SLURM: Resource Management from the Simple to the Sophisticated
SLURM User Group 2010
October 2010

Morris Jette (jette1@llnl.gov)
Danny Auble (auble1@llnl.gov)

S&T Principal Directorate - Computation Directorate
The Genesis of SLURM

- In late 2001, Lawrence Livermore National Laboratory began an open source strategy for high performance computing.

- Two major software components were lacking:
  - Parallel file system: Lustre
  - Resource manager: SLURM

- Primary design goals:
  - Scalable to >10k nodes, >100k processors
  - Highly portable
State of Computer Scheduling at LLNL in 2001

- DPCS (Distributed Production Control System, LLNL project)
  - RMS (Resource Management System, Quadrics product)
  - LoadLeveler (IBM product)
  - Proprietary resource managers
SLURM Design Begins

- Design and development began in 2002
  - 2.0 designers/developers at LLNL
  - 0.3 designer/developer at Linux NetworX
  - Extensive experience with job scheduling and distributed computing

- Quadrics RMS used as a reference model
  - Simple and worked well
  - Not open source, portable or sufficiently scalable

- DPCS to decide where and when to start jobs
Initial Design Decisions

- **Portability**
  - Use plugins extensively for alternative implementations (network, MPI, authentication, etc.)

- **Scalability**
  - Highly multi-threaded
  - Independent read and write locks by data structure
  - Fault-tolerance: No single point of failure
  - Node name expressions: “linux[0-1023]”

- **GNU Public License (GPL version 2)**
First Deployments in 2003

- Very fast, but very simple
  - First-In First-Out scheduling (relied upon DPCS to prioritize work, backfill scheduling, etc)
  - Schedule whole nodes only
  - Supported Linux with either Ethernet or Quadrics interconnect
  - No accounting
  - 64k Lines Of Code
Procurement of ASCI Purple (IBM SP) and BlueGene/L systems causes two simultaneous major SLURM porting efforts in 2004 and 2005
- IBM SP used AIX and IBM Federation switch plus IBM-specific tools
- BlueGene/L used 3-D torus interconnect and custom management interface from IBM
- Many plugins added
- 76,000 lines of code added in two years
Moab Selected as Job Scheduler in 2006

- Major effort to thoroughly integrate SLURM and Moab

- Due to differences between DPCS and Moab, SLURM not only needed to be integrated with Moab, but functionality needed to be added
  - Moab’s user management database (Gold) and tools were not sufficiently scalable for LLNL
  - SlurmDBD, sacct, and sacctmgr were developed as replacements for Gold
  - Job accounting added to SLURM
Leverage existing database

- Job prioritization plugin
- Many resource limits by user/bank

Advanced reservations

Resource allocations optimized for network topology

Power down idle nodes and restart on demand
SLURM and the Grid
Version 2.2, Release Scheduled Late 2010

- SLURM commands operate between clusters, even of different architectures (e.g. status a BlueGene/L from a traditional Linux cluster)

- Scheduling of generic resources (e.g. GPUs)

- Major improvements for high-throughput computing
  - Throughput of >120,000 jobs per hour
SLURM Job Scheduling in 2010

SLURM (Cluster 1) ➔ SLURM (Cluster 2) ➔ ... ➔ SlurmDBD (SLURM Database Daemon) ➔ MySQL
Where is SLURM Today

- Running on many of the largest computers in the world
- An attractive alternative to commercial schedulers
  - Scalable and powerful
  - Open source and freely available
  - Under active development
  - Actively supported
- Contributions from about 70 people: LLNL, BSC, Bull, CEA, HP, NUDT, etc.
SLURM Timeline

- 2001: Decision to begin project
- 2003: First deployments at LLNL, 64k Lines Of Code
- 2005: Deployed on IBM/SP with AIX plus BlueGene/L
- 2007: Added database for user/bank management and accounting records
- 2007: Fully integrated with Moab
- 2009: Sophisticated scheduling mechanism added
- 2010: Managing resources on the grid, deployed on BlueGene/P, Cray XT and Cray XE
- 2011: Deploy on BlueGene/Q
Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacture, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.