The PPA Bullet Cluster
New batch compute environment

Yemi Adesanya 5/21/2013

Overview of Batch HPC at SLAC

LSF 6.1 environment

General queues

- Available to all users
- ‘Embarassingly parallel’ jobs can run anywhere
- Group fairshares control job priorities
- RHEL5 production, RHEL6 in the future

Specialized queues

- Research programs acquire their own clusters
- Dedicated low-latency Infiniband for OpenMPI
- GPU servers with NVIDIA CUDA
- Mostly RHEL6 64bit

Planning for Lifecycle Replacement

Many batch nodes at/near end of lifecycle

- Maintenance costs go up as servers age
- Run-to-failure policy for oldest servers

Time for a new compute hardware standard

- Introduce Sandy Bridge CPU architecture
- Increase core-per-rack densities
- Modernize server infrastructure

Consolidate and share resources whenever possible

- A single PPA cluster for general and MPI?

Bullet Cluster Hardware Overview

2960-core cluster funded by PPA

185 Dell M620 Sandy Bridge blade nodes

10Gb ethernet and 40Gb QDR Infiniband networking

Up to 1024 cores per WC rack

Blade enclosures ideal for dense compute nodes:

- Higher core densities
- Integrated ‘top-of-rack’ network & management infrastructure
- Faster (cheaper) installation, deployment
- Modular design for potential mid-lifecycle upgrades

Bullet Software Environment

Available to all users for single-slot and MPI HPC

Start with LSF 6.1 and migrate to 9.x later

Bullet hosts are running RHEL6!

- Have users tried running jobs in the test queue?
- Jobs dispatched to RHEL5 or RHEL6 cores by default

All RHEL5 hosts will be eventually upgraded to RHEL6

How can single-slot and multi-slot MPI jobs co-exist?

Bullet cluster with LSF 6.1

Prevent large MPI jobs from being shut out

1024-slot test jobs ran with success

Revisit the recipe used for existing KIPAC MPI queue:

- Separate MPI queue for Bullet farm
- Give higher priority to multi-slot jobs
- Group single-slot jobs on fewest number of hosts
- MPI jobs can reserve cores in advance

Current limitations with 6.1

No sandbox environment (VMs, cgroups) for jobs

The 6.1 rules only work if jobs “play nice”

How do we apply rigid machine resource limits:

- Memory usage
- Available CPU cores
- Network bandwidth

How can we migrate running jobs between nodes?

Limit the number of running jobs per-user?