

# Slurm 20.11 and Beyond Open Q+A

Tim Wickberg  
SchedMD



# Slurm 20.11 and Beyond

Tim Wickberg  
SchedMD

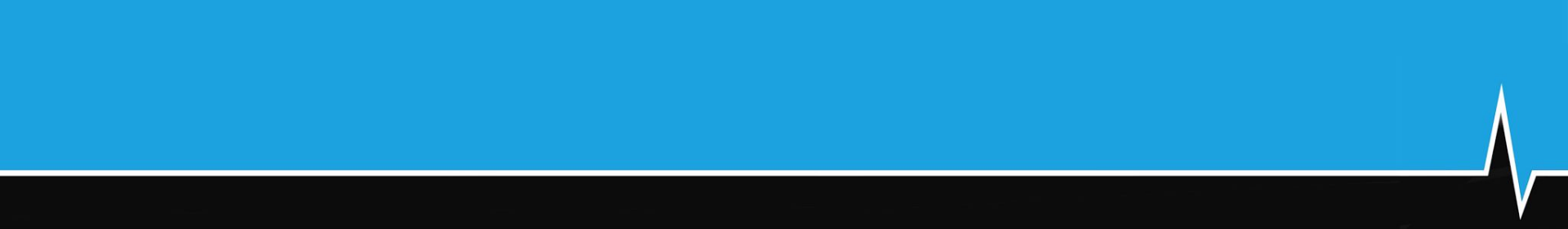
# Slurm Releases

- 20.02 - February 2020
- 20.11 - November 2020
- 21.08 - August 2021

# Slurm Release Schedule



- Slurm major releases come out every nine months
- Major release numbers are the two digit year, period, two digit month
  - 20.02 ⇒ 2020, February
- Maintenance releases, such as 20.02.5, come out roughly monthly for the most recent major release
- Two most recent major releases are still supported
  - This is 20.02 and 19.05 currently



# Slurm 20.02 Release

Tim Wickberg  
SchedMD

# REST API

- See separate presentation in publication archive
- Initial version handles common slurmctld interactions

# AuthAltTypes

- Allow slurmctld to talk different authentication protocols simultaneously
- Add a new auth/jwt plugin
  - Users can request tokens through `scontrol token`

# Configless Slurm

- New way to setup the cluster
- See examples from "Field Notes 4" earlier today



# Retroactive WCKey Updates



- "sacctmgr update jobid=<foo> set newwckey=correctkey"
- Supports selection by user, current wckey, and can limit to a specific date range
- Rerolls usage so sreport data is updated as well

# OverSubscribe=EXCLUSIVE

- Update OverSubscribe=EXCLUSIVE to always assign all TRES in the job to the job
  - OverSubscribe=EXCLUSIVE is used to always provide full-node allocations on a partition
  - Current (Slurm <= 19.05) behavior is to assign all CPUs and Memory, but not to assign any further GRES automatically

# FastSchedule is gone



- Remove FastSchedule option
  - FastSchedule=0 does not work properly with cons\_tres
  - Deprecated in 19.05.3+, you will see errors in slurmctld/slurmd log files warning about this
- New SlurmdParameters=config\_overrides
  - Replaces FastSchedule=2 functionality
  - Used for test/development when you need to lie about the actual hardware
  - Still not recommended for production use

# burst\_buffer/datawarp additions

- Adding % replacement syntax for #DW / #BB directives
- Replace the symbol with the correct value for the job

#DW / #BB Symbol	Replacement
\\	Stop further symbol processing.
%%	A single % symbol.
%A	Job array's master job allocation number.
%a	Job array ID (index) number.
%d	WorkDir.
%j	Job ID.
%u	User name.
%x	Job name.

# Prolog/Epilog Refactoring



- Move Prolog/Epilog/PrologSlurmctld/EpilogSlurmctld behind a new plugin interface - "PrEpPlugins"
  - Current script functionality moves into the "script" plugin type
  - Allows easier access to the underlying job launch data

# Adjustments to PMI



- Change how libpmi.so (PMI1) links to avoid direct dependency on libslurm.so.<VERSION>
- Workaround for OpenMPI statically linking to our libpmi.so, and thus inheriting a dependency on libslurm.so.<VERSION>
  - Which then breaks your OpenMPI installs for each Slurm upgrade

# NodeSet syntax for slurm.conf



- New "NodeSet" syntax for slurm.conf
- Define a NodeSet as a set of Nodes
  - Or, select all nodes with a given Feature defined
- And then use the node sets interchangeably with the node names as part of your Partition definitions
  - Rather than error-prone copy+pasting long lists

# NodeSet syntax for slurm.conf

```
NodeName=node[0001-0005] Features=e2620v4,xeon,qdr,v100  
NodeName=node[0006-0010] Features=e2620v5,xeon,qdr  
NodeName=node[0011-0015] Features=e2620v6,xeon,ndr
```

```
NodeSet=v100nodes Feature=v100  
NodeSet=random Nodes=node[0001,0003,0008-0013]  
NodeSet=imaginary Feature=e2620v6
```

```
PartitionName=v100 Nodes=v100nodes  
# -> node[0001-0005]  
# these two sets overlap in part, but the slurmctld would de-duplicate for us:  
PartitionName=somestuff Nodes=random,imaginary  
# -> node[0001,0003,0008-0015]
```



# "Magnetic" Reservations



- Add "Magnetic" option to Reservations
- Jobs with matching account/qos settings will be eligible to run in these reservations even if they have not specified --reservation on the submission
  - They will still be considered for execution outside of the reservation



# Slurm 20.11 Roadmap

# REST API

- Extend to cover common slurmdbd interactions
- See Nate Rini's presentation from SLUG'20 for further details

# IPv6

- IPv6 support throughout Slurm
- For 20.11, Slurm will require a CommunicationParameters option to enable dual-stack support
- For 21.08 (tentatively), we will enable dual-stack by default, with an option to force IPv4-only

# MariaDB SSL Connection Support



- See StorageParameters in slurmdbd.conf(5) for setup

# Heterogeneous Job Steps



- Similar to HetJobs, but extended to step launch within an existing "normal" job

# Heterogeneous Job Steps

```
tim@blackhole:~$ salloc -N 2 --exclusive
salloc: Granted job allocation 24217
tim@blackhole:~$ srun -N 1 echo a : -N 1 echo b
```

```
a
```

```
b
```

```
tim@blackhole:~/slurm$ sacct -j 24217
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
24217	bash	general	root	8	RUNNING	0:0
24217.extern	extern		root	8	RUNNING	0:0
24217.0+0	echo		root	1	COMPLETED	0:0
24217.0+1	echo		root	1	COMPLETED	0:0

# --threads-per-core

- Previously only affected allocation
- Now influences placement of tasks
- Implies `--cpu-bind=threads`
- Like `--hint=nomultithread`, but more control for threads>2



# --threads-per-core

- Max number of threads per core
  - `srun -n1 -c2 --threads-per-core=2 prog`
    - Places two cpus on one 2 threaded core
  - `srun -n1 -c2 --threads-per-core=1 prog`
    - Places one cpu per core

# Job Step Allocation Behavior



- For 20.11, job steps default to exclusive access to their assigned resources within a job
- Past behavior of automatically overlapping job steps was confusing, and make further job step enhancements difficult
- 20.02 and prior behavior is available through the new --overlap flag to srun, although all steps must agree to overlapping use
  - This now matches the "OverSubscribe" logic for when to share resources

# Job Step Allocation Behavior



- Unfortunately this breaks the existing use of `SallocDefaultCommand` to move a user's `salloc` terminal to the compute nodes
- As such, `SallocDefaultCommand` has been removed, in favor of the new....

# "Interactive" Job Step



- Easier way to force a user's terminal to the compute nodes when using salloc
- Replaces complicated SallocDefaultCommand settings with a new "Interactive" step
- LaunchParameters=use\_interactive\_step to enable
- Tracked in accounting appropriately, and will not cause confusing step-launch issues for GRES or GPUs

# "Interactive" Job Step



- New InteractiveStepOptions option gives some control over the default launch behavior
  - Similar to the remove SallocDefaultCommand rules
  - Primarily intended for sites that were using --export tricks to modify the environment
  - Default is "--interactive --preserve-env --pty \$SHELL" which covers most existing uses for SallocDefaultCommand

# TRES

- Add new `--ntasks-per-gpu` option
  - Does what it says on the tin

# Mail Type

- New "Invalid Dependency" mail type
  - Message sent when job is removed due to invalid dependencies
    - Due to DependencyParameters=kill\_invalid\_depend

# Reservations

- Allow users to delete reservations
  - Enable with new SlurmctldParameters=user\_resv\_delete option
  - Only permitted if they would have been allowed to run within the reservation
- Allow multi-reservation job submission:
  - `sbatch --reservation=foo,bar,baz myjob.sh`



# Reservations



- New AllowGroups access control on a reservation
  - Permit access by UNIX group
- Skip the next occurrence of a repeating reservation:
  - `scontrol update reservation=weekly_resv skip`

# scrontab

- Permit users to submit recurring jobs, with a crontab compatible syntax for recurrence
- And add a new "scrontab" user command to manage them
- Must add ScronParameters=enable to your config to enable
- Highly recommended that your job\_submit.lua be modified to route these to a dedicated partition and set strict limits

# Experimental RPC Queuing



- New (undocumented) SlurmctldParameters option "enable\_rpc\_queue"
- Enables experimental RPC queuing mechanism in slurmctld, which may alleviate high contention between RPC processing threads in some environments



# Slurm 20.11 Anti-Roadmap

# Code removals



- "Layouts / Entities"
  - Finally removed
- Message Aggregation
  - Removed in favor of (experimental) RPC queuing mechanism in slurmctld



... and Beyond

# Expose Additional Scheduling Details



- Mark nodes blocked from running jobs by a future larger job as something other than IDLE
  - Exact display name still TBD (e.g. PreAllocated)
  - Accounting will still reflect these nodes as IDLE, but at least sinfo will separate them and keep your users from complaining that their job won't launch while nodes are IDLE
- Expose a timestamp of the last backfill cycle to consider the job for execution
  - Useful for backfill tuning

# HPE Cray Shasta Support



- In progress





# Open Forum

Danny Auble + Tim Wickberg

Copyright 2020 SchedMD LLC

<https://schedmd.com>

# SchedMD is Hiring



- SchedMD is always looking to hire experienced systems programmers and support staff
- <https://schedmd.com/careers.php>
- [jobs@schedmd.com](mailto:jobs@schedmd.com)

# End Of Stream



- Thanks for watching!