Enabling opportunistic resources for CMS Computing Operations

Dirk Hufnagel (FNAL)

for CMS Computing

CHEP, 14.04.2015

Overview

- Why now ?
- General strategy for opportunistic resources
- What do we need to run CMS jobs ?
- Different types of resources we want to use
- Gordon SuperComputer at SDSC

Why now ?

- LHC Run 2 will pose interesting Computing challenges

  ~ Factor 6 more computing demands
  ~ Factor 2 more resources

- Have used LS1 to improve code and algorithms, but Computing in Run 2 is still expected to be a resource constrained environment and very different from Run 1.

- Are there resources out there we can use to alleviate some of the constraints ?

General strategy for opportunistic resources

- CMS resources are organized as a single HTCondor Global Pool created/accessed via GlideInWMS.

  (caveat: Tier0 resources use a separate pool, not for technical but support reasons)

- Any opportunistic resource we want to use should be integrated into the Global Pool via GlideInWMS in a (longterm) sustainable fashion.

  (caveat: ignoring efforts to directly register resources into the Global Pool without going through GlideInWMS, ie. HTCondor startd)

What do we need to run CMS jobs ?

- Hardware and software architecture that can run CMS software, for the moment that is mostly AMD64 and SL6 (RHEL6) compatible
- GlideInWMS factory can submit pilots to resource.
- Pilot can call back to get work (outgoing network).
- Site specific settings (proxy servers, data access, stageout rules, etc)
  (provided local contextualization/cvmfs or local)
- Access to CMS software via cvmfs (local/remote proxy)
- Access to CMS conditions (local/remote proxy)

What do we need to run CMS jobs ?

- Data access:

  CMS xrootd data federation (AAA)

- Stageout rules:

  to “close” / “large” CMS site
  merges run at that site

- Local/Remote Proxy

  efficiency question, not going into it here

Types of resources : non-CMS grid

- No problems submitting pilots

- Is cvmfs available ?
- How do we get site specific settings to the job ?

=> parrot wrapper script in pilot fixes both

makes cvmfs available
provides site specific settings

had this working last year, testing involved FermiGrid plus a handful of OSG sites

was never integrated into production, will get back to this

Types of resources : Clouds

- Assuming EC2 interface no problems submitting pilots, factory can instantiate VM with built-in pilot
- Image includes cvmfs and contextualization to point at site specific settings in cvmfs
- CMS Tier0 resources at CERN are all cloud: GlideInWMS started VM in Openstack CERN AI

  6k cores scale tested
  9k cores deployed now
  ~15k cores planned for Run 2

Types of resources : Batch Clusters

- Submit the pilot ? No CE to connect to…

  Current biggest use case is an allocation we have at the Gordon super computer at SDSC.

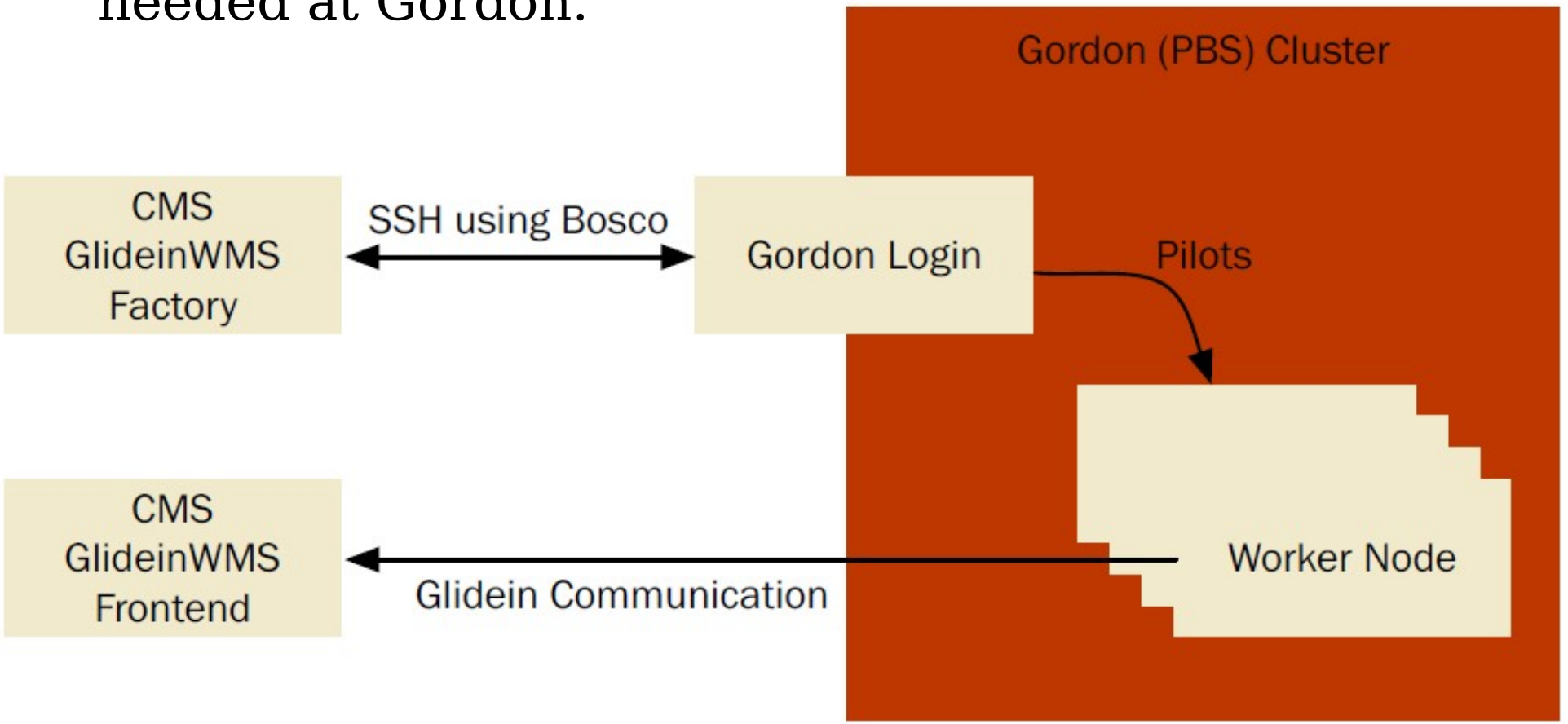
- Is cvmfs available ?
- How do we get site specific settings to the job ?

  In general the parrot wrapper could be the same answer as for the non-CMS grid sites, but SDSC mounted cvmfs for us, so no need. Could contextualize the environment enough to pull site specific settings from cvmfs.

  Exploring parrot for this use case at NERSC.

Using BOSCO submission mode inside glideInWMS

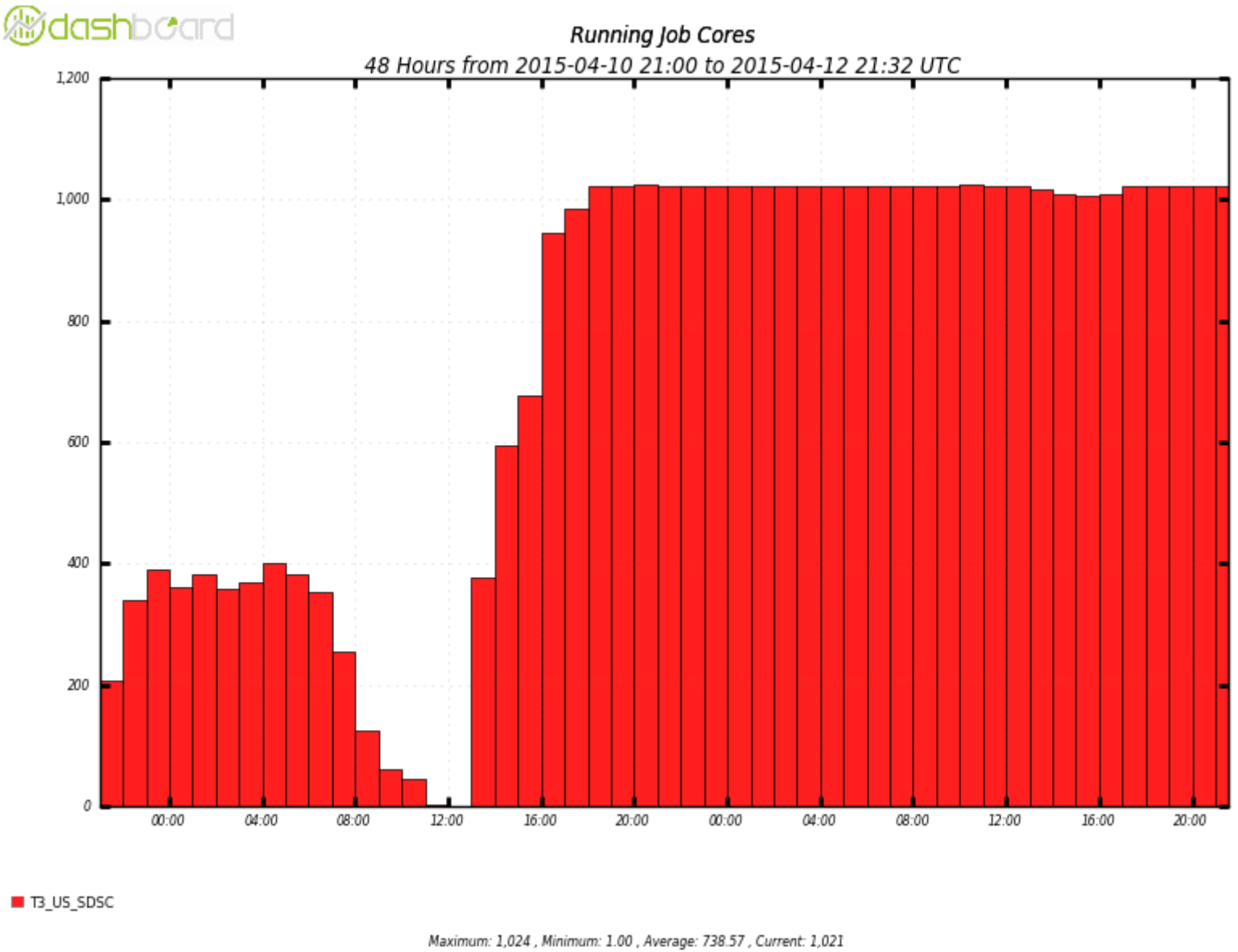
- Use an ssh tunnel from the factory to submit a pilot to a batch slot at Gordon. All supported out of the box, no custom factory or anything like that. Some BOSCO code needed at Gordon.



SDSC status

- We can run workflows at SDSC, using standard CMS production procedures. We aren't in the Global Pool yet, but otherwise this looks exactly like running workflows at any other CMS site.
- MC processing jobs ran at SDSC, staged out to FNAL, merge jobs then ran at FNAL.
- Used FNAL factory, but SDSC entry is still CMS specific (contains user name of account at Gordon). Eventually want to have this be VO-agnostic.

SDSC status


dashboard
Running Job Cores
48 Hours from 2015-04-10 21:00 to 2015-04-12 21:32 UTC
1,200
1,000
800
600
400
200
0
00:00
04:00
08:00
12:00
16:00
20:00
00:00
04:00
08:00
12:00
16:00
20:00
T3_US_SDSC
Maximum: 1,024 , Minimum: 1.00 , Average: 738.57 , Current: 1,021

Summary

- Clouds are usable out of the box if EC2 interface exists.
- Batch clusters are now usable more or less out of the box (assuming cvmfs is available), but still some manual setup work to be done. Will work on further integration and taking of the rough edges.
- Work ongoing on parrot integration. Not needed for all resources, but would allow access to more generic (and also smaller) resources.

Backup

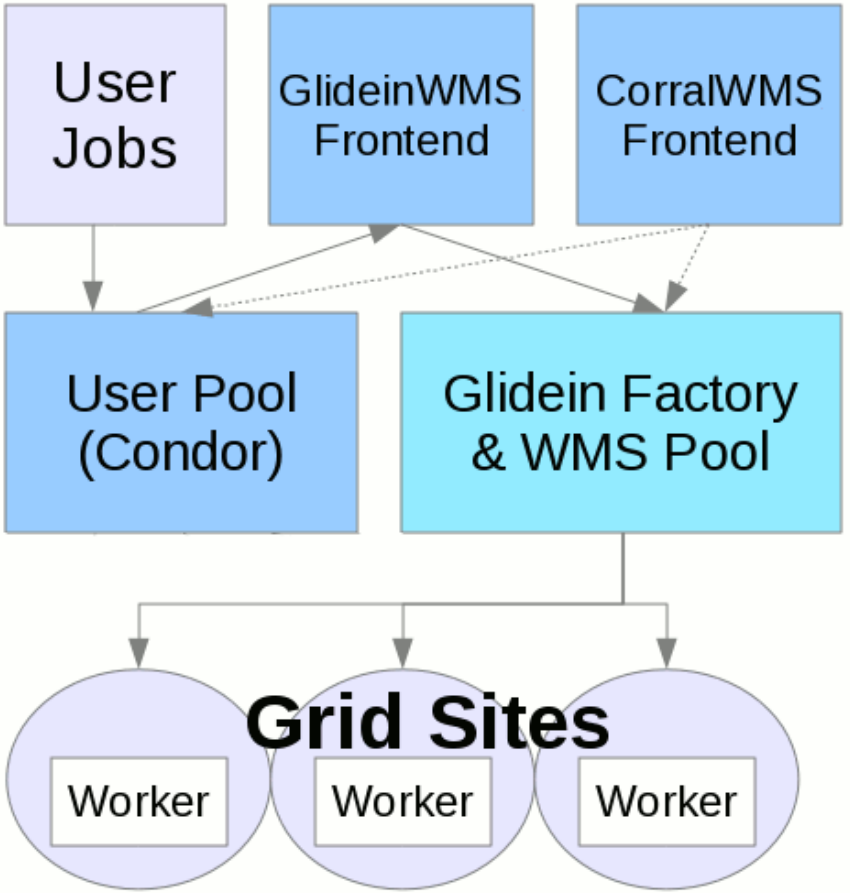
Introduction of terms

- GlideInWMS : pilot based workflow management system used by CMS, makes grid (and other) resources accessible via HTCondor (see next slide)
- Parrot : tool for attaching existing programs to remote I/O systems through the filesystem interface (as user)
- BOSCO : allows local submission to remote (batch) resources via ssh tunnel into the remote resource (see later slide for details and diagram)
- cvmfs : network file system based on HTTP, optimized to deliver experiment software in a fast, scalable, and reliable way

GlideInWMS

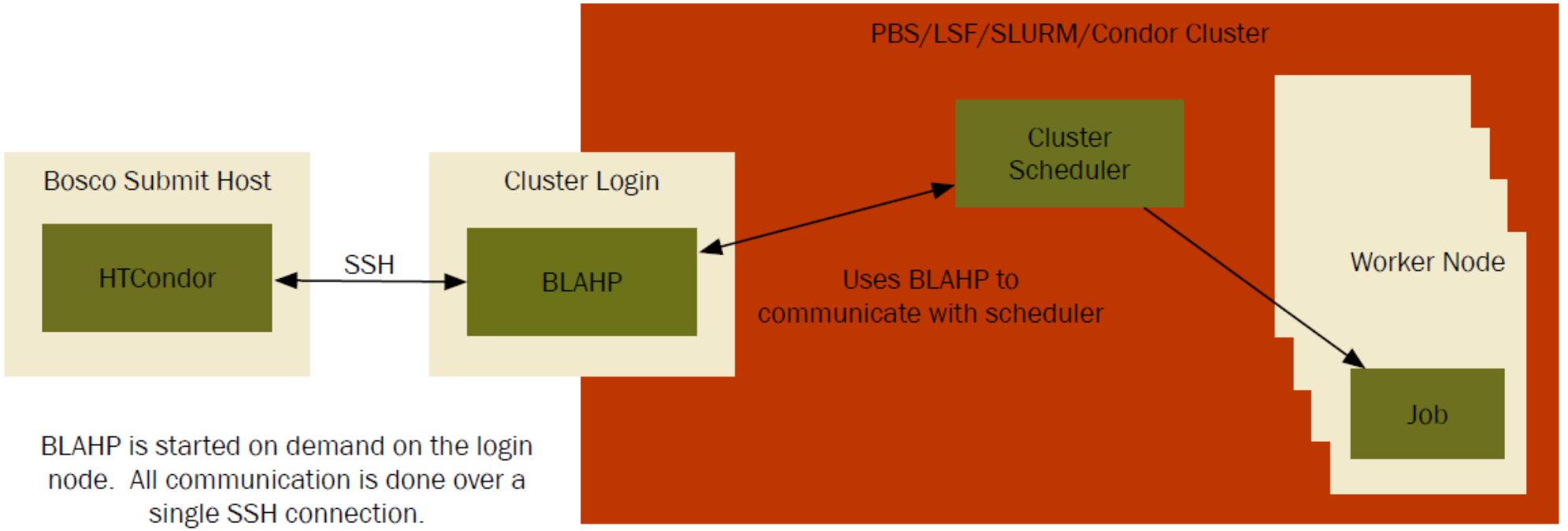
http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/doc.prd/index.html

Follow the link and check out animation



BOSCO

- BOSCO is a tool to allow submitting jobs to remote accounts from your local desktop (ie. you have an account at CERN, but want to submit to LSF from your desktop at your home institution). It uses HTCondor under the hood.



Outlook on “BOSCO mode”

- The question of how much “BOSCO mode” resources can we eventually expect has come up…
- If we just talk about super computers and allocation based resources, the answer is probably a handful or two at the outside.
- OTOH, if you look at the complexity of this compared to setting up a CE at the site, this could be attractive for many (smaller) sites…
- Also, eventually this should be multi-VO usable
  (it is now actually, but at high cost to factory ops)