Federated Cluster Support

Brian Christiansen and Morris Jette
SchedMD LLC

Slurm User Group Meeting 2015
Background

- Slurm has long had limited support for federated clusters
  - Most commands support a “--cluster (-M)” option to route requests to different clusters
- Submitted jobs are routed to one cluster
- Each cluster operates independently
  - Job IDs on each cluster are independent (two jobs can have same ID)
  - No cross-cluster job dependencies
  - No job migration between clusters
  - No unified view of system state, each cluster largely independent
New Capabilities

● **Job Migration**
  ○ Pending jobs automatically migrated to less busy clusters

● **Fault Tolerance**
  ○ Participating clusters will take over work of a failed cluster

● **Cross-cluster Job Dependencies**

● **Unified Views**

● **Easy Administration**
  ○ Add/remove clusters to/from the federation with simple configuration change, no extra information required in database
Related Work

- Some work has been done on addressing these shortcomings in functionality, but was lacking in scalability and was never integrated into the official release.
- The major problem was the use of a single daemon to manage the mapping of job ID to cluster.
- The slurmdbd maintained a table identifying which job IDs were on each clusters.
- Slurmdbd used for job dependency testing.
- Placed very heavy load on slurmdbd to locate jobs.
Design Goals

- **Performance**: Little to no reduction in throughput of each cluster, performance scales with cluster count
- **Scalability**: No reduction in scalability of individual clusters, able to support many federated clusters
- **Fault tolerant**: No single point of failure
- **Ease of use**: Unified enterprise-wide view, minimize change in user interface
- **Stability**: No change in behavior for clusters not explicitly placed into a federation
Eliminating the Bottleneck
Eliminating the Bottleneck

- Need mechanism to identify the cluster associated with a job ID without using slurmdbd lookup
- Make use of 32-bit job ID
  - Embed cluster ID within the job ID
  - Bit 31: Flag for federated cluster job ID
  - Bits 23-30: Cluster ID (0 to 255)
  - Bits 0-22: Job ID (0 to 8,388,607)
- Unique job ID across all clusters
  - Large but unique: 2164339463 (ClusterBit + ClusterID:2 + JobID:78,599)
Job Submission

- sbatch, salloc, srun supported
- Get available clusters (IP address + port) from local slurmctld
  - Local slurmctld keeps a cache from the slurmd
  - slurmd is the backup for cluster information
- Submit a “master” job to a randomly selected cluster
  - Perform light weight check to verify job can run on cluster
- “phantom” jobs are submitted to a number of clusters
- All jobs contain job ID and the locations of all “phantom” jobs
Job Submission

MAGIC: TBD

ClusterA
ClusterB
ClusterC
ClusterD
Job Submission
Cross-Cluster Job Dependencies

- Dependencies created with standard `--dependency=` syntax
- "Master" controller will check status of remote jobs on other clusters
Fault Tolerance / Job Migration

- Controllers coordinate taking over the job if:
  - “master” controller is down for a period of time
  - Job can be started sooner
- The RPCs for jobs started on another cluster will be re-routed
- Lots of moving parts to consider and to prevent split-brain
  - Don’t have two jobs running at the same time
Fault Tolerance / Job Migration
Configuration

- Parameter identifying whether cluster is part of a federation
  - Federation=yes|no
- Parameter to control how many “phantom” jobs to spawn
  - PhantomJobs=#
- Possible to move cluster into or out of federation without Slurm restart (scontrol command)
Unified Views

- Provide unified view of federated clusters by default
- Subset of federated clusters can be requested using -M option
- `squeue`, `sinfo`, `sprio` will output cluster name in separate column
  - `squeue` can sort by cluster name

<table>
<thead>
<tr>
<th>CLUSTER</th>
<th>JOBID</th>
<th>PARTITION</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClusterA 2164339463</td>
<td>debug</td>
<td>wrap</td>
<td>brian</td>
<td>PD</td>
<td>0:00</td>
<td>10</td>
<td></td>
<td>(Resources)</td>
</tr>
<tr>
<td>ClusterA 2164329686</td>
<td>debug</td>
<td>wrap</td>
<td>brian</td>
<td>PD</td>
<td>0:00</td>
<td>10</td>
<td></td>
<td>(Priority)</td>
</tr>
<tr>
<td>ClusterB 2197893831</td>
<td>power</td>
<td>wrap</td>
<td>brian</td>
<td>PD</td>
<td>0:00</td>
<td>10</td>
<td></td>
<td>(Resources)</td>
</tr>
<tr>
<td>ClusterA 2164333731</td>
<td>long</td>
<td>wrap</td>
<td>brian</td>
<td>PD</td>
<td>0:00</td>
<td>10</td>
<td></td>
<td>(Resources)</td>
</tr>
<tr>
<td>ClusterC 2265002695</td>
<td>debug</td>
<td>wrap</td>
<td>brian</td>
<td>PD</td>
<td>0:00</td>
<td>10</td>
<td></td>
<td>(Priority)</td>
</tr>
<tr>
<td>ClusterC 2265002695</td>
<td>debug</td>
<td>wrap</td>
<td>brian</td>
<td>R</td>
<td>0:36</td>
<td>1</td>
<td>c11</td>
<td></td>
</tr>
<tr>
<td>ClusterA 2164333936</td>
<td>debug</td>
<td>wrap</td>
<td>brian</td>
<td>R</td>
<td>0:36</td>
<td>1</td>
<td>a[01-10]</td>
<td></td>
</tr>
<tr>
<td>ClusterB 2197893831</td>
<td>power</td>
<td>wrap</td>
<td>brian</td>
<td>R</td>
<td>0:36</td>
<td>1 b180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClusterC 2265002695</td>
<td>debug</td>
<td>wrap</td>
<td>brian</td>
<td>R</td>
<td>0:36</td>
<td>102</td>
<td>c[1-10,15,51-100,120-140,143-160]</td>
<td></td>
</tr>
<tr>
<td>ClusterA 2164333932</td>
<td>short</td>
<td>wrap</td>
<td>brian</td>
<td>R</td>
<td>0:39</td>
<td>1</td>
<td>a52</td>
<td></td>
</tr>
<tr>
<td>ClusterA 12345</td>
<td>short</td>
<td>wrap</td>
<td>brian</td>
<td>R</td>
<td>0:39</td>
<td>1</td>
<td>d79</td>
<td></td>
</tr>
<tr>
<td>ClusterA 2164333933</td>
<td>short</td>
<td>wrap</td>
<td>brian</td>
<td>R</td>
<td>0:39</td>
<td>1</td>
<td>a113</td>
<td></td>
</tr>
</tbody>
</table>
Schedule

- Development to begin 4Q 2015
- Included in next major release, version 16.05
Questions?