Slurm 23.02, 23.11, and Beyond

Tim Wickberg
Chief Technology Officer

Slurm User Group 2023

slurm | SCHEDMD

SLUG Miscellanea

Presentations

- SLUG'23 presentations be available through the publication archive
  - https://slurm.schedmd.com/publications.html
  - Will be uploaded in ~10 days, a note will be sent to the slurm-users list when available
  - Presenters - Thank You!
    - And if you have any last-minute corrections, or if you would prefer your slides not be included in the archive, please email slugpresent@schedmd.com ASAP

Development Cycle

Release Cycle

- Major releases are currently made every nine months
- Version is the two digit year, two digit month:
  - 23.02 - February 2023
  - 23.11 - November 2023
  - 24.08 - August 2024
- Major releases are supported for 18 months
  - Currently: 22.05 and 23.02
  - After November: 23.02 and 23.11
- Maintenance releases are made roughly monthly
  - Usually only for the most recent major release
    - One main exception - security releases will be made for all supported major releases

Development Process

- Most larger work is handled through sponsored projects
  - SchedMD support only covers maintenance
- Some projects - those of wider community interest - may be handled internally on a best-effort basis

Enhancement Requests

- Our Bugzilla installation catalogs outstanding requests under the "Sev 5 - Enhancement" severity level
  - Unless indicated through the "Target Release" field, SchedMD has not committed to delivering that enhancement (if ever)
    - Currently 551 open tickets... around 30 may make it into a release
- Customer requests are automatically re-routed to Sev 4 on submission for triage, may move to Sev 5 if we agree that's an interesting potential feature
  - Working to be a bit more opinionated on these, rather than leaving them in limbo indefinitely, so they may be closed instead of left unassigned
  - Most enhancements will stay in Sev 5 indefinitely, unless sponsored

Slurm 23.02 - February 2023

New scrun command

- Directly launch OCI-compliant container images
- Slurm's version of crun / runc
- Refer to the "Containers" talk from Tuesday for more details

New --tres-per-task option

- Allow jobs to be modeled as a number of tasks, with all appropriate resource types scaled directly by the number of tasks requested
  - Task can request licenses, GRES, CPUs, memory
  - Note - can't automatically propagate to srun within a batch script in 23.02

AllowAccounts - automatic recursion

- Update the "AllowAccounts" access control to automatically extend access to all child accounts

License Preemption

- When running with preemption, license usage is not considered by default, and jobs will not be preempted to free up licenses
- This is an issue especially when using licenses to represent cluster-wide resources, as they won't be reclaimed to allow higher-priority work to preempt
- Enable with PreemptParameters=reclaim_licenses

Licenses

- https://slurm.schedmd.com/licenses.html#remote_licenses
- Remote licenses can now be set with "flags=absolute"
  - Means the per-cluster assignments are by explicit license count, instead of percent
  - slurmdbd.conf option of AllResourcesAbsolute=yes to enable this by default
- New "LastConsumed" value, designed to be frequently updated with current license server utilization values
  - Propagated to slurmctld automatically
  - Controller automatically factors that current status in when deciding how many licenses can be used for new jobs

```
LicenseName=foobar44@licsrv42
    Total=0 Used=0 Free=0 Reserved=0 Remote=yes
    LastConsumed=0 LastDeficit=0 LastUpdate=2023-02-02T18:20:57
```

Cloud nodes enhancements

- Pass list of requested features to ResumeProgram
- Reset active features on CLOUD nodes
- Allow for Node Weight to be considered on CLOUD nodes
- New flag to automatically power down "Exclusive" nodes once jobs are completed

Reservation Enhancements

- Add a Comment field to reservations
- Show active reservations on each node in 'scontrol show node'
- Support node addition and removal from a reservation through scontrol with += and -= on the node list

Accounting Tweaks

- New FailedNode field
  - Set for jobs that have been terminated due to a node failure
  - Help triage hardware issues

New job completion plugin

- New jobcomp/kafka plugin

Performance Improvements

- **Halved** the number of MUNGE interactions by slurmctld

Flexible Node Counts

- In addition to min and max node counts, allows the user to specify acceptable node counts
  - E.g., --nodes=20,40,80,160
- Also allows for a step function specification
  - E.g., --nodes=10-30:5 is equivalent to --nodes=10,15,20,25,30

"Explicit" GRES Flag

- Currently, all GRES are allocated to a job when --exclusive is set
- New GRES Flag "Explicit" avoids allocating that GRES by default for --exclusive jobs
  - Will only allocate it when explicitly requested

Debug option handling

- New 'scontrol setdebug <level> nodes=node[1-10]' sub-command
  - Allows dynamic changes to debug level on specified nodes
- 'scontrol setdebugflags flag,flag2,flag3 nodes=node[1-10]' also added

JSON and YAML

- Greatly extended support for JSON and YAML output from user commands
- Now allows many command filtering options to be used as well

RPC Rate Limiting

- New optional per-user RPC rate limiting mechanism
  - Backs off client commands if they're being too chatty
  - Sends new dedicated response code telling the command to sleep for a second before retrying, rather than crashing the user command
  - Can avoid having 'while true; do squeue; done' overload slurmctld

Slurm 23.11 - November 2023

SlurmDBD Overhaul

- The "right-left tree" data structure was used to represent the association hierarchy in a flat row-oriented fashion
  - Unfortunately, insertion and deletion is O(n)
    - And can trigger O(n) row updates in the database
    - Which cause O(n) updates to slurmctld
  - New "lineage" approach significantly improves performance
    - Especially when heavily scripting against external accounting systems
    - Must move slurmctld to 23.11 alongside slurmdbd to see benefits
      - Otherwise slurmdbd must maintain both structures for backwards-compatibility

srun --external-launcher

- Common MPI stacks use `srun` internally to launch their own launch processes
  - `orted`, `hydra`, ...
- Newer sbatch options - such as --tres-per-task - cannot be inherited by srun without causing layout issues for mpirun/mpiexec
- New internal --external-launcher flag is automatically propagated back to srun through mpirun/mpiexec, and indicates srun is being used to bootstrap an external MPI stack
  - Provides all resources on each node to process, does not try to interpret other Slurm layout options
- Automatically injects four environment variables into job, all set to "--external-launcher":
  - OMPI_MCA_plm_slurm_args
  - PRTE_MCA_plm_slurm_args
  - HYDRA_LAUNCHER_EXTRA_ARGS
  - I_MPI_HYDRA_BOOTSTRAP_EXEC_EXTRA_ARGS

Fixing 'scontrol reconfigure'

- Ensure 'scontrol reconfigure', SIGHUP, and restarting slurmctld/slurmd processes all have equivalent results
- Currently, certain changes cannot take effect within the process through 'scontrol reconfigure', and require a process restart
  - Which changes can be safely applied through "scontrol reconfigure" currently are... unintuitive, and undocumented
  - Will finally allow for plugin changes and NodeName changes without issue

Change SlurmctldHost settings without breaking running jobs

- In Slurm 23.02 and older, changes to SlurmctldHost are not possible with jobs running on the system
  - The slurmstepd processes load their configuration when the step is launched, and have no mechanism permitting updates
  - Once a job/step completes, the slurmstepd needs to communicate directly with slurmctld... if you change the IP address of the SlurmctldHost this will fail, and running jobs will never complete
  - Change allows for slurmstepd processes to be pushed updates by slurmd automatically

Additonal HA Sanity Checks

- The "Field Notes" presentation mentioned a... *hypothetical*... issue that can happen if the StateSaveLocation is not mounted on your backup controller
  - Backup asserts control, has no job state available, and will start killing jobs off when the slurmd processes on the compute node re-register
- Backup will now check on the heartbeat file, refuse to take control if it is missing
  - Primary controller frequently updates a timestamp in the heartbeat file
    - Used to prevent backups from asserting control too aggressively in a network partition event
  - Protects against misconfiguration of StateSaveLocation, as well as an array of potential filesystem problems

New auth/slurm and cred/slurm plugins

- New internal authentication and job credential plugins
  - Alternative to MUNGE
  - Builds off existing capabilities - unix socket authentication through SO_PEERCRED (used by slurmstepd to authenticate RPCs), plus auth/jwt authentication plugin
- Simple HMAC scheme (SHA-256) built off JWT
  - Separate from existing auth/jwt plugin
  - Will require a shared key that is shared throughout the cluster
    - `/etc/slurm/slurm.key`
    - Similar security posture to MUNGE
- Will allow for future extension and flexibility...

LDAP-less control plane

- Support running the slurmctld without LDAP
  - Optional capability enabled through auth/slurm's credential extensibility
  - Username, uid, gid, groups will be captured alongside the job submission
  - auth/slurm permits the login node to securely provide these details, which auth/munge cannot due to protocol limitations

TRES Reservations

- Allow for TRES-oriented reservations
  - E.g., reserve 200 GPUs alongside 800 CPUs
- `scontrol create reservation=test start=now duration=5 account=foo tres=gres/gpu=1`

Extensible Features

- Set of key=value pairs, with the values provided by site-specific scripts
  - Can be integers, floats, or string types
  - Values refreshed periodically (on node ping)
    - Flag can mark an extensible feature as unchanging after node boot
- New job submission flag to allow users to filter the cluster nodes for suitable locations
  - Similar, but separate, from existing feature/constraint syntax
- Loosely functionally equivalent to LSF's ELIM feature
  - Not necessarily recommended for most sites

Relative QOS limits

- Flag allows QOS to be specified as a percentage of the cluster's total resources
  - Or an individual partition, if used as a PartitionQOS

Debian Packaging Support

- Provide Debian / Ubuntu package tooling
  - Packages will be under a common slurm-smd-* prefix
  - Package layout will be closer to RPM layout from slurm.spec

OpenAPI, --json/--yaml option updates

- Significant refactoring of the OpenAPI plugin code now allows for most --json/--yaml command-line options to use their filtering options
- New optional arguments allow CLI tools to provide output through a specific OpenAPI plugin version
    - Defaults to current OpenAPI schema
- See Nate's REST presentation for additional details

topology/block

- New topology/block plugin - and associated plumbing - that forcibly respects a "block" oriented topology on certain new hardware platforms
- Ensures jobs are always placed on optimal set of switches, rather than what is currently available
  - Existing topology/switch plugin is best-effort, and will launch jobs on *available* resources immediately rather than wait indefinitely for a better fit
  - Downside: system utilization can collapse if not kept in check

Soft Time Limits

- Allow a job to provide the **expected** run time in addition to the traditional hard time limit
  - Use this value for backfill planning, rather than the usual time limit
  - Increases system utilization, especially for systems with a few large jobs and a constant flow of higher-throughput
- Not recommended for most general-access systems, as users would be incentivized to submit all work with a very short soft limit to get it running immediately
  - Designed for more "cooperative" environments with a smaller user base
  - Optional, must be explicitly configured to enable

Cloud / Dynamic Nodes

- Slurm's configuration files don't have network details for the dynamic nodes
  - But commands such as srun and sdiag need to communicate directly with those nodes
  - Initial dynamic node support relied on flattening all communication by disabling fanout, and passing network details through environment variables and other means

Cloud / Dynamic Nodes

- Network changes in 23.11
  - The cloud_reg_addrs option has been removed
    - Option told slurmctld to automatically update it's address cache with the inbound IP address when slurmd registered
    - Now the default for behavior for cloud / dynamic / dynamic_future nodes
  - CommunicationParameters=NoAddrCache option removed
    - No longer needed that cloud_reg_addrs is the default
- Message Fanout
  - Fanout now works with cloud and dynamic nodes
  - Passes node addresses through dynamic tree automatically
  - Allows offload of internal bookkeeping operations (node ping, reconfigure) to the slurmd processes again, reduces network load on slurmctld

Cloud / Dynamic Nodes

- RoutePlugin=route/partition
  - Use Partitions as the boundary for message fan-out
    - Acts independently of the topology/tree plugin, which can still be used for scheduling if desired
  - Useful for multi-zone / multi-network clusters to limit potential failure propagation

Cloud / Dynamic Nodes

- Cloud InstanceId InstanceType
  - sacctmgr show instances
  - Useful to track what class of cloud hardware was used

Cloud / Dynamic Nodes

- Revamped networking - "Alias Addresses"
  - Client commands get alias addresses automatically through appropriate RPCs
    - Or through new dedicated RPC
  - Clients don't rely on older "alias_list" approach now
    - Remove SLURM_NODE_ALIASES
    - For large-scale cloud node launch, this prevents the job environment from exploding, as that variable could be massive in practice

SelectTypeParameters=CR_LL_GRES

- Similar to CR_LLN... but favor nodes with least-loaded GRES
  - CR_LLN only considers CPU occupancy
  - This allows you to steer jobs to nodes have the least occupied GPUs instead

Shards

- Shards allow for GRES (e.g., GPUs) to be cooperatively split
  - See https://slurm.schedmd.com/gres.html#Sharding for further details
  - Similar to NVIDIA's MPS, but without any specific hardware cooperation
  - No enforcement of cooperation - not recommended for most systems
- Enhancements focus on allowing a job to have shards across multiple GPUs within a single node, as well as enabling --tres-per-task to work seamlessly with these shards

… and Beyond

Slurm 24.08

ReservedCoresPerGPU

- Dedicate cores on node to GPU work
  - Cores only assigned if the corresponding GPU has been allocated to the job
  - Allows for CPU-based workloads to better overlap into GPU nodes, without threatening to starve the GPU workloads and risk idling the (expensive) GPUs
- Currently, the same use case can be partially covered by using the MaxCPUsPerNode setting on a Partition
  - But that doesn't easily scale with a heterogeneous mix of nodes, and requires splitting work across multiple partitions

Node Features

- Allow for Node Features changes without rebooting the node
  - The node_features plugin interface was originally designed for CPU NUMA/memory layout changes for the Intel KNL chips, and assumed any changes would require the node to reboot to take effect
  - But most, e.g., GPU mode changes can be done live
    - Inconvenient to need a node reboot for all changes
  - Currently required by the node_features stack, although can be faked by using the "-b" option to slurmd with some careful scripting in your RebootProgram

Further auth/slurm extensions

- Capture and send processes' SELinux context as part of the auth token

Backfill tweaks

- topology/block can lead to throughput issues under high fragmentation
  - Backfill scheduler is "conservative" in existing implementation
    - Will never stall the launch of a higher priority job, will always plan for it to start ASAP, and only then plan other jobs around it
    - With the topology strictly enforced, fragmentation can lead to considerable delays... but launching large jobs on the first available fitting set of nodes may perpetuate high-level fragmentation
  - Exploring approaches to mitigate these issues, potentially develop a heuristic that is willing to delay larger job launches in favor of reduced fragmentation, and higher utilization rates

… and even further beyond*

*if ever

Scope Limit for MPI Plugins

- Refactor the mpi plugin interface to run most hooks as the user, rather than uid 0
- CVE-2023-41915 implies we cannot always trust the MPI libraries we build against...

Standalone Step Management Layer

- Build on the isolation of the step management code (see Brian's talk from earlier)
  - Potentially allow a lightweight independent step management process to run underneath a Slurm (or other WLM) allocation
- Extend the step management layer with support for CWL or other workflow standards

Converged Computing

- Slurm cooperatively scheduling alongside other cloud orchestration layers, such as Kubernetes
- Extend/adopt official support for projects like CoreWeave's SUNK

Upcoming Events

Tim Wickberg
CTO

Upcoming Events

- SC'23 - Booth 463
- Slurm Community Birds-of-a-Feather, Thursday, Nov. 16, 12:15-13:15, Room 201-203

SLUG'24
September 2024

Open Forum

Tim Wickberg
CTO

Slurm User Group 2023

slurm | SCHEDMD


SCHEDMD
The Slurm Company