Slurm Inter-Cluster Project

Stephen Trofinoff
CSCS
Via Trevano 131
CH-6900 Lugano
24-September-2014
Functionality pertaining to operations spanning different clusters is what this project refers to as “inter-cluster” features.

Slurm primarily operates on a per-cluster basis.
- The `slurmctld` is the “brain” of the system for each “cluster”.

These features are currently limited in Slurm.
- Notable example is the “-M” option to several commands such as `sbatch`.

The aim of this project is to explore some possible implementations of some such features.
Several pairs of systems where we have the main job on one system and the “post-processing” job on the other (job-chaining would be useful)

- Rosa/Julier
- Daint/Pilatus
- Albis/Lema

Partner site requesting the ability to simultaneously start jobs

Slurm community has spoken about this and related topics off and on for some time
Proposed a local site project to explore some ideas here at CSCS--initial tentative goals of the project:

1. Implement job chaining across clusters
2. Enhance commands like `squeue` and `scontrol` to display foreign jobs
   - This can currently be done only if the user knows which system the job is on
3. Implement a means of launching jobs on separate clusters simultaneously
   - A partner site wants this for streaming output from the first job to the second.
4. Enhance the current feature of submitting a job to multiple clusters
   - Currently, the decision on which it would run first is made only once
   - It would beneficial if this decision could be made repeatedly over time.
Background (cont.)

- Discussed/exchanged ideas with SchedMD and LLNL
- After several evolutions agreed upon a general scheme for having common job id’s—at least some things to try
  - This is a base feature that will be needed by all of the higher-level functionality to come.
- Further refined tentative goals of the project
Some Considerations

- Need method to determine on which cluster a job with a given id is queued
- Using a single reserved range of job id’s
- Maintain a relatively static set or grid of participating Slurm clusters
- All clusters of the grid will recognize and honor this range
- For a reserved job id, if a cluster does not have a corresponding job record it will contact the `slurmdbd` for the cluster which does
- User will designate a job as a potential inter-cluster target by use of a command-line option
- Dependencies will be created using existing Slurm syntax
isers

- interClusterJobIdStart: Denotes first job of reserved inter-cluster job range (slurmdbd.conf)
- GridClusters: List of names of Slurm clusters that can participate in the inter-cluster grid (slurmctld.conf)
- ClusterIDMode: Set to “1” to indicate that cluster will participate in the grid (slurmctld.conf)

- User
  - Specifies “--sicp” option on sbatch or salloc command line
  - Uses normal dependency clause syntax
API Example

- System Administrator edits the `slurmctld.conf` to list "ClusterA" and "ClusterB" (on both clusters) as part of the grid and that each will participate. Also edits `slurmdbd.conf` specifying the start of the inter-cluster job id range to start as 500000

```plaintext
[slurmctld.conf]
...
ClusterIDMode=1
GridClusters="ClusterA,ClusterB"
...

[slurmdbd.conf]
...
InterClusterJobIdStart=500000
...
```

- User starts a target job on ClusterA and a dependent job on ClusterB
  - ClusterA: $ sbatch --sicp ajob.sh
  - ClusterB: $ sbatch --dependency=afterok:500000 bjob.sh
Use the existing `slurmdbd` daemon to perform a few central activities
However, attempt to keep communication directly between controllers (as much as possible)
The following Slurm entities needed to be modified
  • `slurmdbd`
  • `slurmctld`
  • **Commands:**
    o `sbatch`
    o `salloc`
Stores a “grid table” correlating cluster names with the contact information of the controller
  • IP address
  • Port number
Stores the starting point, first job id, of reserved range of inter-cluster id’s
Stores table correlating used inter-cluster job id’s and the cluster to which they were assigned
Implementation (slurmdbd--Comm’s.)

- Receives “registration” request from cluster controllers
  - Add entry to its grid table for the cluster
  - Returns this updated grid cluster to ALL currently up clusters
  - Returns the starting point of the reserved range
- Receives request for new inter-cluster job id
  - Picks the next id from range
  - Creates an entry in its used job id table
  - Returns the job id to the requesting cluster
- Receives request for index of cluster where an inter-cluster job id is queued
  - Finds and returns this index
Implementation (slurmctld--Data)

- Stores value indicating if the cluster is in inter-cluster mode
- Cutoff for reserved inter-cluster job id range (received from slurmdbd)
- Stores local copy of the grid table (same as the slurmdbd)
Implementation (slurmctld--Comm’s.)

- Sends the three messages to the slurmdbd already cited
  - “registration” request of cluster
  - Request for new inter-cluster job id
  - Request for index of cluster where an inter-cluster job id is queued
- Sends message to foreign controller requesting status of a given inter-cluster job
Implementation (slurmctlId)

slurmdbd

- t1: cluster registration
- t1b: response (range start and grid table)

ClusterA

- t2b: request a IC job id ($ sbatch –sicp ajob.sh)
- t2c: response w/next IC job id (e.g. 500000)

User (shell)

- t2d: next ic job id (e.g. 500000)

ClusterB

- t3: cluster registration
- t3b: response (range start and updated grid table)

- t4b: request index of cluster with IC target (i.e. 500000)
- t4c: response with cluster index

- t4: $ sbatch –dependency=afterok:500000 bjob.sh
- t4d: next job id (normal job id—e.g. 1234)

tn & tnb: ic job state req & response
Open Questions and Other To-do’s

➢ What should be done for the status if the job record for the remote target job has already been expunged by its controller?
  • Should a counter be added to the `slurmctld’s job_record`?
  • Should a counter be added to the `slurmdbd’s table of used inter-cluster job id’s`? (Have dependent controller always contact slurmdbd for status?)
  • Should we have target controller contact `slurmdbd` if its record is already expunged?
    o Again, should the `slurmdbd` store the status in its i.c. job id table?
    o Or should it simply query the DB at this point?
Open Questions and Other To-do’s

- How long should the `slurmdbd` retain its used inter-cluster job id records?
  - Should there be a counter?
  - Should it occur when job id wrapping occurs?
  - Should it simply be after a certain fix amount of time (configurable in the `slurmdbd.conf`)?
- Currently, the `slurmdbd` must always be started before all `slurmctld`’s for the grid to work properly. This limitation should be removed.
- Doesn’t explicitly handle backup controllers
- What happens if a dependency type has multiple remote jobs on different clusters? (Needs to be tested)
- Miscellaneous pieces of the code still need to be fully developed and refined.
- Need to port initial work to newer version of Slurm—currently using old 2.5.4
- Other parts of the original work proposal still need to be explored.
Discussion

Stephen Trofinoff: stephen.trofinoff@cscs.ch