ATLAS Connect

A US ATLAS project following the OSG Connect pattern

Rob Gardner • University of Chicago

OSG Council Meeting, University of Oklahoma
22 October 2014

Objectives of ATLAS Connect

- Provide:
  - User login service providing a virtual Tier-3 cluster-like processing environment for batch (like OSG Connect)
  - Connect Tier-3 sites to additional resources
  - Connect additional (non-WLCG) resource targets to the Panda WMS (campus clusters, XSEDE clusters)
- Leverage simple, robust services
  - OSG/CI Connect (HTCondor, Globus, Bosco), xrootd, http, AutoPyFactory, Panda

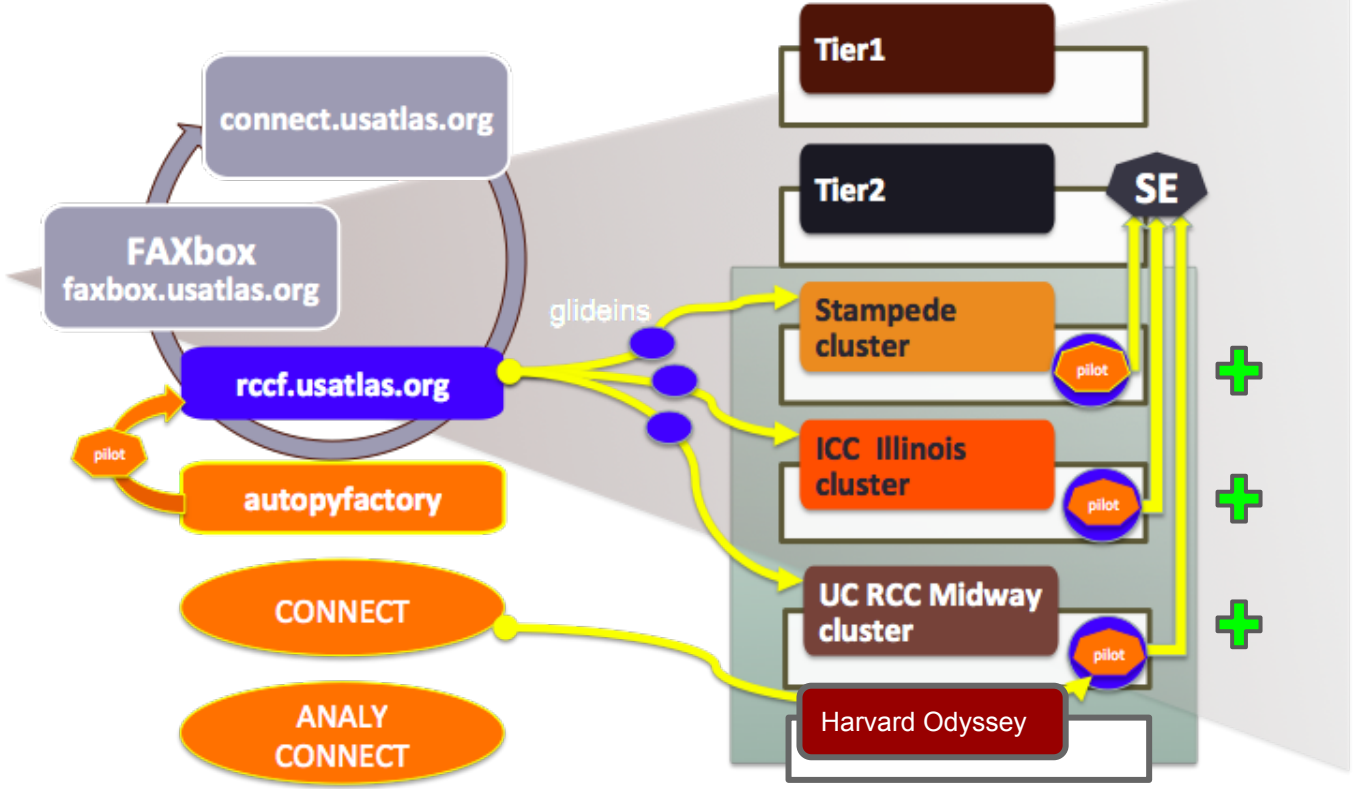
Distributed resource targets

- Goal is to make the connection lightweight
  - Local site provides only a user account, and optionally a squid
- We deal with the heterogeneity:
  - IP connectivity, software access, squid, data access
- Site retains control over priorities
- Operated by US ATLAS staff

User login service

- A simple sign-up service is available from connect.usatlas.org
- Authorization through institutional affiliation
  - Responsibility lies with the PI
- Emulate a Tier 3 login server
- See accompanying document “The ATLAS Connect Virtual Cluster Service”

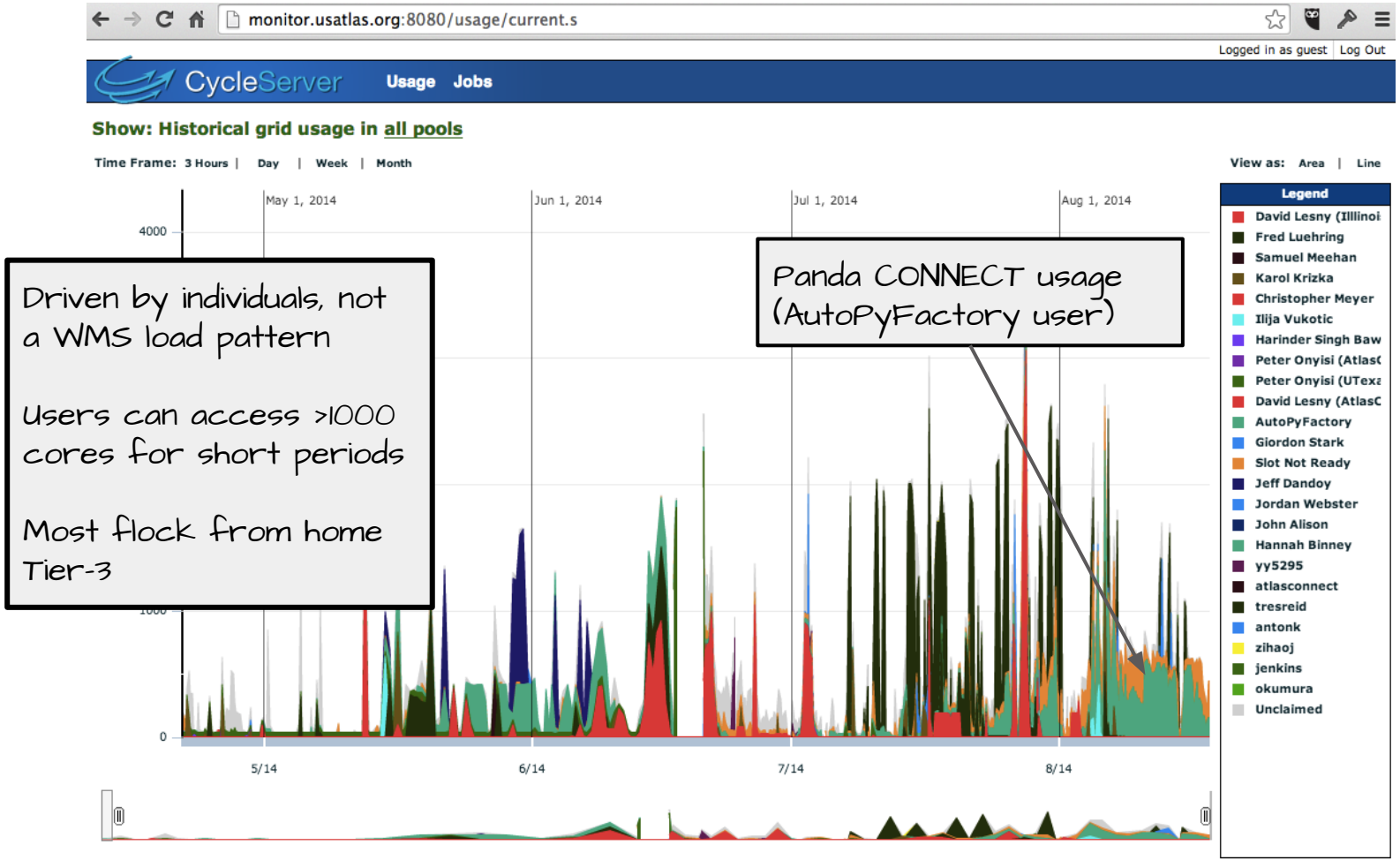
CONNECT panda



Pilots scheduled to glideins receive jobs from CERN

Input data via FAX, outputs written to a Tier-2 storage element "SE"

For many clusters, the issue is access to CVMFS served software


monitor.usatlas.org:8080/usage/current.s
Logged in as guest
Log Out
CycleServer
Usage
Jobs
Show: Historical grid usage in all pools
Time Frame: 3 Hours | Day | Week | Month
View as: Area | Line
May 1, 2014
Jun 1, 2014
Jul 1, 2014
Aug 1, 2014
4000
0
5/14
6/14
7/14
8/14
Legend
David Lesny (Illinoi
Fred Luehring
Samuel Meehan
Karol Krizka
Christopher Meyer
Ilija Vukotic
Harinder Singh Baw
Peter Onyisi (Atlas
Peter Onyisi (UTexa
David Lesny (AtlasC
AutoPyFactory
Giordon Stark
Slot Not Ready
Jeff Dandoy
Jordan Webster
John Alison
Hannah Binney
yy5295
atlasconnect
tresreid
antonk
zihaoj
jenkins
okumura
Unclaimed
Driven by individuals, not a WMS load pattern
Users can access >1000 cores for short periods
Most flock from home Tier-3
Panda CONNECT usage (AutoPyFactory user)

Resource target challenges

Worker node IP connectivity

Software Access

Here of course we are speaking of software distributed via CVMFS

Data Access

Six ways to CVMFS (1/2)

- **NativeCVMFS:** Install CVMFS on every node
  - cvmfs rpms are installed on every node by site administrators (standard for a WLCG site)
  - FUSE mounted http-based CVMFS file system, also compatibility libraries above SL6.x
  - Needs some local disk for the cache
  - Configure for ATLAS, OASIS and MWT2 repositories
  - Best performance
- **ParrotCVMFS:** I/O trap and redirect to a CVMFS Alien Cache
  - Emulates a NativeCVMFS installation but doesn't require FUSE kernel modules
  - Therefore no system changes needed by remote site administrators
  - Performance can be impacted depending on the application
  - For ATLAS, we found certain applications didn't perform well or caused other exceptions
- **nfsCVMFS**: Access CVMFS repositories via an NFS server
  - CVMFS client installed on NFS server
  - /cvmfs exported to compute nodes via NFS
  - Good performance

the WLCG / OSG standard

Deployed and operational on UChicago "Midway" cluster

Six ways to CVMFS (2/2)

- **PortableCVMFS:** User job mounts all repositories
  - Bring a CVMFS client with the job
  - Need to install FUSE and fuse kernel module
  - Needs some local disk for the cache
  - Same performance as NativeCVMFS

Deployed and operational on Illinois Campus Cluster (ICC)

- **Dependency bundling**
  - Use tools to gather dependencies and place into a package, for execution on remote sites: auditing step. (DASPOS helping here with Parrot and PTU)
  - CVMFS is only needed on an "auditing" host, not on the compute node
- **Stratum-R:** Replicate all repositories to a local Linux file system
  - Copy the compressed repositories to a local disk via `cvmfs_server snapshot` from a nearby stratum-1 (in our case, Fermilab)
  - Uncompress and replicate to local disk; becomes a **"Stratum-R", an rsync-able source of CVMFS managed software**
  - Rsync targets targets are shared file systems mounted to worker nodes (e.g. Stampede)

NEW! stratum-r concept under test

Stratum-R: our current best option

- For sites that want to use a project area on a shared filesystem like Lustre or GPFS
  - Campus clusters, XSEDE clusters, HPC sites
- Replicating CVMFS repositories to a Linux file system via “rsync” is not an option
  - Very slow - network latency
  - Also generates load on Squid proxies and Stratum-1 server
- Idea: Build a local Stratum-1 to bypass network, squid and keep overhead local
  - Use “`cvmfs_server snapshot`” to create a Stratum-1 replication (“**stratum-r**”)
  - Use snapshot to incrementally update
  - Install CVMFS client to use **stratum-r** as its source
  - Use “DIRECT” for Squid proxy
  - rsync from “/cvmfs” to local Linux file system on the stratum-r server (all I/O is local)
  - *Then, rsync to remote clusters in the normal way (one file system to another)*
- The target filesystem on the compute site should mounted to all worker nodes
  - **Symlink “/cvmfs” to the location of replicated repositories in the project area**
  - Jobs access repositories from the local disk copy,
    like any other project/application on the cluster

Only local admin action!

Conclusions

We finally are on the doorstep to accessing our allocation on Stampede!

Solution applicable for the next XSEDE or campus cluster

Applies to OASIS → for OSG and Campus sourced applications and software environments on XSEDE sites