

# SLURM at USF!

SLUG 2015 Site Report  
Presented by: John DeSantis



# Research Computing University of South Florida

## **Research Computing provides:**

HPC systems administration  
Software support  
Grant writing and support  
Direct consultation

## **Providing services to:**

6 colleges  
40 research groups  
Over 500 active users



# Research Computing SLURM cluster stats

## CIRCE cluster

- Built on the condominium model
- 487 nodes
- 6,572 cores
- 6 partitions
- Active DDR and QDR Infiniband fabrics
- Active GPU devices in use:
  - Tesla T10 (2)
  - Tesla M2070's (8)
  - Tesla Kepler K20m's (84)
- 789,807 jobs ran in last 90 days



# Motivation to switch scheduler

Long delays in the previous scheduler job dispatch cycle

Constantly high CPU usage for previous scheduler's master process

Daily reservations needed to be scheduled via cron and not guaranteed

Serial requests created fragmentation in resources

“**Guess work**” as to why jobs are pending despite free resources

Jobs accepted without resource verification

Priorities assigned by scheduler difficult to explain

# With SLURM

Job dispatch times significantly reduced

No spike in CPU on scheduling cycles

SLURM has a *daily* reservation option

SLURM's scheduling parameter “**pack\_serial\_at\_end**” eliminated resource fracturing that we were seeing

Job priority calculation less complicated

SLURM preemption based upon QOS works

Installation and upgrades are as easy as documented

No “**guess work**” with pending jobs

***Impossible*** resource requests not accepted



# SLURM accounting and QOS on CIRCE

Associations and QOS **together** allow:

- Easy user administration
- Easy priority administration

Accounts are mapped according to a “default” QOS

Default QOS is determined by user's affiliation to USF

- Limited, normal, faculty, and contributor

normal	user1	normal	normal
normal	user2	contributor	contributor,normal
faculty	user3	faculty	contributor,faculty
hiibaylor	user4	hiibaylor	hiibaylor
normal	user5	normal	normal

- Base priority, grace time, limits, & preemptor status all depend on assigned QOS

normal	1000	01:00:00	512	2000	cluster	DenyOnLimit
faculty	2000	01:00:00			cluster	
contributor	3000	01:00:00	1024	2000	cluster	DenyOnLimit
hiibaylor	5000	00:00:00			contributor,faculty,normal	cluster

# SLURM accounting continued

QOS can also function as a “quick” ACL:

- “badusers” QOS with 0 jobs and 0 CPU's
- DenyQOS=badusers on Partition definition for permanence
- Use scontrol update partitionname=blah denyqos=badusers

Preemption via QOS:

- Allows preemptor & preemptee changes on the fly
- Grace time makes preemptable partitions usable by non-owners

# General configuration of SLURM scheduling

Active primary and backup controllers

Topology “tree” plugin utilized

Active QOS flag “Deny on Limit” for maximum CPU's and submitted jobs

- Depends on “default” QOS

PriorityType=priority/multifactor to weigh jobs based upon:

1) QOS, 2) Partition, 3) Age, and 4) Job size

SelectType=con\_res with CR\_CPU\_MEMORY parameters

Favor small jobs relative to time

SchedulerType=sched/backfill with scheduling parameters:

- bf\_continue
- bf\_max\_job\_user=5
- bf\_max\_job\_test=500
- kill\_invalid\_depend
- pack\_serial\_at\_end

DefaultTime set to 1 hour on all partitions

HealthCheck for compute nodes:

- “Cycled” every 5 minutes
- Performed on all states for self healing

Preemption mode is defined as CANCEL so grace times can be utilized





# Notable Implementation “issues” and solutions

\*SLURM node(s) reporting “LowRealMemory”

SOLUTION: Use what 'free' reports.

SLURM UID not consistent during initial installs across nodes.

SOLUTION: Used puppet module to create user with specific UID.

\*SLURMD limits not appropriate for computational purposes.

SOLUTION: 'PropagateResourceLimits=NONE' and addition of slurmd ulimits.

Orphan process cleaning frequently failing with timeout errors

SOLUTION: Adjustment of MessageTimeout parameter

A single partition spread over heterogeneous resources?

SOLUTION: Use of SLURM's Topology/tree plugin.

GPU partition without keeping nodes idle?

SOLUTION: Use of an extra partition and QOS priority

09/15/2015: Obsoleted by TRES and partition QOS (Thanks, Danny...)

Ensure “On demand” classroom access to resources?

SOLUTION: Partitioning with Accounts

\* Should have RTFM!

# Research Computing SLURM history

## • **Late July 2014**

SLURM installed onto 27 nodes (1 rack) utilizing least-used hardware.

SlurmDBD installed and utilized.

User and host groups created for SLURM “alpha” access and users recruited.

## • **August 2014**

Upgraded SLURM 14.03.3 to 14.03.6 due to bug with interactive sessions.

Added a GPU node for testing SLURM's GRES.

Added 22 more nodes to the alpha cluster.

Added SLURM RPM's to our repositories for ease of provisioning.

Corrected an issue with SLURM user being created with different UID.

Created puppet module for SLURM for ease of deployment.

Added a login node for the alpha cluster.



# “Quick” Research Computing SLURM history continued

## • **September 2014**

Configured and experimented with SLURM's QOS and preemption capabilities.  
Added 21 more nodes to the alpha cluster, one of which chosen as a back-up controller.  
Reorganized partitions and tested “production” preemption.

## • **October 2014**

Testing of SLURM's 'sacctmgr's “dump” and “load” features.  
Tested and utilized node weighting.  
Ready for production

## • **September 2014 – late February 2015**

Notified users of upcoming production change to SLURM for Spring Semester  
Upgraded SLURM 14.03.6 to 14.11.3  
User education and documentation conversion

## • **March 2015**

Migration to SLURM as production scheduler completed on March 3, 2015, while dealing with a RAID 6 failure on a production Lustre file system!



**Questions?  
Comments?**

**Thank you SchedMD!**

