers must be comfortable with day-to-day operations, basic troubleshooting, and initial setup.
- On-Site training is available to assist with initial onboarding and software configuration.
    - Email sales@schedmd.com

Commercial Support

- Our workflow is built around Bugzilla, allowing transparent hand off between support engineers, and thorough tracking.
  - Bugs are public by default, however they can be marked private when required.
  - https://bugs.schedmd.com

Commercial Support

- Specific contractual timelines for each Severity level.
- Support hours are 2 AM to 5 PM Mountain Time / 8:00AM - 11:00PM GMT, Monday - Friday.
- We strive to always significantly exceed the SLA, regardless of system size or complexity.

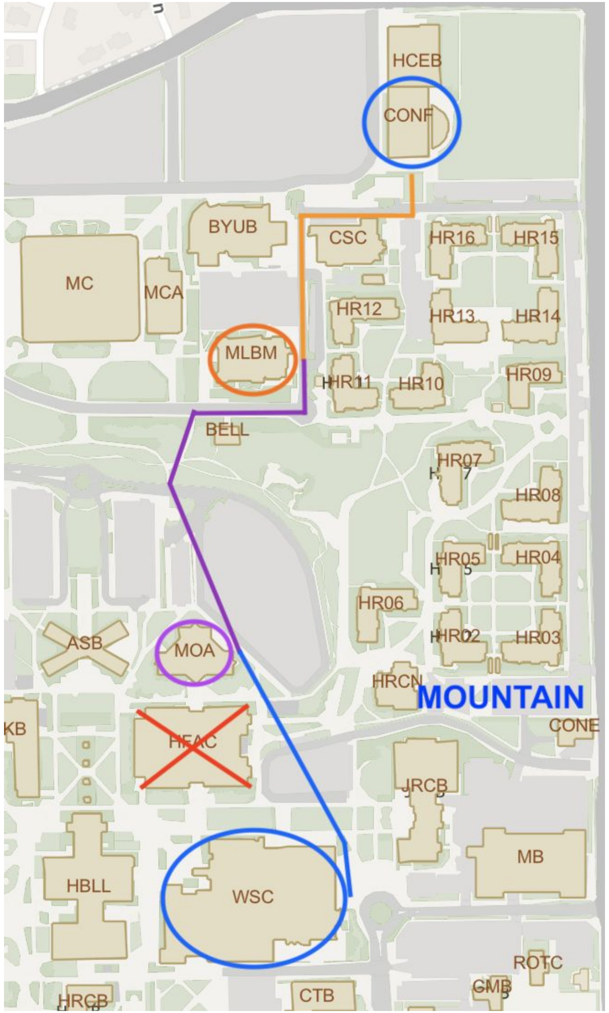
Questions?

SCHEDMD

The Slurm Company

Dinner, 6:30pm at The Skyroom

- Top floor of the Wilkinson Student Center (WSC)
- Campus attractions en route
  - Bell tower - BELL
  - MLBM - life science museum, closes at 9pm
  - MOA - art museum, closes at 6pm
- The X'd out building was recently demolished, stay on the mountain (east) side of campus to avoid some construction quirks


HCEB
CONF
BYUB
CSC
HR16
HR15
MC
MCA
HR12
HR13
HR14
MLBM
HR11
HR10
HR09
BELL
HR07
HR08
HR05
HR04
HR06
ASB
MOA
HR02
HR03
HRCN
MOUNTAIN
CONE
KB
HFAC
JRCB
MB
HBLL
WSC
ROTC
GMB
HRCB
CTB