Lawrence Livermore National Laboratory

LLNL HPC Resource Management Site Report to the 2011 SLURM User Group Meeting

September 23, 2011

Don Lipari
Livermore Computing

Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94551

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

LLNL-PRES-498631

Agenda

- Background
- User Expectations
- Configuration
- Management

Background

- Livermore Computing (LC) is the department devoted to providing HPC services to customers from within and outside LLNL
- LC operates 24x7x52
- Staff include:
  - Operations Staff
  - Hotline Consultants
  - System Administrators
  - Teams of Developers / Support
    - CHAOS
    - Lustre
    - HPSS
    - Batch scheduling / resource management

History of LC Batch Systems

- The DPCS batch scheduler was developed by LLNL and used throughout the ‘90’s and up until 2007
  - Provided a uniform user interface to a disparate collection of clusters each running its own resource manager (NQS, RMS, LoadLeveler)
  - Pioneered grid and fair-share scheduling principles as well as the standby and expedite job services
  - DPCS became LCRM in 2002
- SLURM started development in 2002 and over the ensuing years supplanted all the other resource managers used on LC systems
- Moab was selected to replace LCRM in 2006 to provide a uniform environment across the NNSA labs (LLNL, LANL and Sandia)

User Expectations

- Their jobs will be scheduled at the soonest available time subject to
  - queued and running jobs from other users
  - a well defined policy for determining job priority and preemption
- The job will launch and run reliably across the allocated resources
- Users will be able to obtain an informative status of their job when queued, running, and weeks later after completion
- Users and managers will be able to retrieve reports of job usage statistics and machine utilization
- Most users invest minimal effort in learning / understanding the batch scheduling system

LC Environment

- Linux commodity clusters
- Blue Gene / L,P,Q
- Uniformity / consistency
- OS remains fixed with optional environments selectable by module or dotkit.
- No external cloud connection
- Wrappers were written to emulate legacy (DPCS/LCRM) job scripts
- Job usage and machine utilization is tracked and reported periodically to sponsors

Queues

- All clusters are typically partitioned into two queues
  - batch queue (the default) for the majority of nodes
    - longer wallclock time and larger job size limits
  - debug queue for fast turnaround
    - more restrictive wallclock time and job size limits
    - cron jobs extend the queue limits for evening and weekend hours

Two methods for a job to learn of its impending termination

- Poll for remaining time:
  - SLURM's **`slurm_get_rem_time()`** API
  - libyogrt's **`yogrt_remaining()`** API (most accurate)
- Request a signal when the wallclock limit is imminent:

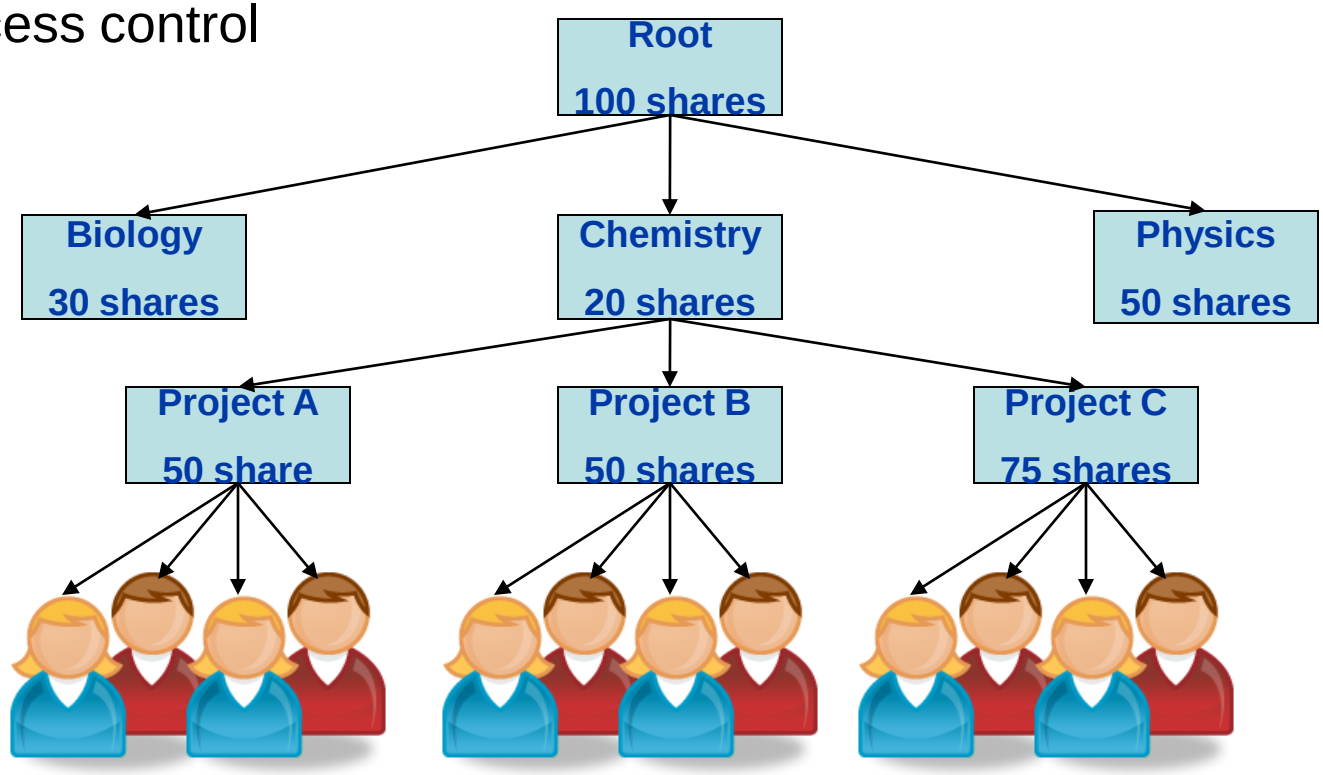
  `sbatch --signal=<sig_num>[@<sig_time>]`
- Default is for SLURM to send job SIGTERM when time expires, wait KillWait (30) seconds, and then send job SIGKILL

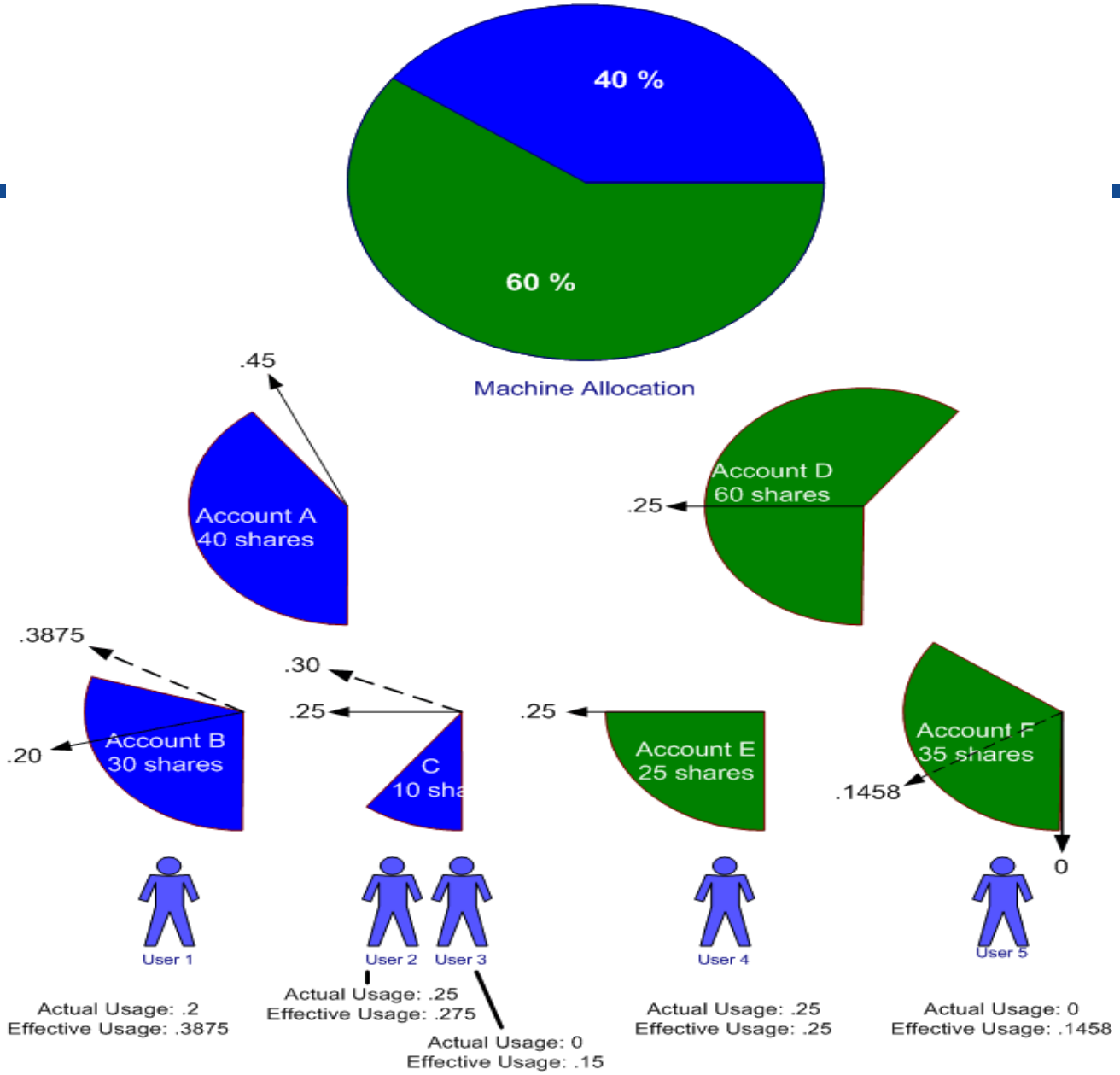
Fair-share scheduling

- One of many factors in determining a job's priority
- The project is promised a minimum portion of a cluster's cpu cycles assuming the project members submit enough jobs to keep the cluster busy
- Provides optimal machine utilization
  - Users from projects that are over-subscribed can still submit jobs and have them run when a cluster would otherwise go idle
  - No need for hard allotments

Bank Account / User Hierarchy

- Independent limits can be assigned on each account
- Form the structure to the fair-share hierarchy
- Access control




40 %
60 %
Machine Allocation
.45
Account A
40 shares
Account D
60 shares
.25
.3875
.20
Account B
30 shares
.30
.25
C
10 sha
.25
Account E
25 shares
Account F
35 shares
.1458
0
User 1
User 2
User 3
User 4
User 5
Actual Usage: .2
Effective Usage: .3875
Actual Usage: .25
Effective Usage: .275
Actual Usage: 0
Effective Usage: .15
Actual Usage: .25
Effective Usage: .25
Actual Usage: 0
Effective Usage: .1458
represents actual usage
represents effective usage

sshare output

Account	User	Raw Shares	Norm Shares	Raw Usage	Effectv Usage	FairShare
root			1.000000	1637823540	1.000000	0.500000
science		990	0.990000	1335402198	0.254931	0.620335
biology		30	0.297000	445134066	0.254931	0.620335
plants		10	0.148500	445134066	0.128258	0.611243
animals		10	0.148500	0	0.016032	0.611250
chemistry		20	0.198000	445134066	0.022608	0.499494
carbon		50	0.152308	0	0.016032	0.611250
helium		5	0.015231	445134066	0.022608	0.499494
sodium		10	0.030462	0	0.016032	0.611250
physics		50	0.495000	0	0.016032	0.611250
force		28	0.210000	0	0.016032	0.611250
mass		28	0.210000	0	0.016032	0.611250
acceleration		10	0.075000	0	0.005554	0.620322
overhead		10	0.010000	302421342	0.005106	0.650281
guests		2	0.005000	98323896	0.002349	0.371655
lc		2	0.005000	204097446	0.003778	0.203534
lc	lipari	1	0.005000	131	0.000044	0.200853

sprio output

JOBID	USER	PRIORITY	AGE	FAIRSHARE	JOBSIZE	PARTITION	QOS	NICE
1006316	robert	500912	50	361	500	1	500000	0
1006429	lucy	500548	43	4	500	1	500000	0
1006460	william	500679	42	136	500	1	500000	0
1006461	william	500679	42	136	500	1	500000	0
1006462	william	500679	42	136	500	1	500000	0
1006463	william	500679	42	136	500	1	500000	0
1006656	ruth	501941	30	910	1000	1	500000	0
1006935	robert	500883	20	361	500	1	500000	0
1006939	stephan	502021	20	0	2000	1	500000	0
1007144	jessie	500517	12	3	500	1	500000	0
1007145	jessie	500517	12	3	500	1	500000	0
1007367	gee	500645	6	138	500	1	500000	0
1007442	jessie	500513	8	3	500	1	500000	0
1007464	lucy	500511	6	4	500	1	500000	0
1007479	stephan	502002	1	0	2000	1	500000	0
1007530	judy	551376	1	50874	500	1	500000	0

Quality of Service (QOS)

- Normal
  - the default QOS
  - nominal job size and wallclock time limits
  - normal job priority
- Standby
  - no job size or wallclock time limits
  - NoReserve flag set allows any standby job to run if it can
  - subject to preemption when non-standby jobs are submitted to the queue
  - assigned a much lower job priority relegating standby jobs to the bottom of the queue
- Exempt
  - no job size or wallclock time limits
  - normal job priority
- Expedite
  - no job size or wallclock time limits
  - assigned a much higher job priority placing jobs to the top of the queue

Roles

- Users
- Bank Account Coordinators
  - allowed to maintain membership and limits of their own bank accounts
- Operators / Hotline Consultants
  - can modify users' jobs
- System Administrators
  - Establish system configuration
  - Responsible for system availability and performance
- "Sales" Team
  - Sell shares of a cluster to PI's
  - Define the bank account hierarchy - shares and limits
  - Held accountable that the shares purchased are actually delivered

Basic scheduler / resource manager interaction

Jobs

Status

- Determines job priorities
- Implements scheduling policy
- Enforces limits
- Workload management

Moab

Jobs

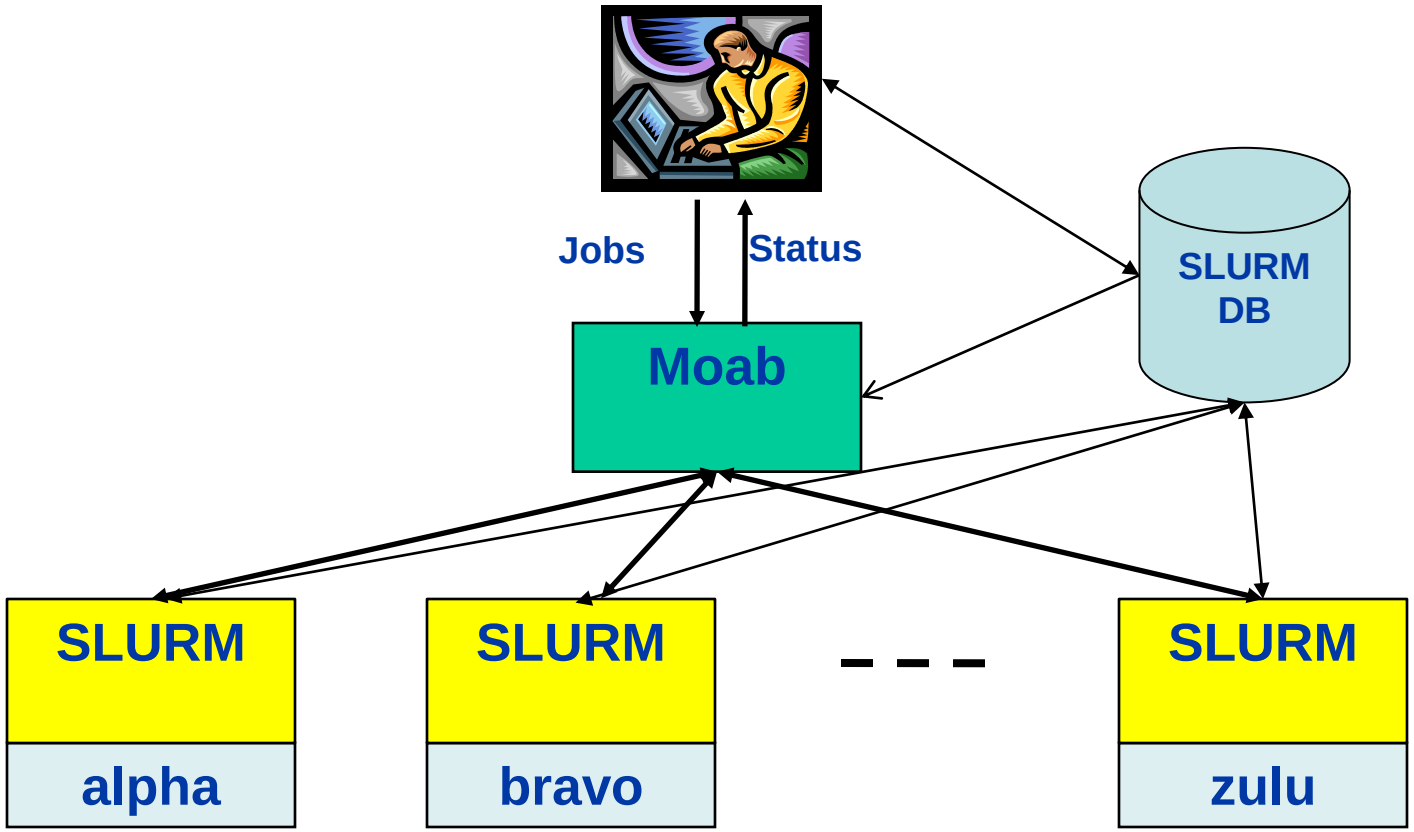
Status

SLURM

cluster

- Manages the computing resources of a single cluster
- Launches job across node(s)
- Provides job status and reports resource usage

Grid Scheduling



Grid Advantages

- Users can submit and status jobs from any cluster within the grid
- Users can submit jobs for multiple clusters and the scheduler launches the job on the soonest available cluster given the job's priority
- Users can submit jobs for one cluster that depend on a job from another cluster
  - e.g., a dedicated cluster with high I/O bandwidth is used to transfer output to storage once a job completes
- (A single user interface to a variety of different resource managers)

Command Comparison

Moab	(LCRM)	SLURM
msub	psub	sbatch
showq	pstat	squeue
checkjob	pstat -f	scontrol show job
(N/A)	(N/A)	sacct
mjobctl –m	palter	scontrol update job
mjobctl -h	phold	scontrol hold
mjobctl -u	prel	scontrol release
canceljob	prm	scancel
mjstat	spjstat	sjstat
mshare	pshare	sshare
mdiag –p	(N/A)	sprio