SLURM At CSCS
SLURM User Group
Barcelona, Spain
9/10-October-2012

Stephen Trofinoff
CSCS
stephen.trofinoff@cscs.ch
General SLURM Timeline

- June 2010—First working port of SLURM to a Cray
- October 2010—Install 2.x.x on Palu a development/production system
- Fall 2010–March 2011 Initial experimentation with the use of SLURM
- April 2011—CSCS goes live with SLURM on premier production system, Rosa (2.3.0-prex).
- Spring 2011–Fall 2011—CSCS begins to migrate other systems from PBS to SLURM; also ports code to Cray XK6 architecture.
- October 2011—Palu decommissioned and replaced with new XK6 named Todi—running SLURM v 2.3.0-pre5.
- November 2011—Rosa upgraded to XE6 running SLURM v. 2.3.0-pre5
- March/April 2012—Upgrade of SLURM level across most of the site to 2.3.4
- March–June 2012—Final PBS systems, Buin & Dole are replaced with Albis & Lema—both running SLURM v. 2.3.4.
General SLURM Timeline

- Spring 2012—Due to particular needs of select users, work with SchedMD to create a basic “Zero-node” job submission scheme.
- Currently, working preparing SLURM code for upcoming Cascade system due in December 2012.
- Currently, working on a basic “least-used node” selection patch for a specific set of users.
Some Past SLURM Work

- XE6 support
- XK6 support
- Zero-Node ('ZN') functionality
- GRES Accounting
- A basic SPANK auto-affinity module (based upon an older LLNL one)
- Transition procedure for preserving jobs across incompatible versions of SLURM
Current SLURM Work

- SLURM preparation for Cascade
- “Load-balancing” node selection using Least-used node.
- SigTerm vs SigKill changes to allow epilogue script to work with ALTD.
Future SLURM Work/Wish List

- Finish additional parts of GRES accounting
- More robust ZN functionality
- SLURM internal cache flush capability via scontrol
- Exclude group/user list for reservations (minor)
- Not having to specify partition when using reservation with nodes not in the default (minor)
The Systems

<table>
<thead>
<tr>
<th>Name</th>
<th>Arch. Type</th>
<th>Number of Nodes</th>
<th>Number of Processors</th>
<th>Node Layout</th>
<th>GPU</th>
<th>Node Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosa</td>
<td>Cray XE6</td>
<td>1496</td>
<td>47872</td>
<td>2x16x1</td>
<td>None</td>
<td>32GB</td>
</tr>
<tr>
<td>Todi</td>
<td>Cray XK6*</td>
<td>272</td>
<td>4352</td>
<td>1x16x1</td>
<td>1 Fermi/node 186 nodes</td>
<td>32GB</td>
</tr>
<tr>
<td>Julier</td>
<td>non-Cray</td>
<td>12</td>
<td>288</td>
<td>2x6x2</td>
<td>None</td>
<td>10-48GB, 2-256GB</td>
</tr>
<tr>
<td>Pilatus</td>
<td>non-Cray</td>
<td>22**</td>
<td>704</td>
<td>2x8x2</td>
<td>None</td>
<td>64GB</td>
</tr>
<tr>
<td>Rothorn</td>
<td>non-Cray</td>
<td>1</td>
<td>256</td>
<td>32x8x1</td>
<td>None</td>
<td>2TB</td>
</tr>
<tr>
<td>Albis</td>
<td>Cray</td>
<td>72</td>
<td>1728</td>
<td>2x12x1</td>
<td>None</td>
<td>32GB</td>
</tr>
<tr>
<td>Lema</td>
<td>Cray</td>
<td>168</td>
<td>4032</td>
<td>2x12x1</td>
<td>None</td>
<td>32GB</td>
</tr>
<tr>
<td>Castor/</td>
<td>non-Cray</td>
<td>32</td>
<td>384</td>
<td>2x6x1</td>
<td>2 per node</td>
<td>24GB</td>
</tr>
<tr>
<td>Pollux</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Arch. Type</td>
<td>Number of Nodes</td>
<td>Number of Processors</td>
<td>Node Layout</td>
<td>GPU/Node</td>
<td>Node Memory</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------------</td>
<td>----------------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Eiger</td>
<td>non-Cray</td>
<td>21</td>
<td>300</td>
<td>2x6x1</td>
<td>2 Fermi GTX480</td>
<td>24GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2x6x1</td>
<td>1 Geforce GTX285</td>
<td>24GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2x6x1</td>
<td>2 Tesla s1070</td>
<td>24GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2x6x1</td>
<td>2 Fermi c2070</td>
<td>24GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2x12x1</td>
<td>2 Fermi m2050</td>
<td>48GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2x12x1</td>
<td>2 Fermi c2070</td>
<td>48GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2x6x1</td>
<td>1 Geforce GTX285</td>
<td>48GB</td>
</tr>
</tbody>
</table>
The Systems Miscellaneous

- Ela—main gateway from outside to systems
- Fojorina01/Fojorina02—Hosts the common slurmdbd for all principal systems
- db.cscs.ch—hosts the central CSCS DB and SLURM DB
The Test Systems

- Gele—A Cray 16-2x16x1-node system w/32GB per node
- Dolomite—set of non-Cray blades, currently using 4-2x6x2 nodes each with ~11GB
- VM's with emulators for XK6 and Cascade
- Use their own DB.
SLURM Features Used

- Basics
  - partition/node configuration options
  - Cray and Cons_res modules for node selection
  - Backfill scheduler
  - Priority multifactor

- Additional
  - Lua scripts (job submission policy enforcement and group priority)
  - Task/affinity
  - GRES (some systems)
  - Accounting (via slurmdbd and MySQL DB)
  - Zero-Node ('ZN') jobs for post-processing (some systems)
  - Various Prologues/Epilogues (some systems)

- Contribs/Misc
  - PAM module (some systems)
  - SPANK module (some systems)

- Some User/Admin features
  - Advanced reservations
  - Job Chaining
SLURM Features Not Used

- Fairshare
- QOS
- Gang Scheduling
- Preemption
- Command wrappers on Cray
## SLURM By System

<table>
<thead>
<tr>
<th>System</th>
<th>SLURM</th>
<th>Sched</th>
<th>Select</th>
<th>Prolog</th>
<th>Epilog</th>
<th>Lua</th>
<th>Priority</th>
<th>ASE</th>
<th>Task/Affin</th>
<th>SPANK</th>
<th>PAM</th>
<th>Accounting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosa</td>
<td>2.3.4</td>
<td>*1,2</td>
<td>backfill</td>
<td>Cray</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>limits</td>
<td>No</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>Todi</td>
<td>2.3.4</td>
<td>*1-5</td>
<td>backfill</td>
<td>Cray</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>Not set</td>
<td>No</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>Julier</td>
<td>2.3.4</td>
<td></td>
<td>backfill</td>
<td>Cons_res</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>limits</td>
<td>Sched</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>Pilatus</td>
<td>2.3.4</td>
<td>*1-4</td>
<td>backfill</td>
<td>Cons_res</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>Assoc.</td>
<td>Sched</td>
<td>Auto Binding</td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>Eiger</td>
<td>2.3.4</td>
<td></td>
<td>backfill</td>
<td>Cons_res</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>Not set</td>
<td>Sched</td>
<td></td>
<td>Yes</td>
<td>slurmdbd</td>
</tr>
<tr>
<td>Rothorn</td>
<td>2.3.4</td>
<td></td>
<td>backfill</td>
<td>Cons_res</td>
<td>Task</td>
<td>No</td>
<td>multifactor</td>
<td>limits</td>
<td>Cpusets</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
</tbody>
</table>
## SLURM By System (Continued)

<table>
<thead>
<tr>
<th>System</th>
<th>SLURM</th>
<th>Sched</th>
<th>Select</th>
<th>Prolog</th>
<th>Lua Epilog</th>
<th>Priority</th>
<th>ASE</th>
<th>Task Affinity</th>
<th>SPANK</th>
<th>PAM</th>
<th>Accounting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albis</td>
<td>2.3.4</td>
<td>backfill</td>
<td>Cray</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>limits</td>
<td>No</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>PPAlbis</td>
<td>2.3.4</td>
<td>backfill</td>
<td>Cons_res $$$</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>limits</td>
<td>No</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>Lema</td>
<td>2.3.4</td>
<td>backfill</td>
<td>Cray</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>limits</td>
<td>No</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>PPLema</td>
<td>2.3.4</td>
<td>backfill</td>
<td>Cons_res $$$</td>
<td>Task</td>
<td>Yes</td>
<td>multifactor</td>
<td>limits</td>
<td>No</td>
<td></td>
<td></td>
<td>slurmdbd</td>
</tr>
<tr>
<td>Castor Pollux</td>
<td>2.3.1</td>
<td>backfill</td>
<td>Cons_res $$$</td>
<td>Task</td>
<td>No</td>
<td>multifactor</td>
<td>N/A</td>
<td>Sched</td>
<td></td>
<td></td>
<td>None</td>
</tr>
</tbody>
</table>
SLURM 2.3.4 Patches In Use

1. Basic Cray XK6
2. Node Memory—Fix bug with "--mem" option on Cray systems.
3. Zero-Node patches—provides limited ability to run "post-processing" jobs on front ends.
4. GRES Count Underflow
5. sacct -N fix
How CSCS Uses SLURM

- SLURM “ecosystem” consists of SLURM and various scripts and utilities built around it
- This ecosystem interacts with the site's general DB—user id's, accounts, allocations
- All systems with accounting have the following cron scripts
  - [cluster]_populate.pl
  - [cluster]_accounting.pl
  - set_priority_pg
How CSCS Uses SLURM

- Time allocations and the associated users granted access by Project Office
- Information stored in the central CSCS DB
- The SLURM “ecosystem” exchanges info w/CSCS DB at several spots including
  - Cron scripts
  - User scripts
- Originally, allocations were on a per-cluster basis
- Now, via our scripts and DB, we provide a “common allocation” that can be/are consumed across various systems
  - CPU hours are weighted depending upon machine/node type
  - Weights can be easily modified over time as needed via DB
SLURM Job Priorities at CSCS

- CSCS maintains the concept of a group priority
- Implemented via cron and lua scripts
- Uses the “nice” value component of multifactor priority equation
- Equation for the group priority factor:
  - group_priority(within budget) = weight_nice*(used/quota – time_factor)
  - group_priority(over budget) = weight_nice*over_used + time_factor*penalty
  - Where:
    - weight_nice = 1000 (constant)
    - time_factor = (now - start_time)/(end_time – start_time)
    - over_used = MAX(used/quota,5.0)
    - Penalty = [based upon project type where]

<table>
<thead>
<tr>
<th>Project Type</th>
<th>alps</th>
<th>lup</th>
<th>cp</th>
<th>small</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalty</td>
<td>10</td>
<td>100</td>
<td>100</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
SLURM Job Priorities at CSCS

- Maintain concept of local and global usage for a given group
- Most systems, deny jobs from over-budget accounts
- On Rosa, “bottom-feeding” jobs allowed (over-budget but have the lowest of possible priorities)
- Use cron script to periodically update priorities for pending jobs
Julier Limits

- Additional limits were enacted for Julier
- MaxCPUs, MaxJobs, MaxSubmit
- Partition definition doesn't handle these
- Used association records instead
- Each user winds up having one record for each partition of each account to which they belong on Julier
Automatic Binding on Pilatus

- Some internal users demanded a simplified automatic binding of a specific pattern
- Solution—created a SPANK module
- Started with old auto-affinity LLNL module, stripped it down and then changed some of the logic
- Binding is now as follows:
  - Only 1 HW thread (vCPU)/core is used
  - Fill across a socket before using next socket
  - Tasks must fit on a single socket (No crossing boundaries)
  - Can have more than one task per socket if all these tasks fulfill above condition
  - If not enough sockets exist to place all tasks of the node completely, according to above rules, job is rejected.
Some Challenges Along the Way

- Dropped Cluster Problem (due to packets from controller to slurmd being too large)
- Unable to launch jobs when only one FE is down (August 2011)
- SLURM & ALPS out-of-sync causing SLURM to get stuck (August/September 2011). SchedMD fixed this.
- Various instances of garbage being written to various fields in the SLURM DB. Had to manually fix some DB entries in some cases.
- Backfill not processing enough jobs at a time on Rosa (Fall 2011?) SchedMD provided a patch that fixed this.
- Understanding proper use of various affinity options (especially at software thread level).
Summary

- CSCS has diverse array of small to mid-size systems
- Successfully manage these resources with SLURM
- Maintain a SLURM “ecosystem” of SLURM instances, DB and scripts to provide both CSCS and users with the desired resource management functionality