Slurm 22.05, 23.02, and Beyond

Tim Wickberg
SchedMD

https://schedmd.com

Slurm 22.05 Release

https://schedmd.com

Dynamic Nodes

- Nodes added/deleted from system without adding them into **slurm.conf**, restarting **slurmctld** or **slurmd**
- Use Cases
  - Multiple dynamic clusters, where nodes are added/removed frequently
  - Temporary addition of a new node(s)
  - Cloud services adding/removing nodes

Dynamic Nodes

- Made adding and removing nodes dynamically just work™
- Alternative to the State=CLOUD model previously used for cloud-bursting
- Only works with select/cons_tres
- New MaxNodeCount option introduced
  - Max number of nodes ever used in the cluster

Dynamic Nodes

- Two ways to add a dynamic node:
    - Dynamic registrations
        - Driven by ephemeral nodes connecting
        - Expectation is that an external control plane is creating nodes on demand
    - Created through the 'scontrol' command
        - Allows slurmctld to request their creation through the power-saving framework

Dynamic Registrations

- `slurmd -Z [--conf=<extra parameters>]`
  - "`-Z`" instructs the node to register dynamically
    - No matching NodeName= line in configuration file
  - "--conf=<extra parameters>" can be used to fill in additional NodeName details, such as GRES, or to override the detected CPUs/Cores/Sockets/Memory values

Dynamic Registrations - Examples

```
If the node hardware is using the built in command
    slurmd -C
and reports the node as
    NodeName=node1 CPUs=16 Boards=1 SocketsPerBoard=1 CoresPerSocket=8 ThreadsPerCore=2 RealMemory=31848

Then
    slurmd -Z --conf "Gres=gpu:2"
Is equivalent to a statically-configured node of
    NodeName=node1 CPUs=16 Boards=1 SocketsPerBoard=1 CoresPerSocket=8 \
        ThreadsPerCore=2 RealMemory=31848 Gres=gpu:2

Then
    slurmd -Z --conf "CPUs=16 RealMemory=30000 Gres=gpu:2"
Is equivalent to a statically-configured node of
    NodeName=node1 CPUs=16 RealMemory=30000 Gres=gpu:2"
```

Creating nodes through scontrol

- `scontrol create nodename=<name> <node options>`
  - Nodes must be State=future or State=cloud
  - Node structure is created, which can then be brought into existence through the power saving (cloud bursting) subsystems

```
> scontrol create NodeName=node[0-99] CPUs=16 Boards=1 SocketsPerBoard=1
CoresPerSocket=8  ThreadsPerCore=2 RealMemory=31848 Gres=gpu:2
State=CLOUD
```

Slurm Configuration Files

- Configless or local/shared slurm.conf still work as before
  - When using configless:
    - gres.conf - recommend “autodetect=nvml” for GPUs
    - Or use "Include /etc/my-node-gres.conf" within gres.conf to include a local file when necessary

Adding Node to Partitions

- By default dynamic nodes aren't added to any partition
- Either set "Nodes=ALL" on the Partition
  - If configured in the partition definition, the partition will always have all nodes in the partition, even new dynamic nodes

```
PartitionName=open Nodes=ALL MaxTime=INFINITE Default=Yes  State=Up
```

Adding Node to Partitions

- Or use Nodesets to allow for dynamic nodes to land in partitions automatically
    - Create nodesets, add the nodeset to the partition.
    - When registering the dynamic node, configure it with a feature to add it to the nodeset.

```
> slurmd -Z -conf="Feature=f1"
```

```
Nodeset=ns1 Feature=f1
Nodeset=ns2 Feature=f2

PartitionName=all Nodes=ALL
PartitionName=p1 Nodes=ns1
PartitionName=p2 Nodes=ns2
PartitionName=p3 Nodes=ns1,ns2
```

Deleting Dynamic Nodes

- To remove a dynamic node you must manually delete the node
    - scontrol delete nodename=<nodelist>
        - Nodes can't be deleted unless they are idle
        - Clear node from reservations
    - Stop the slurmd on the compute node

"Preferred" node constraints

- A list of optional ("soft") constraints to be considered when selecting nodes for a job
  - New "--prefer" option to salloc/sbatch/srun
  - Job launch will prefer those nodes, if possible to satisfy immediately
  - Traditional "hard" constraints (--constraint) will always be respected

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

cgroup v2 support

- Added support for cgroup v2
  - Only cgroup v1 was supported in 21.08 and older
  - Will auto-detect cgroup v1 or v2 support on the system
    - and default to v2 if available
- A number of distributions have moved to deprecate or disable v1 support, so sites are encourage to start migrating soon

Backfill for Licenses

- Licenses were previously ignored in the backfill scheduler
- By default, 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
- New SchedulerParameters=bf_licenses option enables license tracking in the backfill scheduler
    - Off by default, may be turned on in a future release

AcctGatherInterconnect/sysfs

- Add support for gathering network statistics from OmniPath, Slingshot, and other interconnects
    - Simplified this to a single "sysfs" plugin reading stats from /sys/class/net/<interface>/statistics/
    - Able to read and aggregate stats from multiple interfaces, but will consolidate into a single ic/sysfs TRES.

Changes to LLN Support

- LLN ("Least-Loaded Node") previously defined 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 - Without Defaults

- Adding a new option to SlurmDBD to allow operation without DefaultAccounts set for every user
- Not recommended for most sites, but can simplify integration scripting with external accounting systems

slurmscriptd enhancements

- Move MailProg handling into slurmscriptd
  - Significantly improves slurmctld performance on high-throughput systems

REST API

- A number of minor changes and bug fixes
- See https://slurm.schedmd.com/openapi_release_notes.html

Slurm 23.02 Roadmap

https://schedmd.com

scrun

- Directly launch OCI-compliant container images
- Can be used alongside docker tooling to handle container launch
- See separate presentation from SLUG'22 for further 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

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 especially when using licenses to represent cluster-wide resources, as they won't be reclaimed to allow higher-priority work to preempt

AllowAccounts - automatic recursion

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

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

- For jobs that have been terminated due to a node failure, explicitly save the failed node name alongside the accounting data for future triage

New job completion plugin

- Add a new jobcomp/kafka plugin

… and Beyond

Fixing 'scontrol reconfigure'

- Plans to ensure 'scontrol reconfigure', SIGHUP, and restarting slurmctld/slurmd processes all have equivalent semantics
- Currently, certain changes cannot take effect within the process through 'scontrol reconfigure', and require a process restart
  - Which these are is undocumented, and somewhat hard to intuit
- Work to simplify these paths, and allow for additional sanity checks
- Configuration check capability expected as well

Slurm and/or/vs Kubernetes?

- Separate forum discussion in the Slurm Booth
  - Tuesday @ 3:15pm
  - Wednesday @ 4:15pm

Questions?

Next Events

- SLUG'23 will be **in person**, in September 2023
  - And we'll avoid conflicting with NVIDIA GTC
- Look for announcements and call for papers on the slurm-user and slurm-announce mailing lists in the spring