SLURM Operation on Cray XT and XE

Morris Jette
jette@schedmd.com

SchedMD LLC
Contributors and Collaborators

This work was supported by the Oak Ridge National Laboratory Extreme Scale Systems Center.

Swiss National Supercomputing Centre performed some of the development and testing.

Cray helped with integration and testing.
Outline

• Cray hardware and software architecture
• SLURM architecture for Cray
• SLURM configuration and use
• Status
Cray Architecture

• Many of the most powerful computers built by Cray
• Nodes are diskless
• 2 or 3-dimension torus interconnect
  • Multiple nodes at each coordinate on some systems
• Full Linux on front-end nodes
• Lightweight Linux kernel on compute nodes
• Whole nodes must be allocated to jobs
ALPS and BASIL

• ALPS – Application Level Placement Scheduler
  • Cray's resource manager
  • Six daemons plus variety of tools
    – One daemon runs on each compute node to launch user tasks
    – Other daemons run on service nodes
  • Rudimentary scheduling software
    – Dependent upon external scheduler (e.g. SLURM, etc) for workload management

• BASIL – Batch Application Scheduler Interface Layer
  • XML interface to ALPS
SLURM Architecture for Cray

• Many tools dependent upon ALPS
  • Use SLURM as scheduler layer above ALPS and BASIL, not a replacement
SLURM and ALPS Functionality

- **SLURM**
  - Prioritizes queue(s) of work and enforces limits
  - Decides when and where to start jobs
  - Terminates job when appropriate
  - Accounts for jobs

- **ALPS**
  - Allocates and releases resources for jobs
  - Launches tasks
  - Monitors node health
SLURM Architecture for Cray (Detailed)

- **Slurmctld**
  - (SLURM controller daemon)
  - (primary or backup)
  - Coordinates all activities

- **Slurmd**
  - (SLURM job daemons)
  - (Active on one or more service nodes)
  - Runs batch script

- **BASIL**
- **ALPS**
- **Compute Nodes**
Job Launch Process

1. User submits script

Slurmctld
(SLURM controller daemon)
(primary or backup)
Coordinates all activities

Slurmd
(SLURM job daemons)
Runs batch script

BASIL
ALPS
Compute Nodes
Job Launch Process

1. User submits script
2. Slurmctld creates ALPS reservation

Slurmctld (SLURM controller daemon) (primary or backup) Coordinates all activities

Slurmd (SLURM job daemons) Runs batch script

- BASIL
- ALPS
- Compute Nodes
1. User submits script

2. Slurmctld creates ALPS reservation

3. Slurmctld sends script to slurmd

Slurmctld (SLURM controller daemon) (primary or backup)
Coordinates all activities

Slurmd (SLURM job daemons)
Runs batch script

BASIL
ALPS
Compute Nodes
Job Launch Process

1. User submits script
2. Slurmctld creates ALPS reservation
3. Slurmctld sends script to slurmd
4. Slurmd claims reservation for specific session ID and launches interpreter for script

Slurmctld
(SLURM controller daemon)
(primary or backup)
Coordinates all activities

Slurmd
(SLURM job daemons)
Runs batch script

BASIL
ALPS
Compute Nodes

#!/bin/bash
srun a.out
1. User submits script
2. Slurmctld creates ALPS reservation
3. Slurmctld sends script to slurmd
4. Slurmd claims reservation for specific session ID and launches interpreter for script
5. aprun (optionally using the srun wrapper) launches tasks on compute nodes

#!/bin/bash
srun a.out
Almost all Cray-specific logic is in a Resource Selection plugin (as SLURM does for IBM BlueGene systems). The select/cray plugin in-turn calls the select/linear plugin to provide full-node resource allocation support including job preemption, memory allocation, optimized topology layout, etc.
SLURM's select/cray plugin

**Select/cray**

- **Select/linear**
- **Libalps (SLURM module)**
- **BASIL**
- **ALPS (via MySQL)**

**Libemulate (SLURM module)**

**OR**

- Emulates BASIL and ALPS for development and testing
- Emulate any size Cray on any test system
- To use, build SLURM with configure option of –with-alps-emulation
# Sample slurm.conf file for Cray system
# Selected portions

SelectType=select/cray  # Communicates with ALPS

FrontEndName=front[00-03]  # Where slurmd daemons run
NodeName=nid[00000-01023]
PartitionName=debug Nodes=nid[00000-00015] MaxTime=30
PartitionName=batch Nodes=nid[00016-01023] MaxTime=24:00:00
**sr**un Command

- SLURM has **sr**un command for Cray systems that is a wrapper for both **salloc** (to allocate resources as needed) and **aprun** (to launch tasks)
- Options are translated to the extent possible
- Build SLURM with configure option –*with-srun2aprun* to build wrapper
  - Otherwise **sr**un command advises use of **aprun** and exits
srutn Options

- If no allocation exists, the srutn options are translated directly to salloc options to create a job allocation
  - Many srutn option only apply when a job allocation is created
- After an allocation is created
  - Most common options are translated from srutn to aprun (task count, node count, time limit, file names, support for multiple executables, etc.)
  - Some srutn options lack aprun equivalent and vice-versa
    - srutn's “–alps=” option can pass any other options to aprun
  - SLURM environment variables are not currently set
  - There are fundamental differences in I/O
    - For example, ALPS does not support per-rank I/O streams
sview of Emulated System

<table>
<thead>
<tr>
<th>Name</th>
<th>NodeAddr</th>
<th>NodeProtoName</th>
<th>State</th>
<th>CPU Count</th>
<th>Used CPU Count</th>
<th>Cores</th>
<th>Sockets</th>
<th>Threads</th>
<th>Real Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>rd000000</td>
<td>070</td>
<td>c5-Ocs57n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00001</td>
<td>060</td>
<td>c5-Ocs57n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00002</td>
<td>160</td>
<td>c5-Ocs57n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00003</td>
<td>170</td>
<td>c5-Ocs57n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00004</td>
<td>171</td>
<td>c5-Ocs66n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00005</td>
<td>161</td>
<td>c5-Ocs66n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00006</td>
<td>061</td>
<td>c5-Ocs66n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00007</td>
<td>071</td>
<td>c5-Ocs66n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00008</td>
<td>072</td>
<td>c5-Ocs66n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00009</td>
<td>073</td>
<td>c5-Ocs66n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00010</td>
<td>173</td>
<td>c5-Ocs66n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00011</td>
<td>172</td>
<td>c5-Ocs55n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00012</td>
<td>162</td>
<td>c5-Ocs54n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00013</td>
<td>163</td>
<td>c5-Ocs54n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00014</td>
<td>063</td>
<td>c5-Ocs54n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00015</td>
<td>062</td>
<td>c5-Ocs54n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00016</td>
<td>052</td>
<td>c5-Ocs63n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00017</td>
<td>053</td>
<td>c5-Ocs63n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00018</td>
<td>043</td>
<td>c5-Ocs63n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00019</td>
<td>042</td>
<td>c5-Ocs63n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00020</td>
<td>142</td>
<td>c5-Ocs63n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00021</td>
<td>143</td>
<td>c5-Ocs63n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00022</td>
<td>133</td>
<td>c5-Ocs63n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00023</td>
<td>132</td>
<td>c5-Ocs62n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00024</td>
<td>131</td>
<td>c5-Ocs61n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00025</td>
<td>091</td>
<td>c5-Ocs61n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00026</td>
<td>041</td>
<td>c5-Ocs61n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00027</td>
<td>141</td>
<td>c5-Ocs61n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00028</td>
<td>140</td>
<td>c5-Ocs60n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00029</td>
<td>040</td>
<td>c5-Ocs60n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00030</td>
<td>050</td>
<td>c5-Ocs60n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00031</td>
<td>150</td>
<td>c5-Ocs60n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00032</td>
<td>250</td>
<td>c4-Oc2x7n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00033</td>
<td>350</td>
<td>c4-Oc2x7n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00034</td>
<td>340</td>
<td>c4-Oc2x7n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00035</td>
<td>240</td>
<td>c4-Oc2x7n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00036</td>
<td>241</td>
<td>c4-Oc2x6n3</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00037</td>
<td>341</td>
<td>c4-Oc2x6n2</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00038</td>
<td>351</td>
<td>c4-Oc2x6n1</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00039</td>
<td>251</td>
<td>c4-Oc2x6n0</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
<tr>
<td>rd00040</td>
<td>252</td>
<td>c4-Oc2x5n5</td>
<td>idle</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2000M</td>
</tr>
</tbody>
</table>
sview of Emulated System
Caveats

- Some SLURM functionality has no ALPS equivalent
  - Independent I/O by task
  - Output labeled by task ID
- Some ALPS options have no SLURM equivalent
  - `srun` wrapper has –`alps` option to pass arbitrary arguments to `aprun`
- Some options are similar, but impossible to directly translate
  - Task binding syntax
  - Per-task vs. per-CPU limits
• SLURM environment variables are not set
  • Many need to be set on a per-node or per-task basis, so ALPS must do this
  • Under development by Cray
Caveats (continued)

- SLURM GUIs (actually the *curses* and GTK libraries they use) have limited scalability
  - Scales to a few thousand nodes
  - Currently each position displayed represents a unique X, Y, Z-coordinate
  - If multiple nodes share an X, Y, Z-coordinate, the information for only one node is displayed
  - We found this better than displaying each node independently and providing confusing topology information, but could change this if desired
Status

• SLURM has been running reliably over ALPS at Swiss National Supercomputer Centre since April 2011

• Validated by Cray in July/August 2011