CMS use of allocation based HPC resources

Dirk Hufnagel (FNAL) for CMS Offline&Computing

CHEP 2016 San Francisco

13th October 2016

Overview

- Motivation
- HPC resources (US only)
- Strategy
- Progress

Motivation

HPC resources traditionally not used much by HEP.

Increased computing demands (especially looking ahead to HL-LHC) force us to look for more resources and the biggest possible gains are in areas we haven't already used much.

HPC potentially can provide a lot of resources.

HPC Resources (US only)

Name	Institution	Architecture	Start Date
Stampede	TACC	100k core Intel Sandy Bridge	2013
Stampede 2	TACC	Xeon+Phi	approved
Comet	SDSC	47k core Intel Haswell	2015
Edison	NERSC	133k core Intel Ivy Bridge	2013
Cori Phase 1	NERSC	52k core Intel Haswell	2015
Cori Phase 2	NERSC	632k core Intel Knights Landing (4x HT)	2016
Mira	ANL	786k core IBM PowerPC	2012
Theta	ANL	~150k core Intel Knights Landing (4x HT)	2017
Aurora	ANL	~3M core Intel Phi 3rd generation	2019
Titan	Oak Ridge	299k core AMD Opteron + GPUs	2012
Summit	Oak Ridge	~3400 nodes IBM Power 9 + GPUs	2017

NSF funded , DOE funded

Strategy

CMS already spends more of it's cpu budget on event reconstruction than on event generation and simulation and the ratio will become even more skewed with increased pileup. Accumulated core hours since ~February 2016:

~155M GenSim

~175M Data Reco + MC DigiReco

~190M User Analysis

Want to be able to run our full range of workflows, including Data Reco and MC DigiReco, on HPC resources.

- large data input/output, more stress on storage and network and in general more difficult to run efficiently
- non-x86 resources less interesting

Strategy

First order goal was to make HPC resources look just like any other CMS/OSG site (integrate it into glideInWMS system).

Limited amount of effort in CMS CompOps, the more these new resources can work like grid resources, the more different resources we will be able to easily support.

Don't have much local storage, so plan to not use any of it except for jobs temporary scratch space.

Was an open question if we could fully succeed in this, but we wanted to at least try before looking at more complex (and harder to implement) solutions.

Progress – SDSC Comet

We have infrastructure close by at UCSD (CMS site) and a good relationship with the SDSC admins.

Predecessor Gordon cluster was actually the first HPC site used for CMS Computing, we ran workflows consuming a few million cpu hours there.

Runs via Virtual Cluster interface in a mode being integrated into CMS CompOps.

Progress – TACC Stampede

Access via CE.

Still commissioning (had jobs running last week, still issues with stageout).

Progress – NERSC Edison (and Cori)

Bulk of our efforts have been spend here.

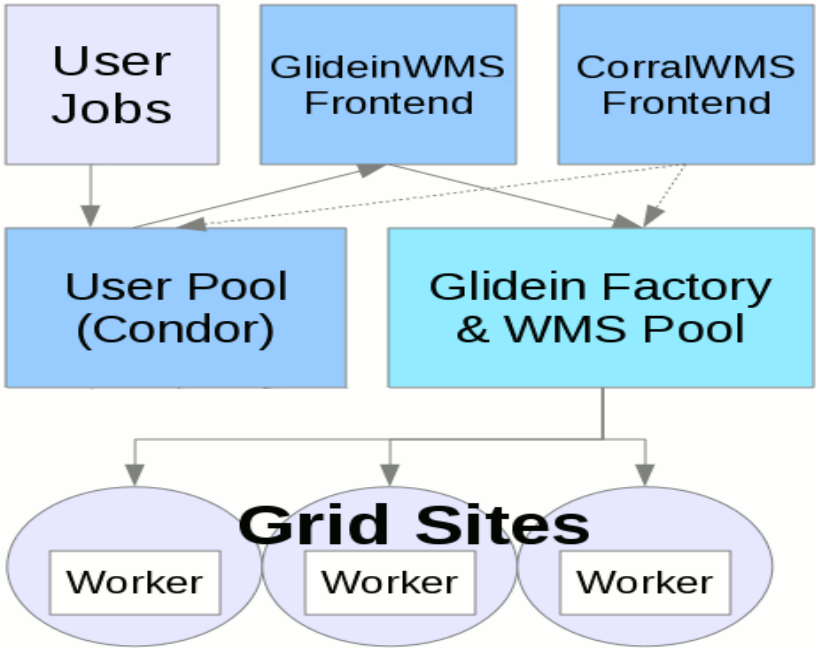
Class of HPC resource that is challenging, but not “impossible” (Cray SuperComputer that is very different from standard batch cluster, but: x86, nodes have outbound network, in principle all we need).

Cori Phase 2 one of the first HPC resources with Intel Knights Landing.

Two challenges:

- Submission (how to get GlideInWMS pilots onto worker nodes)
- Runtime (how to make jobs run)

GlideInWMS



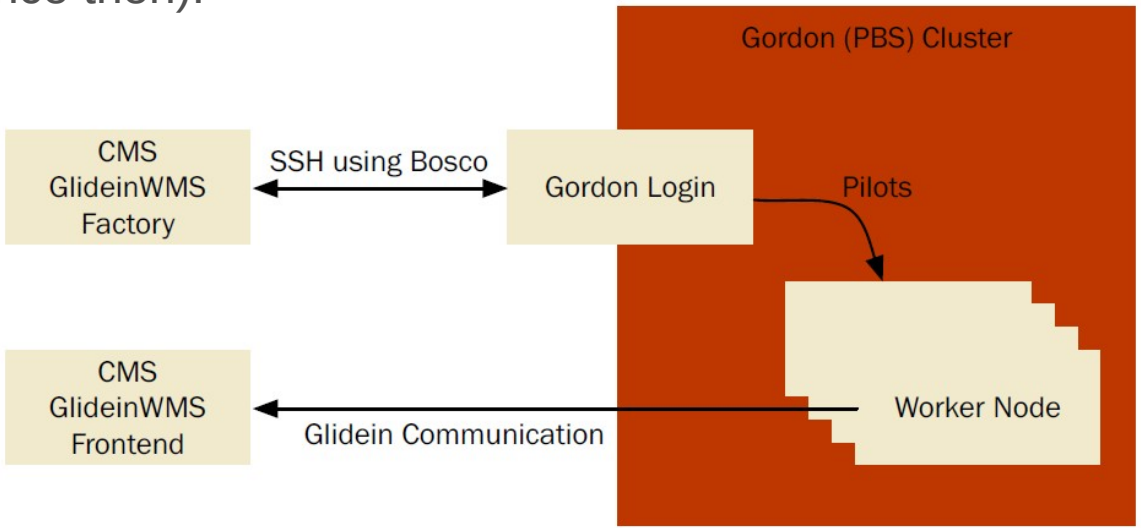
Follow link to watch animation

Progress – NERSC Edison (and Cori) Submission

Bosco style (remote ssh) pilot submission with local wrapper code to interface with site batch system.

Complication: NERSC migrated Cori/Edison to SLURM in January, HTCondor did not support native SLURM (fixed since then).

Example diagram for the SDSC Gordon cluster where we first used this



Fermilab

Progress – NERSC Edison (and Cori) Runtime

Worker nodes run Compute Node Linux, a stripped down Linux version for Cray worker nodes. No chance to run our software directly on that.

- ➔ support shifter (NERSC developed container system)

Cannot mount cvmfs on worker nodes (software in container).

NERSC worker nodes have outgoing network.

NERSC provides local squid proxies (for conditions data).

Progress – NERSC Edison (and Cori) Status

From the outside NERSC looks like a normal CMS site. On the inside:

- glideInWMS factory submits pilot via remote ssh
- BOSCO wrapper code submits pilot to SLURM
- BOSCO wrapper code configures shifter container
- pilots starts up in an SL6 shifter container
- pilot runs jobs
- jobs read data remotely via AAA xrootd federation
- jobs stages out remotely to FNAL

Tested MC generation/simulation but also Data/MC reconstruction. NERSC is used for the ongoing ReReco campaign and will be used for the MC DigiReco campaign.

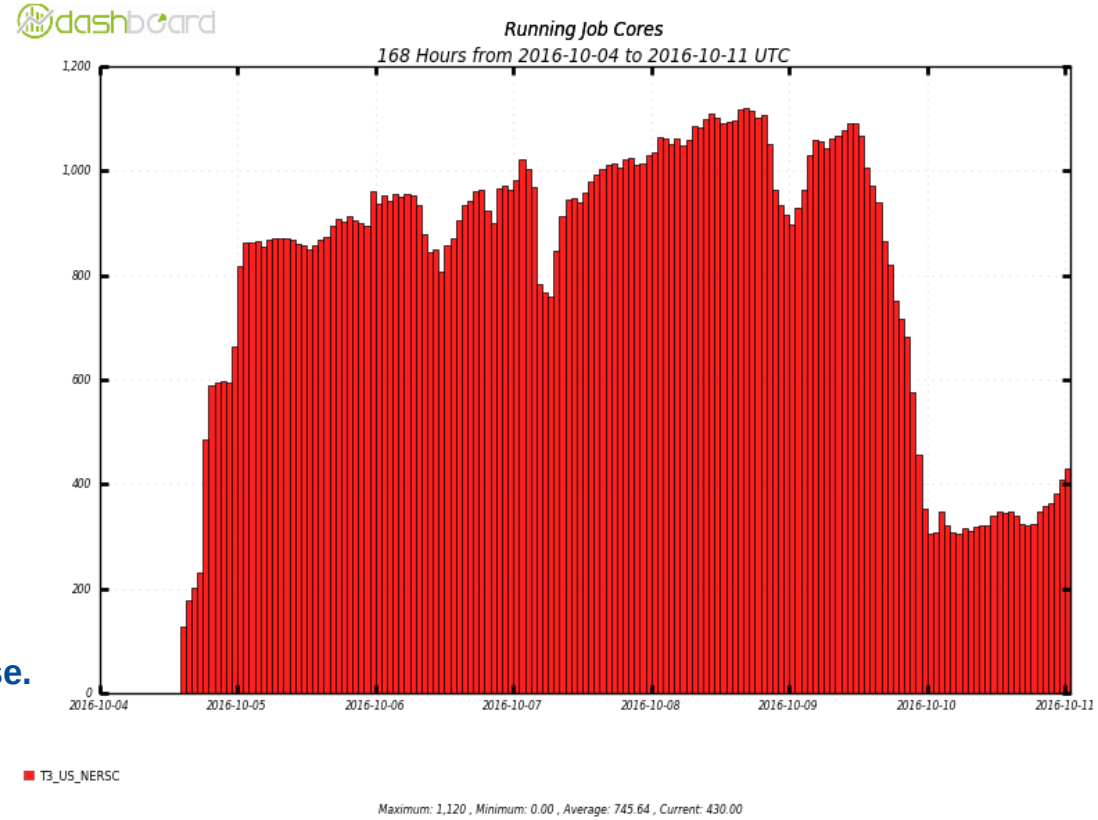
Progress – NERSC Edison (and Cori) Status

We are using two allocations:

1.5M cpu hours commissioning
5M cpu hours production

Haven't used up too much of it yet, only finished commissioning recently and still in the process of scaling up production use.

Limited by current Cori downtime.



Progress – NERSC Edison (and Cori) Problems

Still working out final solution to automate building shifter containers with the full CMS cvmfs repo (currently using Docker containers with only a single CMSSW version).

Bandwidth limits reading input data via AAA starve cpus on the nodes (current efficiency order 50%). Working with NERSC to fix this.

Edison is insanely busy at the moment due to Cori downtime, limits scale.

Number of TCP connections to the outside limited for cluster. Has caused problems with GlideInWMS (which needs quite a few TCP connections). Work ongoing both on our end to reduce number of TCP connections and on the NERSC side to increase the limits.

Conclusion

SDSC usable, being deployed in production. TACC almost usable.

Have managed to use NERSC as just another CMS site, it is used in production, but still working on efficiency and reliability improvements.

Integration into overall CMS Computing Infrastructure not perfect yet, working on smoothing out rough edges (mostly related to storage-less site concept).

Integration of US HPC resources via HEPCloud as an extension of Tier1 in the future, see related talks.