Slurm SC21 BoF

Tim Wickberg
SchedMD

Agenda

- Slurm Release Schedule
- Slurm 21.08 Release Review
- Slurm Roadmap - 22.05 and 23.02
- Open Q+A

Questions?

- Please ask questions in the Zoom chat
  - I won't see questions through Slido
- Feel free to ask questions throughout
  - I'll try to respond if they're immediately relevant
  - Otherwise I may defer until the Open Q+A at the end

Recording

- The BoFs are not being recorded through the SC21 HUBB
- A recording will be uploaded to the SchedMD YouTube channel
  - https://youtube.com/c/schedmd-slurm
- Slides will be available on the Publications page as well
  - https://slurm.schedmd.com/publications.html

SchedMD is hiring!

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

Slurm Release Schedule

Tim Wickberg
SchedMD

Slurm Releases

- 20.11 - November 2020 - released during SC'20 BoF!
- 21.08 - August 2021
- 22.05 - May 2022
- 23.02 - February 2023

Slurm Release Schedule

- Slurm major releases come out every nine months
- Major release numbers are the two digit year, period, two digit month
  - 21.08 ⇒ 2021, August
- Maintenance releases, such as 21.08.1, come out roughly monthly for the most recent major release
- Two most recent major releases are supported
  - These are 21.08 and 20.11 currently

Slurm 21.08 Release

Job submission command

- Un-modified command now captured by default
- Available through new '-o SubmitLine' output format in sacct

Job submission command

```
tim@blackhole:~$ sacct -o JobId,User,SubmitLine%50
JobID             User                                            SubmitLine
------------ --------- --------------------------------------------------
1358               tim               sbatch --wrap sleep 1000 --exclusive
1358.batch
1359               tim          sbatch --wrap sleep 1000 --exclusive -N 2
```

Store batch scripts in SlurmDBD

- New AccountingStoreFlags=job_script option in slurm.conf
  - As well as new AccountingStoreFlags=job_env
- 'sacct --batch-script' and 'sacct --env-vars' to fetch them
- Note: 21.08.4 fixed a security issue with this where unprivileged users could retrieve these files
  - Please upgrade before enabling

Store batch scripts in SlurmDBD

```
tim@blackhole:~$ sbatch --wrap "sleep 1000" --exclusive -N 2
Submitted batch job 1360
tim@blackhole:~$ sacct --batch-script -j 1360
Batch Script for 1360
--------------------------------------------------------------------------------
#!/bin/sh
# This script was created by sbatch --wrap.

sleep 1000

tim@blackhole:~$
```

New "PLANNED" node state

- PLANNED now shown instead of IDLE for nodes that are being held empty while waiting for a multi-node job to launch

New "PLANNED" node state

Slurm 20.11:

```
tim@blackhole:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1359   general     wrap      tim PD       0:00      2 (Resources)
              1358   general     wrap      tim  R       1:49      1 node0004
tim@blackhole:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
general*     up    4:00:00      1   idle node0005
general*     up    4:00:00      1  alloc node0004
```

New "PLANNED" node state

Slurm 21.08:

```
tim@blackhole:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1359   general     wrap      tim PD       0:00      2 (Resources)
              1358   general     wrap      tim  R       1:49      1 node0004
tim@blackhole:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
general*     up    4:00:00      1   plnd node0005
general*     up    4:00:00      1  alloc node0004
```

RS256 token support in auth/jwt

- Keys specified through a JWKS file
  - Such as those generated by AWS Cognito
- AuthAltParameters=jwks=/path/to/my.jwks
- May be used alongside existing HS256 support

RS256 token support in auth/jwt

```
tim@blackhole:~$ grep jwks /etc/slurm.conf
AuthAltParameters=jwks=/etc/slurm/jwks.json,jwt_key=/root/jwt_hs256.key
tim@blackhole:~$ cat jwks.json
{"keys":[{"alg":"RS256","e":"AQAB","kid":"fZqKj+4Zw9OhMC4XNtWWGQC8n8iDxVoy6HLMLkONNuY=","
kty":"RSA","n":"7Lm5UDivRbAXNQ9-Fl5vVty1fA1jTTRrN9RJTlXoiFMJPgfgWqDHOWAIO2OtQur3bsGMckUQ_
7ZbRwZnbtMeDZ-QGAb-gWJ5mjxCegRD0xPC9QoulZzNDm3oB_56jsMDRuYUI6Q0qvC3QiXzurmNtUJwmRhE1mlTwQ
wc5b-b8mJBYHjIW3ROAAe3Onr9T7NPenQ1BzOi8DKYo35RwJEQYCz0hRsX2cpztOhBTDU5nvgkY1I6f1bQtgpMT6j
Z1HFjjX7IQGVcIjU0W3F_rj-0JAccmFlskog3Vyos0cA7WRvQdJc2iMulznBAoeLsNRJ0rp0A361APDQQdcnoeI7C
9w","use":"sig"},{"alg":"RS256","e":"AQAB","kid":"/zFkNPInxOO+4p7u2ccOSLQnMMxaulgPRr+3/0j
1YMs=","kty":"RSA","n":"vMo6Ad50H8wOEvWIYyRXVXH7wB-aob9Um1GG2W-XCY4Eb7bSoqMDBTZZZgCb1IAzG
megs7QXuA5O699Jfs0LrupC9TVB_zWkiU4DAIdB9RUeSBubmPCDJMobSK3L4UWVnqGdSf_c078CyyoumNSFhwRddo
tdzAKglRxMiCzvy3Zgldx3l3iNpeQRUTWJ_x8Du5eiirjqB4zdof9vwQ_DFVP0c9zRWZSheV7XD3lnqv1sBMYVYZs
DxX_FBGU5flG8ExIZV2pV0jbHva7N1V6k3J69rwYfG5E9-d-JZKEXyIFMHPAl8zZQmUgEvXVusIJe6STJLKHgSZAw
a-eFKiQV6w","use":"sig"}]}
```

Improved cgroup subsystems

- Significant refactoring work for the task/cgroup, proctrack/cgroup, and jobacct_gather/cgroup plugins
- Still only supports cgroup v1
- All cgroup interactions now handled centrally
  - Preparation for future cgroup v2 support

burst_buffer/lua

- "Generic" "Burst Buffer" support
- Really a means of handling pre- and post- job setup
  - Asynchronously
  - Compute nodes not yet assigned
- Avoids wasting compute node time for large job starts by handling setup and teardown tasks while they are still running other jobs

burst_buffer/lua

- See separate presentation by Marshall at SLUG'21 for further details

New 'slurmscriptd' process

- Developed alongside burst_buffer/lua
- fork()+exec() in slurmctld is **very** expensive for systems with high-throughput and high job counts
- Instead, the slurmscriptd process launches scripts on behalf of slurmctld
  - Limited to burst_buffer and PrologSlurmctld/EpilogSlurmctld
  - Expect to expand this in the future

Fixes to job_container/tmpfs

- job_container/tmpfs was snuck into the 20.11.5 maintenance release early
- Strong early adoption exposed a number of design issues with how the slurmd/slurmstepd shared responsibility for the namespaces
- Fixed with further refactoring in 21.08

json and yaml output

- sacct, sinfo, and squeue now have --json/--yaml options for output
- Uses same underlying serialization/translation code as slurmrestd, but in the standalone command
- Output only

json and yaml output

```
tim@blackhole:~$ sacct --json|jq .jobs|head -n 14
[
  {
    "account": "root",
    "comment": {
      "administrator": null,
      "job": null,
      "system": null
    },
    "allocation_nodes": 1,
    "array": {
      "job_id": 0,
      "limits": {
        "max": {
          "running": {
```

Shared libraries and 'srun --bcast'

- Added new feature to 'srun --bcast' to allow it to automatically identify and broadcast required shared libraries as part of job launch
- Creates a directory alongside the broadcasted executable, and prepends that into LD_LIBRARY_PATH as part of step launch
- Avoids "thundering herd" issues on parallel filesystems on massively parallel job launches
  - Single srun process reads the executable and libs

Shared libraries and 'srun --bcast'

- Enabled through BcastParameters=send_libs
  - Disabled by default
  - Or through 'srun --bcast --send-libs ./my_program'
- New BcastExclude option can set system library directories to ignore
  - Defaults to "/lib,/usr/lib,/lib64,/usr/lib64"
  - No point in sending ld-linux-x86-64.so.2 or libc.so.6

OCI Container Support

- Initial support for launching processes in OCI containers
- See Nate's presentation from SLUG'21 for further details

Improved Job Step Throughput

- Significant performance improvements to job step launch
- Nicely complements past performance work with experimental SlurmctldParameters=enable_rpc_queue option

PowerSave/Cloud changes

- SuspendTime, SuspendTimeout, ResumeTimeout on partitions
  - SuspendTime will enable PowerSave if disabled at global level
  - Helps with "hybrid" (bursting from on-premise) setups
  - With SuspendTime on the partition, you can disable PowerSave at the global level and enable on specific cloud partitions
    - SuspendTime=INFINITE
    - PartitionName=cloud … SuspendTime=300

PowerSave/Cloud changes

- JSON mapping of jobs to nodes available to ResumeProgram

```
SLURM_RESUME_FILE=/proc/1647372/fd/7:
{
   "all_nodes" : "cloud[1-3]",
   "jobs" : [
      {
         "job_id" : 140814,
         "nodes" : "cloud[1-3]",
      },
      {
         "job_id" : 140815
         "nodes" : "cloud[1-2]",
      }
   ]
}
```

Extra

- Node has an "Extra" field that is managed through scontrol
  - Same node comment but has named "Extra"

SELinux Support for Job Launches

- Can be used to capture a requested context at job submission time
  - Context must be validated within a job_submit plugin
  - MUNGE does not include the context, no way to securely capture
- Once validated by site-provided job_submit, processes for that job will be launched within that context

SELinux Support for Job Launches

- Disabled by default, must be configured with “--enable-selinux”
- SELinux must be enabled on the nodes, but is not needed on the ctld hosts
- Enable job_submit plugin and write an appropriate script to do context validation

Slurm 22.05 Roadmap

"Preferred" node constraints

- A list of optional ("soft") constraints to be considered when selecting nodes for a job
  - Likely using "--prefer" as the option to salloc/sbatch/srun
  - Job launch will prefer those nodes, but fall back to any nodes if that cannot be satisfied immediately
  - Traditional "hard" constraints (--constraint) will always be respected

cgroup v2 support

- Add support for cgroup v2
  - Only cgroup v1 is currently supported
  - Certain distributions are starting to disable cgroup v1 support by default

Backfill Scheduling Support for Licenses

- Licenses are currently ignored in the backfill scheduler
- If licenses are currently unavailable for a job, no future reservation will be made for it
- This is obviously not ideal for sites with heavy license usage, and can lead to starvation of larger license-dependent jobs

GPU Sharding

- Allow for cooperative GPU sharing between separate jobs
- Allows administrators to define a number of "Slices" for a GPU
  - Jobs can request between zero and all slices
  - All slices allocated to the job from a single GPU, cannot span between cards
- Caveat: no hardware enforcement
  - Jobs must cooperate effectively

AcctGatherInterconnect plugins

- Add support for gathering network statistics from OmniPath and Slingshot interconnects

Changes to LLN Support

- LLN ("Least-Loaded Node") currently defines the least-loaded nodes as those with the most idle cores
- This can lead to counter-intuitive behavior in partitions with mixed hardware
- Definition will change to LLN being the lowest proportion of allocated cores to total cores within the node

Accounting - Operation Without Defaults

- Adding a new option to SlurmDBD to allow operation without DefaultAccounts set for every user

Slurm 23.02 Roadmap

Truly Dynamic Nodes

- Move away from current FUTURE node handling
  - Support truly dynamic node addition and removal from the cluster
- Some underlying work will be in 22.05, but will not be ready until 23.02

License Management Improvements

- Changes to improve flexibility of remote licenses managed through sacctmgr
- Allow management by explicit license count rather than percentages

License Preemption

- When running with preemption, license usage is not currently considered, and jobs will not be preempted to free up licenses
- This is an issue when using licenses to represent cluster-wide resources

Questions?

Open Q+A

End Of Stream

- Thanks for watching!