



Field Notes From the Frontlines of Support


Jason Booth
SchedMD



Slurm User Group Meeting 2021

Agenda

All times are US Mountain Daylight (UTC-6)



Time	Speaker	Title
9:00 - 9:50	Jason Booth	Field Notes 5: From The Frontlines of Slurm Support
10:00 - 10:25	Nate Rini	REST API <i>and also</i> Containers
10:30 - 10:50	Marshall Garey	burst_buffer/lua and slurmscriptd
11:00 - 11:25	Nick Ihli	Slurm in the Clouds
11:30 - 11:50	Tim Wickberg	Slurm 21.08 and Beyond

Copyright 2021 SchedMD LLC

<https://schedmd.com>

Welcome



- Five separate presentations, five separate streams
- Presentations will remain available for at least two weeks after SLUG'21 concludes
- Presentations are available through the SchedMD Slurm YouTube channel
 - <https://youtube.com/c/schedmdslurm>
- Or through direct links from the agenda
 - https://slurm.schedmd.com/slurm_ug_agenda.html

Asking questions



- Feel free to ask questions throughout through YouTube's chat
- Chat is moderated by SchedMD staff
 - Tim McMullan, Ben Roberts, and Tim Wickberg
 - Also identified by the little wrench symbol next to their name
- Questions will be relayed to the presenter by the moderators
 - Some may be deferred to the end if they cannot be relayed in a timely fashion
 - Or some may be answered by the moderators in chat directly



Field Notes From the Frontlines of Support

Jason Booth
SchedMD

Knowledge Nuggets

A collection of useful Slurm knowledge nuggets



Copyright 2021 SchedMD LLC
<https://schedmd.com>

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 [Bug 7609](#)
 - 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
 - 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.
- **Simple Linux Utility for Resource Management** (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
- 2017 11 Employees
- ...
- 2021 28 Employees that support, train, develop and plan.
 - 22 Support/Development staff

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 - Slurm Client Base



- Because Slurm is open source we don't know how many sites are running Slurm
- We do know how many sites SchedMD supports
 - 2013 - 67
 - 2014 - 124
 - 2015 - 193
 - 2016 - 244
 - 2017 - 292
 - ...
 - 2021 - 460 (286 individual), (174 Sub-sites)

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 slurmd host is preferred
- Fast path to the StateSaveLocation
 - IOPS this filesystem can sustain is a major bottleneck to job throughput
 - At least 2 directories and two files created per job
 - 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 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



Upgrading

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

>=

commands



Must stay within 3 major releases.

E.g. {21.08, 20.11, 20.02} is okay,
but {21.08, 20.11, 20.02, 19.05} is not.

Within each major release, you can mix the
maintenance release versions without issue.
E.g. {20.02.5, 20.02.4, 20.02.3,
20.02.2, 20.02.1, 20.02.0} 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/21.08.0/ --sysconfdir=/apps/slurm/etc/  
# ln -s /apps/slurm/21.08.0 /apps/slurm/dbd  
# ln -s /apps/slurm/21.08.0 /apps/slurm/ctld  
# ln -s /apps/slurm/21.08.0 /apps/slurm/d  
# ln -s /apps/slurm/21.08.0 /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.
 - For larger databases, or more unusual systems, you may want to test the upgrading/conversion on a copy of the full production database on a separate machine.
 - Older MySQL versions (5.5 and before) have had problems with later conversion processes.

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



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.
- 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



- We do have plans to make this less painful long-term.
- The new `cons_tres` plugin has split some of these data structures apart, and will eventually let us change this.
- This is blocked until `cons_res` is removed.

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.



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 - Continued



- 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.

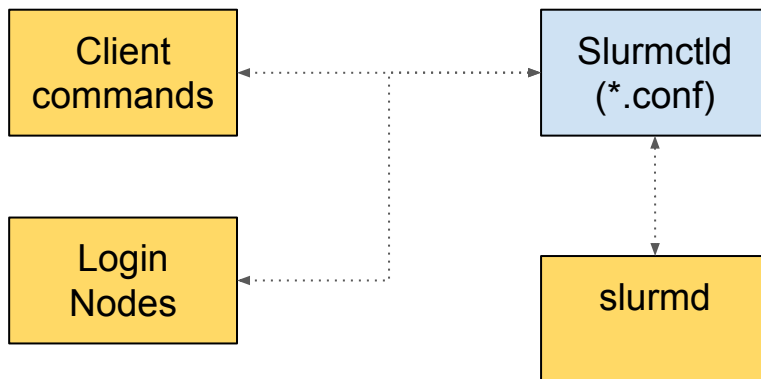
Configless - Continued - example “--conf-server”

- --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
 - If slurmd is not running it will check DNS
 - WARNING - RPC flood is possible




Configless - Continued - Client Commands

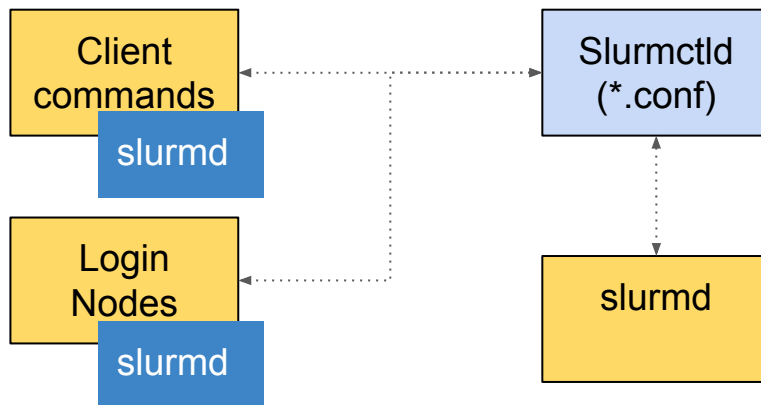


- We generally suggest that you run a slurmd to manage the configs on those nodes that run client commands, including submit or login nodes
- Without a slurmd to cache configs it can cause a bit of an RPC storm if the site has client commands request configs directly from the server.

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 slurmd 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

Configless - Continued

- Known issues / limitations
 - 9330 (Fixed 20.02.6) - Perl API now functions with configless (integration scripts)
 - The included configs will NOT be shipped to the slurmds.
 - 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
- 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.



Archiving Job Scripts

Archiving Job Scripts



- New feature
 - The ability to store job scripts in the slurmdbd
 - Added in 21.08 and sponsored by **Yale**.
 - [Bug 7609](#)

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.
 - 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
 - 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
- 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




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.
 - 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



- New (undocumented) SlurmctldParameters option 
 - "enable_rpc_queue"
 - Slurm 20.11 / added Oct 27, 2020
- Enables experimental RPC queuing mechanism in slurmctld, which may alleviate high contention between RPC processing threads in some environments
- Please log a bug with us if you see issues

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
 - 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

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
[2021-07-26T16:37:09.627] [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: 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 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)'
```

Copyright 2021 SchedMD LLC

<https://schedmd.com>

Cgroups - A Quick Note On TaskPlugin

- NOTE: It is recommended to stack task/affinity,task/cgroup together when configuring TaskPlugin, and setting TaskAffinity=no 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**
- We are looking at completely removing the cgroups.conf TaskAffinity parameter at some point.
 - Site should migrate to using "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 back trace
 - Core files:
 - `gdb -ex 't a a bt' -batch slurmd core.12345`
 - If a crash happens at startup, the following can be used.
 - `gdb --args slurmd -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
 - 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 - continued



- SchedMD offers “Level-3” support only.
 - Our definition of these levels are online at <https://www.schedmd.com/support.php>
- Customers must to 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 - continued



- 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 - continued



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





Questions?

Next Session



- The next presentation is by Nate Rini: "REST API *and also* Containers"
- Starts at 10am Mountain Daylight Time (UTC-6)
- It is on a separate YouTube Live stream
- Please see the SchedMD Slurm YouTube channel for links

End Of Stream



- Thanks for watching!