# OSG Technology Report

Brian Bockelman
OSG AHM 2016

# Overview Talks Are Difficult

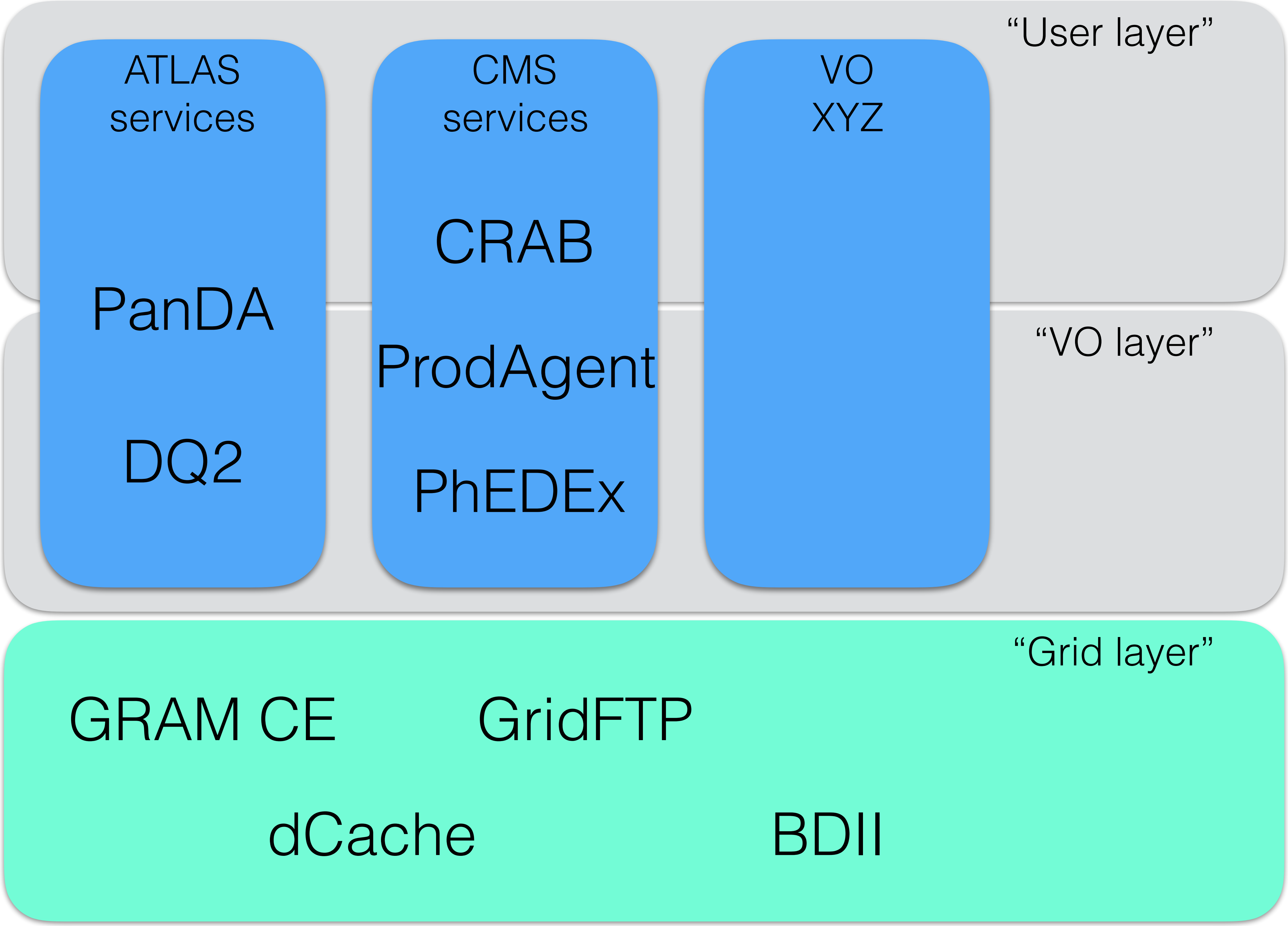
- This talk tries to give the big picture view of the activities of a large team.
- Which is to say I provide almost no details or anything interesting.
    - Where possible, I try to provide links to relevant further information.
- This being the technology area, I focus on a narrow portion of the OSG - the software and technology stacks!

# What do we do?

- The OSG technology team…
  - … produces the OSG Software stack.
  - … investigates and adapts new technologies.
  - … assists other teams in using our technologies.

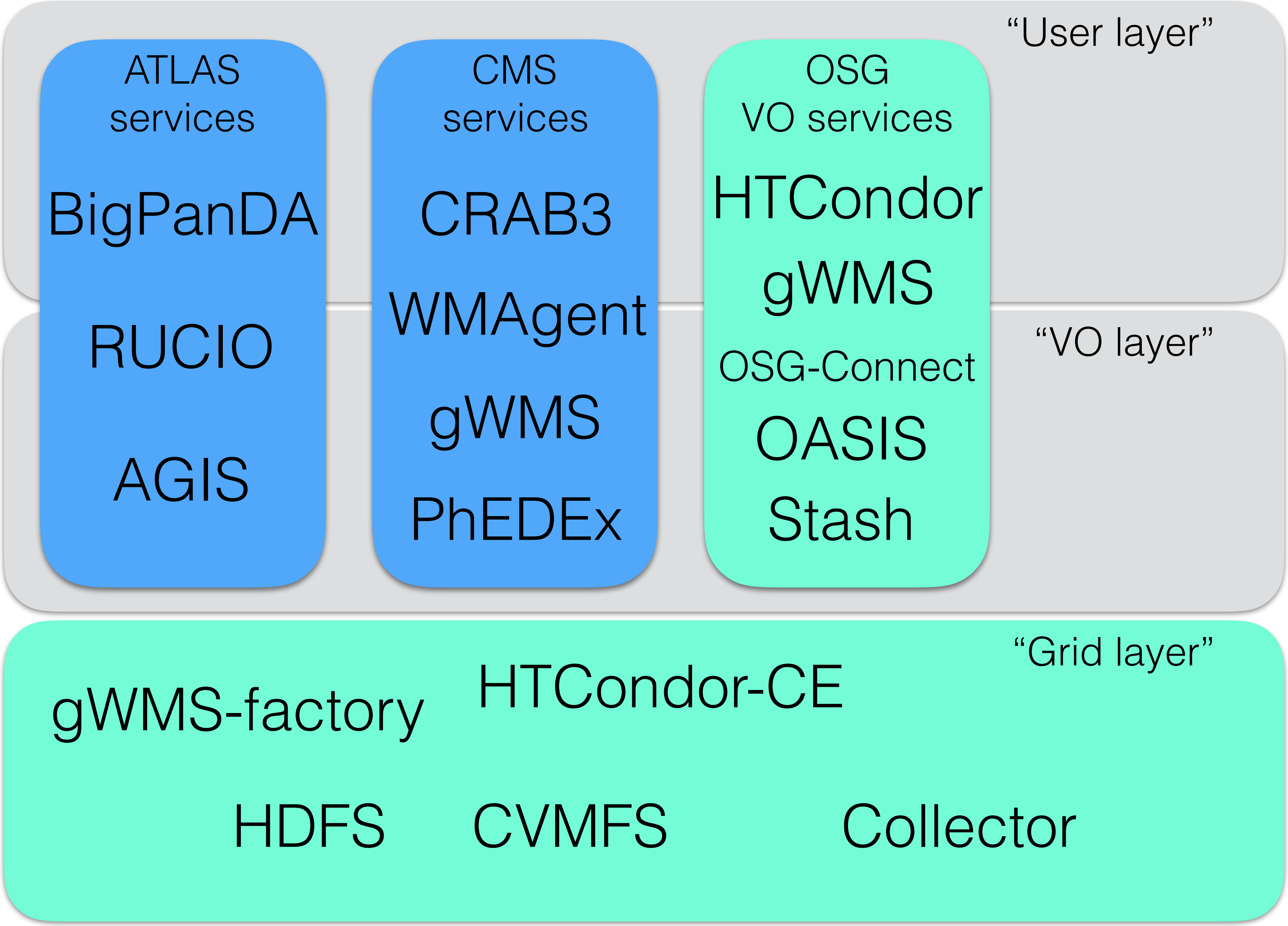
# The OSG Fabric of Services

(A ridiculous simplification; circa 2005)



# The OSG Fabric of Services

(A ridiculous simplification; circa 2016)



"User layer"

"VO layer"

"Grid layer"

ATLAS services

BigPanDA

RUCIO

AGIS

CMS services

CRAB3

WMAgent

gWMS

PhEDEx

OSG VO services

HTCondor

gWMS

OSG-Connect

OASIS

Stash

gWMS-factory

HTCondor-CE

HDFS

CVMFS

Collector

# The OSG Fabric of Services

- Unsurprisingly, many of the service names have changed.
- The conceptual difference is the OSG VO is run by the OSG, meaning we have to select a few higher-level services to run. This is a reference platform.
  - One of many platforms you can build.
- Broadly, the reference platform consists of:
  - **Job and Resource provisioning**: HTCondor and GlideinWMS.
  - **Software distribution**: OASIS (OSG-branded CVMFS service).
  - **Data distribution**: Stash (data federation).

# Software

# This Year in OSG Software

- The OSG Software team has been able to reliably ship updates to our production-quality software stack every month.

- We often keep two concurrent release series; currently OSG 3.2 (security updates only) and 3.3.

    - As a new series is the only way we drop support or (purposely) break backward compatibility, new series occur a bit faster than I had predicted.

    - Release lifetime is somewhere in the neighborhood of 2 years.

# 2015 Highlights

- **HTCondor-CE 2.0**:  Significant improvements in monitoring and batch system support.

- Improved **EL7 support**:  Now supported for all components except bestman2 and GUMS.

- **CVMFS 2.2.0**:  Significant new features to support data federations and authorization.

- **Globus Toolkit 6.0**:  Notable for how painless rollout was (compare to prior years).

- **HTCondor**: In 2015, greatly improved track record of keeping up-to-date with the upstream release.

# Looking ahead: OSG 3.4

- My personal goals for OSG 3.4 (2017):
  - **Drop GRAM** (very likely).
  - **Hadoop 3.0** (waiting for error correcting codes to make official release).
  - **Drop bestman2** (aspirational goal).
  - **Drop GIP** / osg-info-services (seems feasible).

# A word on GRAM

- While most dates are undecided, the sequence is:
  - **April**: “`yum install osg-ce-condor`” will no longer pull in GRAM.  GRAM can be installed separately and still configured.
  - Early 2017: Remove OSG GRAM CEs from OSG-run pilot factories.
  - Early 2017: Ship OSG 3.4.0 without GRAM.
  - Early 2017: Remove GRAM support from tools such as osg-configure or Gratia.

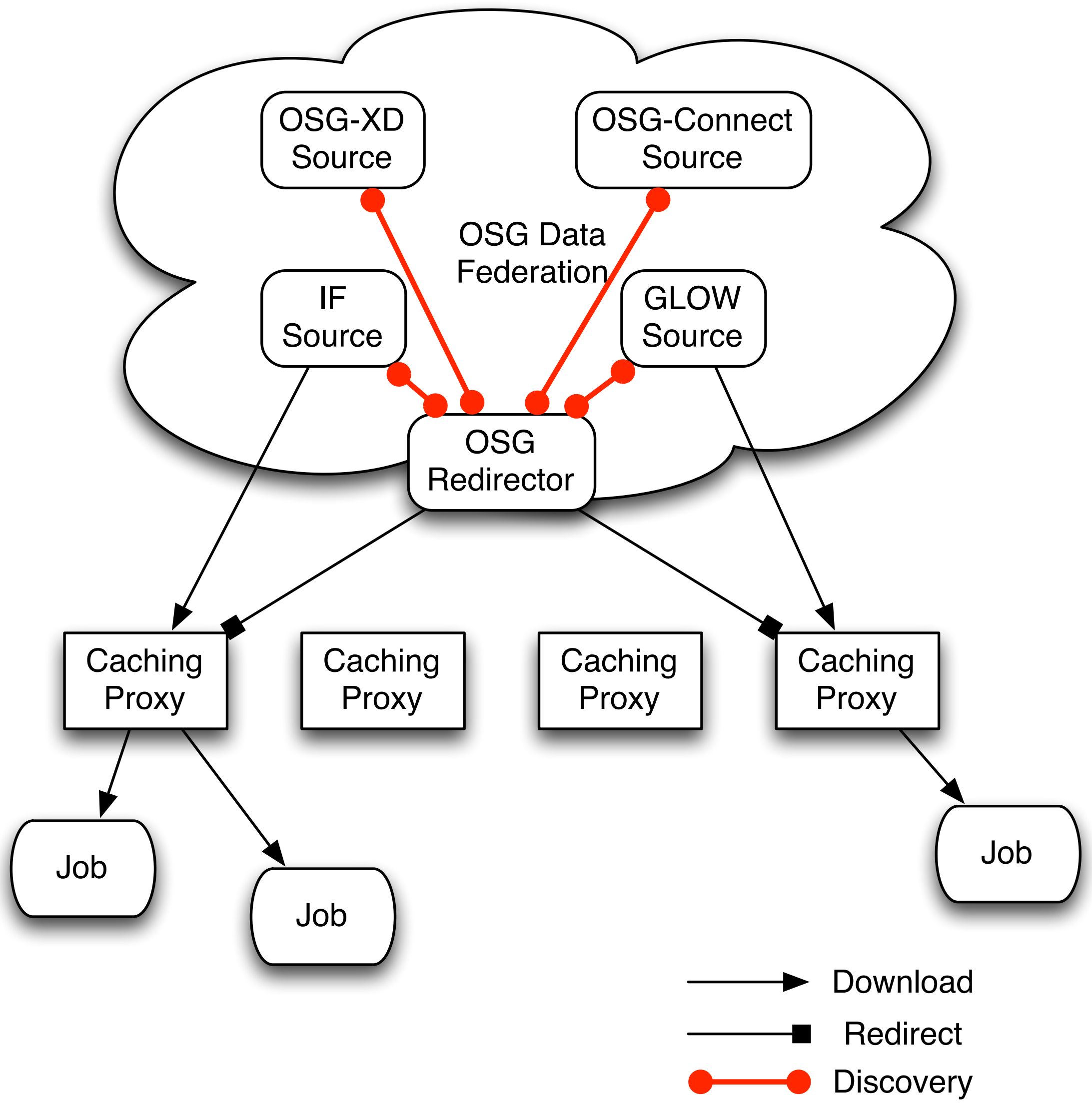
# Technology

# Resource Provisioning

- OSG offers a robust resource provisioning service utilizing GlideinWMS (primarily) and a few other tools.  In addition to the traditional grid sites, this software can provision:
  - VMs via Amazon Web Services https://indico.fnal.gov/contributionDisplay.py?contribId=15&confId=10571 (done by GlideinWMS software *but* not using the OSG service).
  - Glideins accessing their XD allocations https://indico.fnal.gov/contributionDisplay.py?contribId=33&confId=10571 (Using resources at TACC Stampede - more later).

# Stash

- Last year, we rolled out a preliminary service offering for data access, *Stash*.
  - A VO adds a source or origin server to a OSG data federation.
- Jobs can access these sources through an OSG-run proxy.
  - OSG validates each proxy is sufficiently powerful for larger working set sizes



# An assist from CVMFS

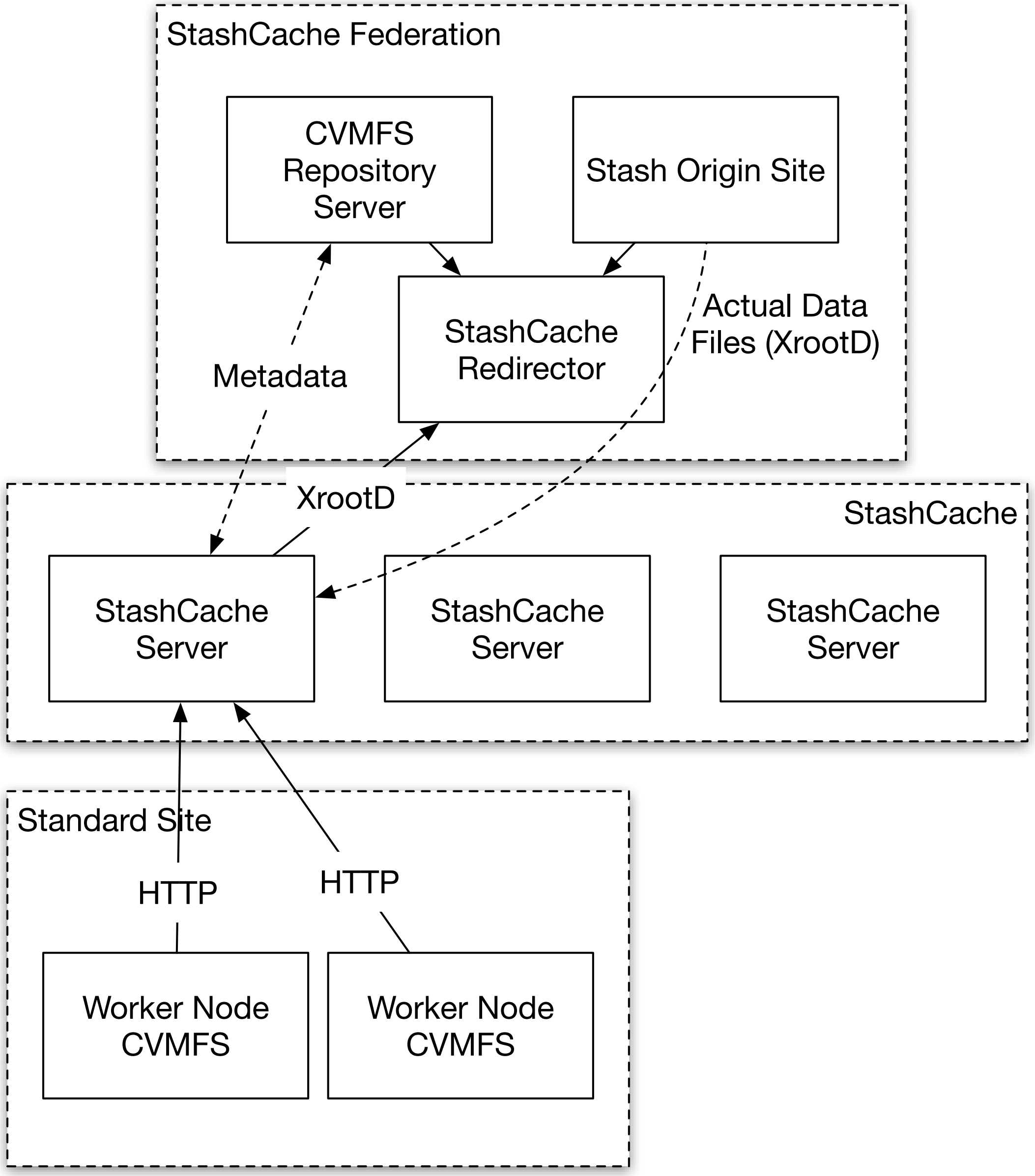
- Stash functions as a data access method but feedback was it was difficult to use.

- Hence, we worked with CVMFS team to combine the metadata scalability of CVMFS with the data distribution of data federations.

- The new stash.osgstorage.org repo is simply a mirror of any file placed into the user’s public directory on OSG-Connect.

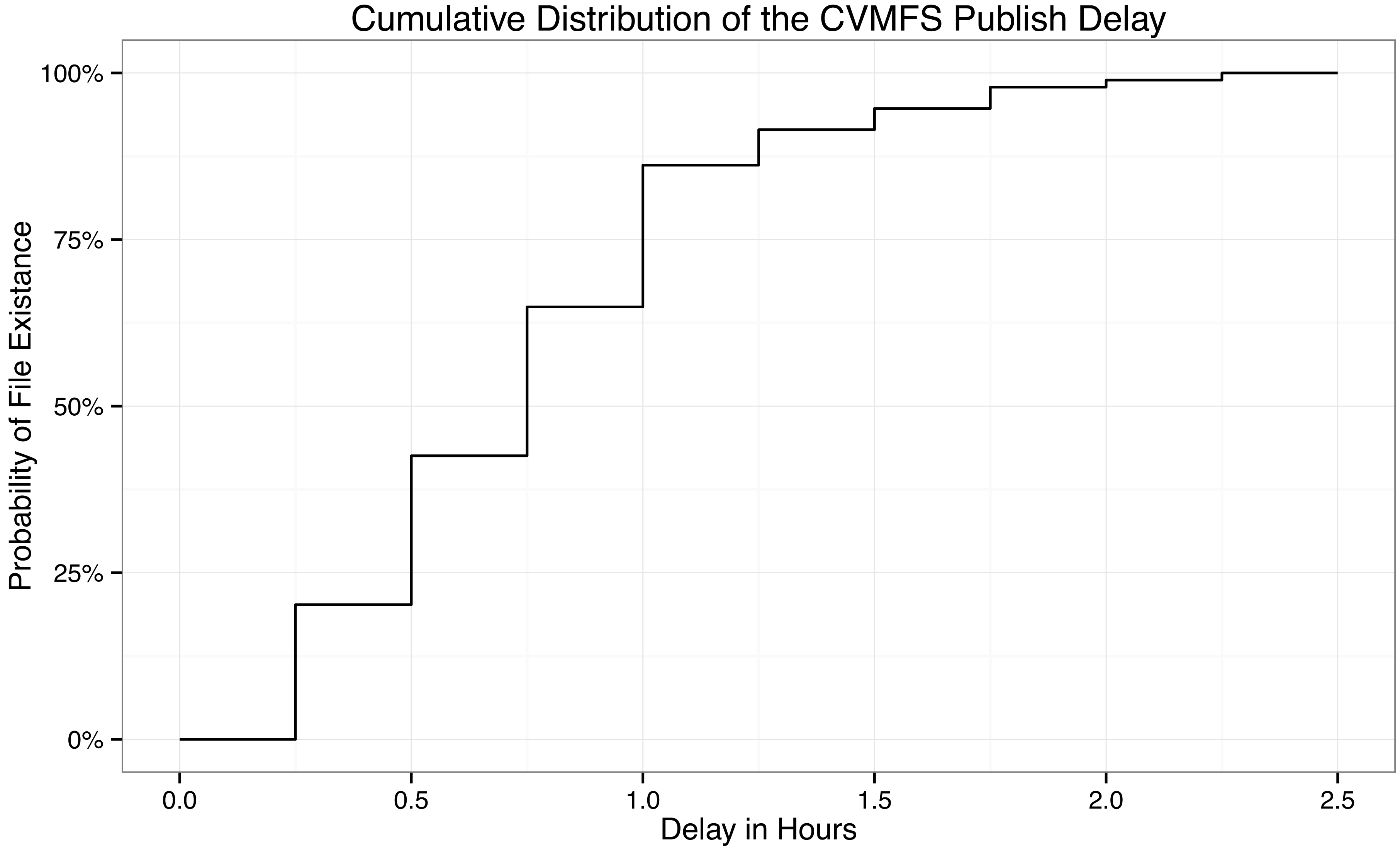
# Combining CVMFS and Stash

- “Regular” XrootD StashCache Federation - provides a single endpoint for locating files.
    - Distribute a series of caches around the OSG: “StashCache”.
- CVMFS contacts the caching servers over HTTP
- If a file is missing, caching servers contact the federation for the data
- Worker nodes pull data from the caching servers to local disk.
    - Finally, the FUSE mount delivers this to the job.



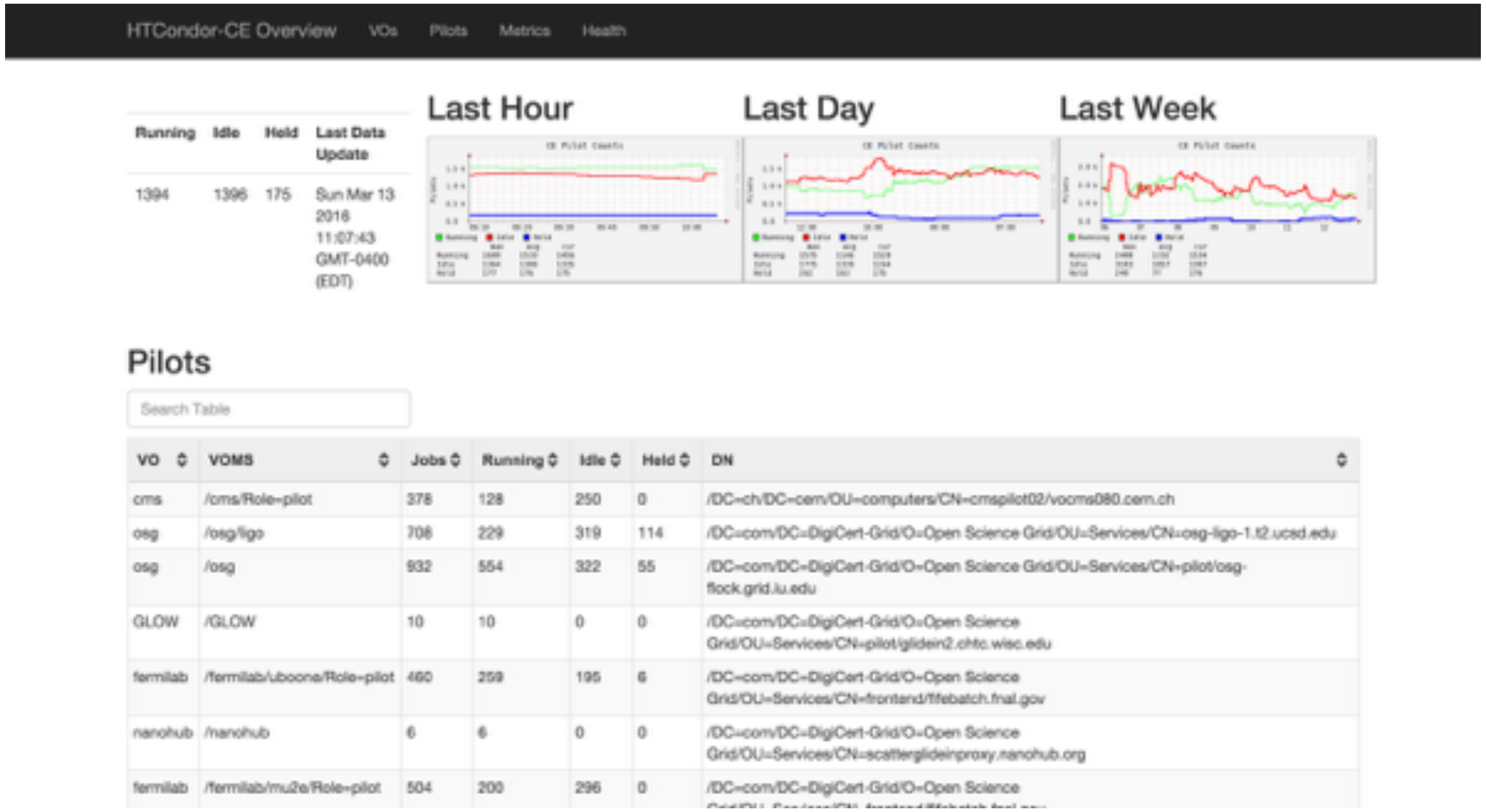
# Synchronization Times

- We’ve started to measure the synchronization delays between CVMFS and worker nodes.
- Very new effort: most relevant result is the methodology works!  Currently, around 85% of updates take an hour or less.



# HTCondor-CE

- Still finishing the tails of batch system support.
- Starting to focus more on helping to provide visibility



| Running | Idle | Held | Last Data Update |
|---|---|---|---|
| 1394 | 1396 | 175 | Sun Mar 13 2016 11:07:43 GMT-0400 (EDT) |

## Pilots

Search Table

| VO | VOMS | Jobs | Running | Idle | Held | DN |
|---|---|---|---|---|---|---|
| cms | /cms/Role=pilot | 378 | 128 | 250 | 0 | /DC=ch/DC=cern/OU=computers/CN=cmspilot02/vocms080.cern.ch |
| osg | /osg/ligo | 708 | 229 | 319 | 114 | /DC=com/DC=DigiCert-Grid/O=Open Science Grid/OU=Services/CN=osg-ligo-1.t2.ucsd.edu |
| osg | /osg | 932 | 554 | 322 | 55 | /DC=com/DC=DigiCert-Grid/O=Open Science Grid/OU=Services/CN=pilot/osg-flock.grid.iu.edu |
| GLOW | /GLOW | 10 | 10 | 0 | 0 | /DC=com/DC=DigiCert-Grid/O=Open Science Grid/OU=Services/CN=pilot/glidein2.chtc.wisc.edu |
| fermilab | /fermilab/uboone/Role=pilot | 460 | 259 | 195 | 6 | /DC=com/DC=DigiCert-Grid/O=Open Science Grid/OU=Services/CN=frontend/fifebatch.fnal.gov |
| nanohub | /nanohub | 6 | 6 | 0 | 0 | /DC=com/DC=DigiCert-Grid/O=Open Science Grid/OU=Services/CN=scatterglideinproxy.nanohub.org |
| fermilab | /fermilab/mu2e/Role=pilot | 504 | 200 | 296 | 0 | /DC=com/DC=DigiCert-Grid/O=Open Science |

# HTCondor Pilot View

- New `condor_ce_status` command output shows the payload information:

```
[root@red ~]# condor_ce_status
Worker Node         State     Payload ID      User       Scheduler                         Job Runtime BatchID    BatchUser  Jobs   Pilot Age
red-c0801.unl.edu   Unclaimed                                                              0+15:29:58  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   436690.16       cmsdataops cmsgwms-submit2.fnal.gov          0+04:00:15  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   437109.65       cmsdataops cmsgwms-submit2.fnal.gov          0+04:24:41  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   437616.56       cmsdataops cmsgwms-submit2.fnal.gov          0+00:07:12  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   437132.39       cmsdataops cmsgwms-submit2.fnal.gov          0+04:09:59  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   437110.8        cmsdataops cmsgwms-submit2.fnal.gov          0+04:22:43  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   437214.89       cmsdataops cmsgwms-submit2.fnal.gov          0+03:46:19  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   437135.91       cmsdataops cmsgwms-submit2.fnal.gov          0+04:06:04  5803811.0  cmsprod      37   0+15:30:18
red-c0801.unl.edu   Claimed   437113.83       cmsdataops cmsgwms-submit2.fnal.gov          0+04:16:42  5803811.0  cmsprod      37   0+15:30:18
red-c0803.unl.edu   Unclaimed                                                              0+01:36:42  5804385.0  cmsprod      34   0+01:37:03
red-c0803.unl.edu   Claimed   437620.6        cmsdataops cmsgwms-submit2.fnal.gov          0+00:08:25  5804385.0  cmsprod      34   0+01:37:03
red-c0803.unl.edu   Claimed   61125.0         cmst1      vocms0311.cern.ch                 0+00:48:54  5804385.0  cmsprod      34   0+01:37:03
red-c0803.unl.edu   Claimed   437536.40       cmsdataops cmsgwms-submit2.fnal.gov          0+00:46:01  5804385.0  cmsprod      34   0+01:37:03
```

- Looking to extend this to do simple payload accounting on the CE: allows you to see *who* is using your CE.
  - Relies on VOs to self-report.

# HTCondor-CE: New Friends

- CERN’s next generation batch system is HTCondor; after an evaluation period, they based the corresponding CEs on HTCondor-CE.
    - We see this as the seed of a new collaboration: not just visibility within other “social circles” but also code contributions.
- We helped organize a “HTCondor Week” in Europe to grow the community.
    - We were **overwhelmed** by the variety of sites and use cases that we found.
- I’m excited to see how this will grow in 2016!

https://indico.cern.ch/event/467075/

# Joint Projects

# LIGO: An Ancient History

- LIGO was an OSG stakeholder in the early days.
- However, we scared them off for a few reasons, including:
  - **OSG was hard to use**: Payload jobs were sent to GRAM using Condor-G. Quite unreliable and a foreign interface to users.
  - **No solution for software / data**: We asked sites to provide NFS mounts ($OSG_APP, $OSG_DATA) but these were inconsistently deployed and had no management tools.
  - **User-unfriendly requirement of certificates**: The process of getting a DOEGrid certificate was grueling.
- Running opportunistically on OSG required more effort / expertise / blood / sweat / tears than LIGO had to spare. Cost/Benefit didn’t make sense!

# LIGO

- We started working with LIGO around HTCondor Week in 2015.

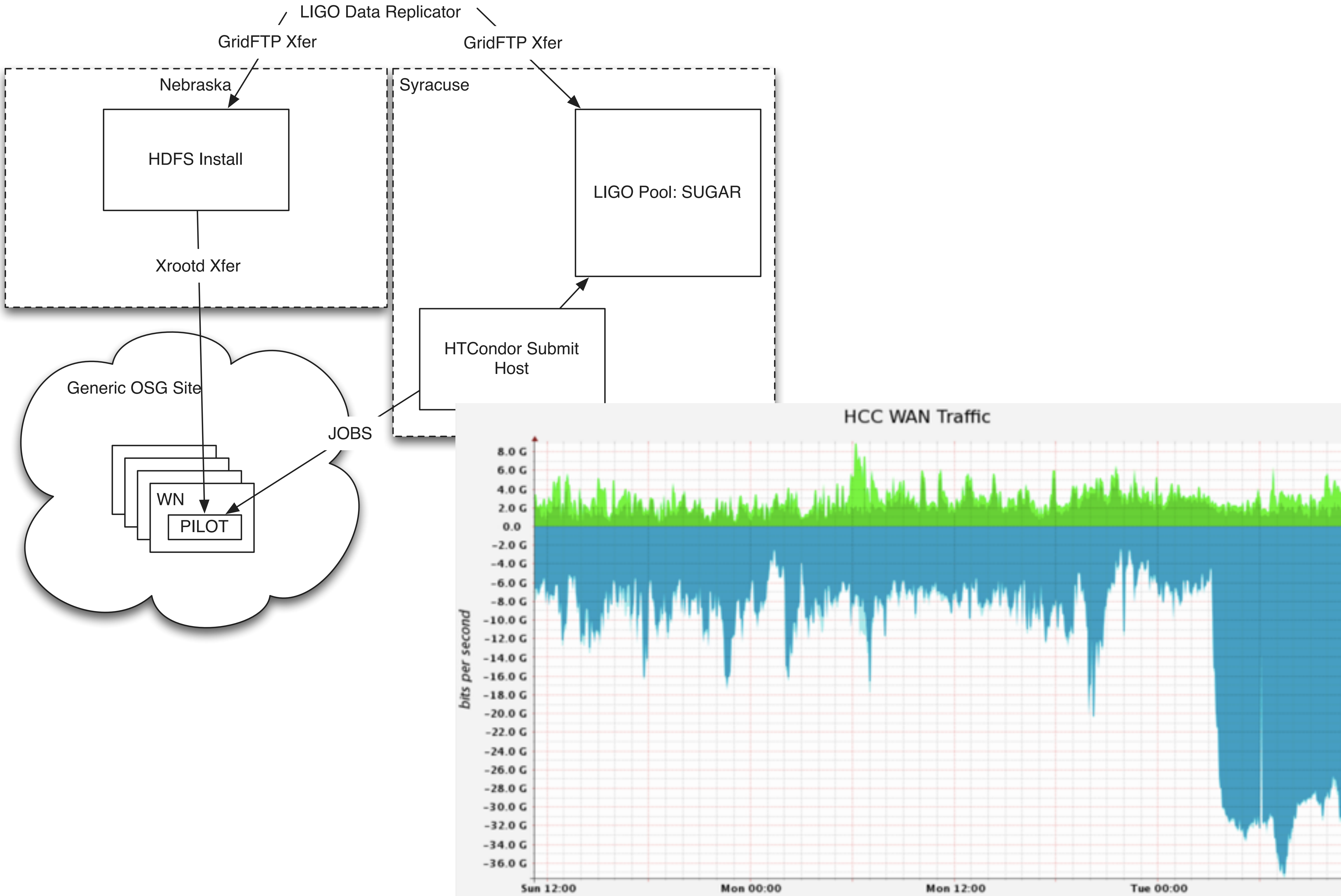
- The effort really started to accelerate around October.

    - For 2-3 weeks, we were seeing an order-magnitude increase of jobs per week.

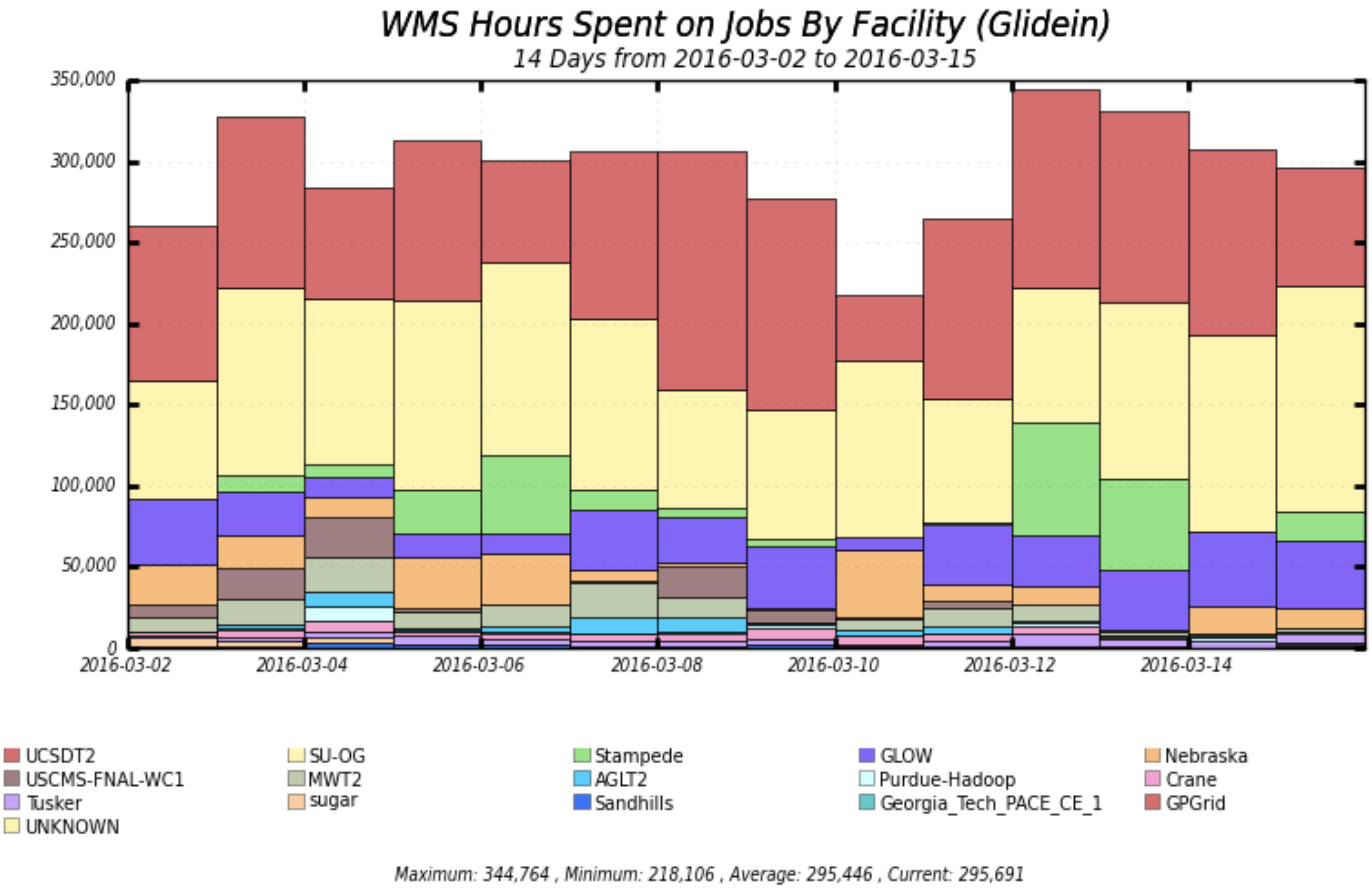
- By the end of the year, several million CPU hours.

- Wonderful chance to have the technology team see the “reference platform” in action.  Several lessons learned have fed back into the OSG Software stack.

# LIGO


LIGO Data Replicator
GridFTP Xfer
GridFTP Xfer
Nebraska
HDFS Install
Xrootd Xfer
Syracuse
LIGO Pool: SUGAR
HTCondor Submit Host
Generic OSG Site
JOBS
WN
PILOT
HCC WAN Traffic
8.0 G
6.0 G
4.0 G
2.0 G
0.0
-2.0 G
-4.0 G
-6.0 G
-8.0 G
-10.0 G
-12.0 G
-14.0 G
-16.0 G
-18.0 G
-20.0 G
-22.0 G
-24.0 G
-26.0 G
-28.0 G
-30.0 G
-32.0 G
-34.0 G
-36.0 G
bits per second
Sun 12:00
Mon 00:00
Mon 12:00
Tue 00:00

# Last two weeks - >4M hours



This includes partial use of a 2M SU allocation at Stampede (green).

# XD / Stampede

- LIGO brought an interesting challenge: can they utilize their XD allocation as part of the same infrastructure?
  - Initially submitted glideins by hand to SLURM.  These pulled down and ran LIGO jobs.
  - After initial successes - and verifying that the site infrastructure felt “sufficiently like home” - we switched to GRAM-based submissions, using the local infrastructure.
  - Today, a single Pegasus-based workflow can run seamlessly on both OSG opportunistic and XD allocations.
- **Lesson learned**: TACC&Stampede - technically and organizationally - is a resource we can interoperate with and can leverage more in the future.
  - Similar to SDSC a few years back, I hope we have planted the seeds of a successful collaboration.
  - A few technical changes could still make a big difference…
  - Regardless - this is something that is ready to repeat elsewhere!

# Parting Shots

- OSG Technology team has a broad range of activities - from fixing simple bugs to partnering with projects that are pushing our boundaries.
    - The OSG Technology team hopes to continue to ship a stable base - for the next years!
    - By streamlining our base software layer year-over-year, we are able to tackle a wider range of problems.
- The OSG VO provides a reference platform - sometimes a euphemism for “guinea pig” - that gives the Technology team insight to what users need.
    - This allowed us to spend quite some time studying difficult data issues throughout the year.