Computation Institute

USATLAS MIDWEST TIER2

ATLAS Connect Status

Rob Gardner

Computation and Enrico Fermi Institutes

University of Chicago

US ATLAS Computing Facility Workshop at SLAC

April 7, 2014

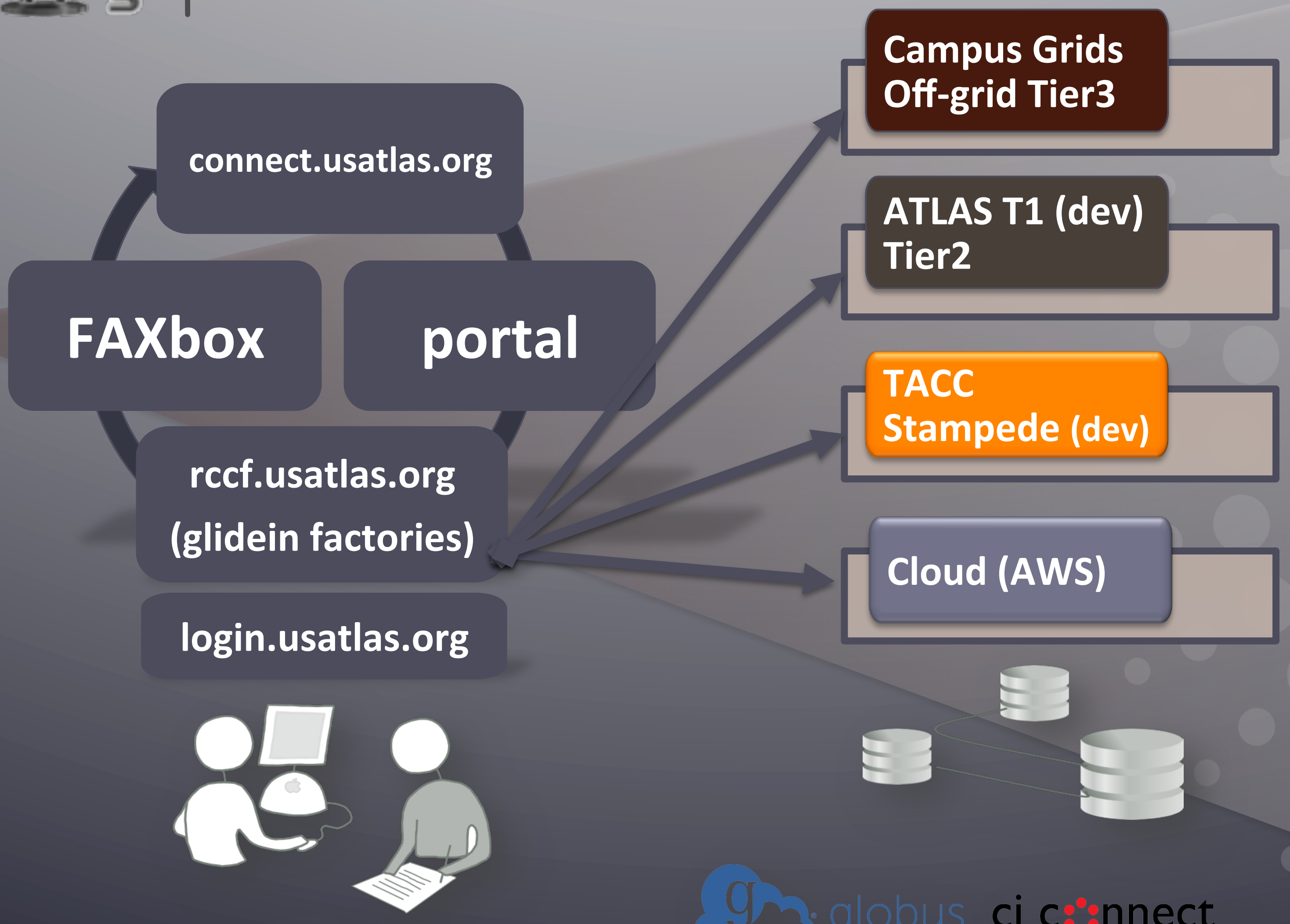
THE UNIVERSITY OF CHICAGO

efi.uchicago.edu
ci.uchicago.edu

Three Service Types

- ATLAS Connect **User**
  - A user login service with POSIX visible block storage
  - Similar to OSG Connect
- ATLAS Connect **Cluster**
  - Job flocking service from a Tier 3
- ATLAS Connect **Panda**
  - Connect Panda to non-grid resources (cloud, campus clusters, and some HPC centers)

CONNECT user



connect.usatlas.org

FAXbox

portal

rccf.usatlas.org

(glidein factories)

login.usatlas.org

Campus Grids

Off-grid Tier3

ATLAS T1 (dev)

Tier2

TACC

Stampede (dev)

Cloud (AWS)

globus ci connect

US ATLAS Computing Facility

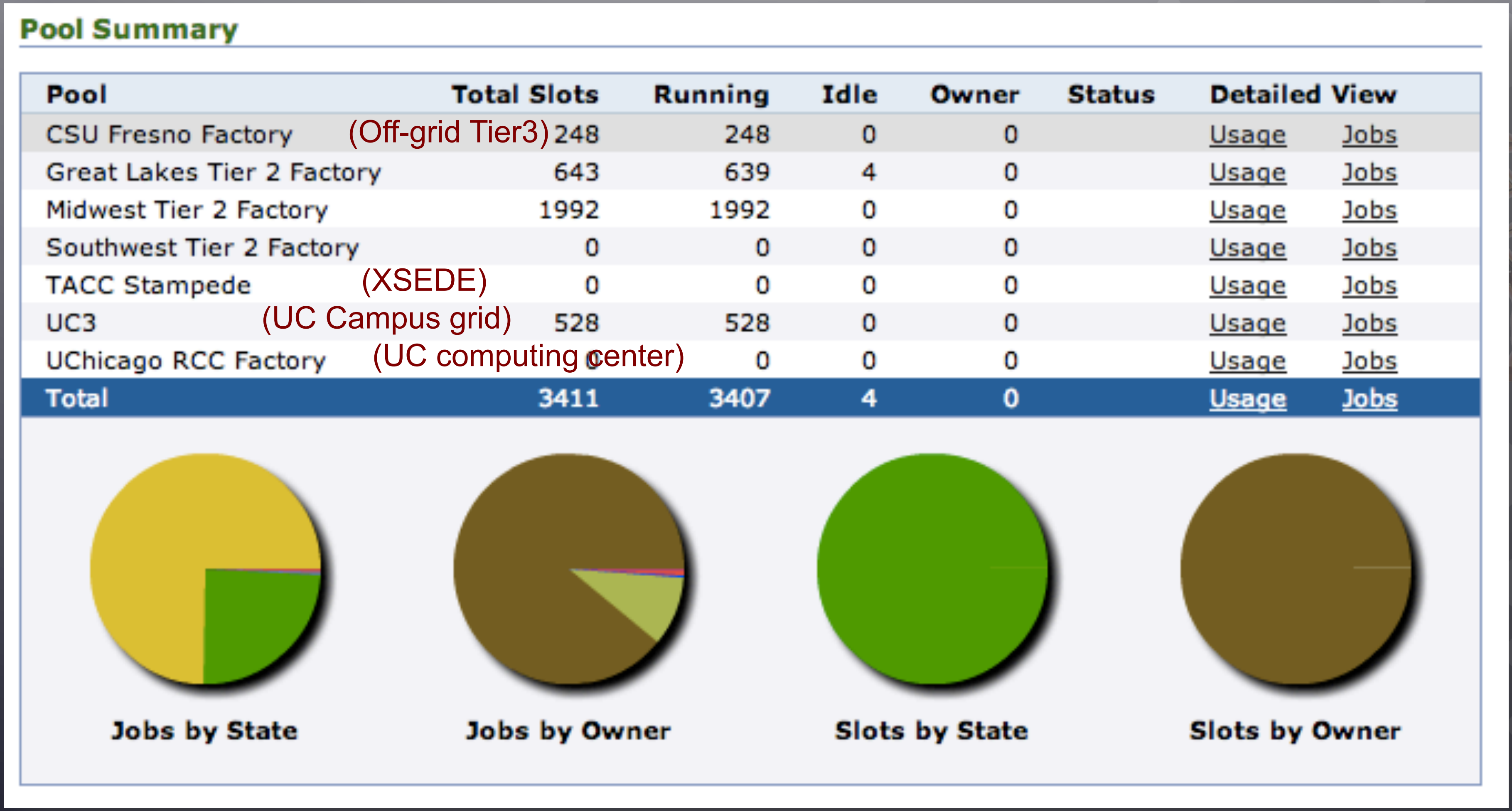
Open Science Grid

Looks like a very large virtual Tier3

- Users want to see quick, immediate “local” batch service
- We want to give them the illusion of control through availability
- Most Tier3 batch use is **very** spikey
- Use beyond pledge and other opportunistic resources to elastically absorb periods of peak demand
- Easily adjust virtual pool size according to US ATLAS priorities

Current resource targets

- Pool size varies depending on demand, matchmaking, priority at resource

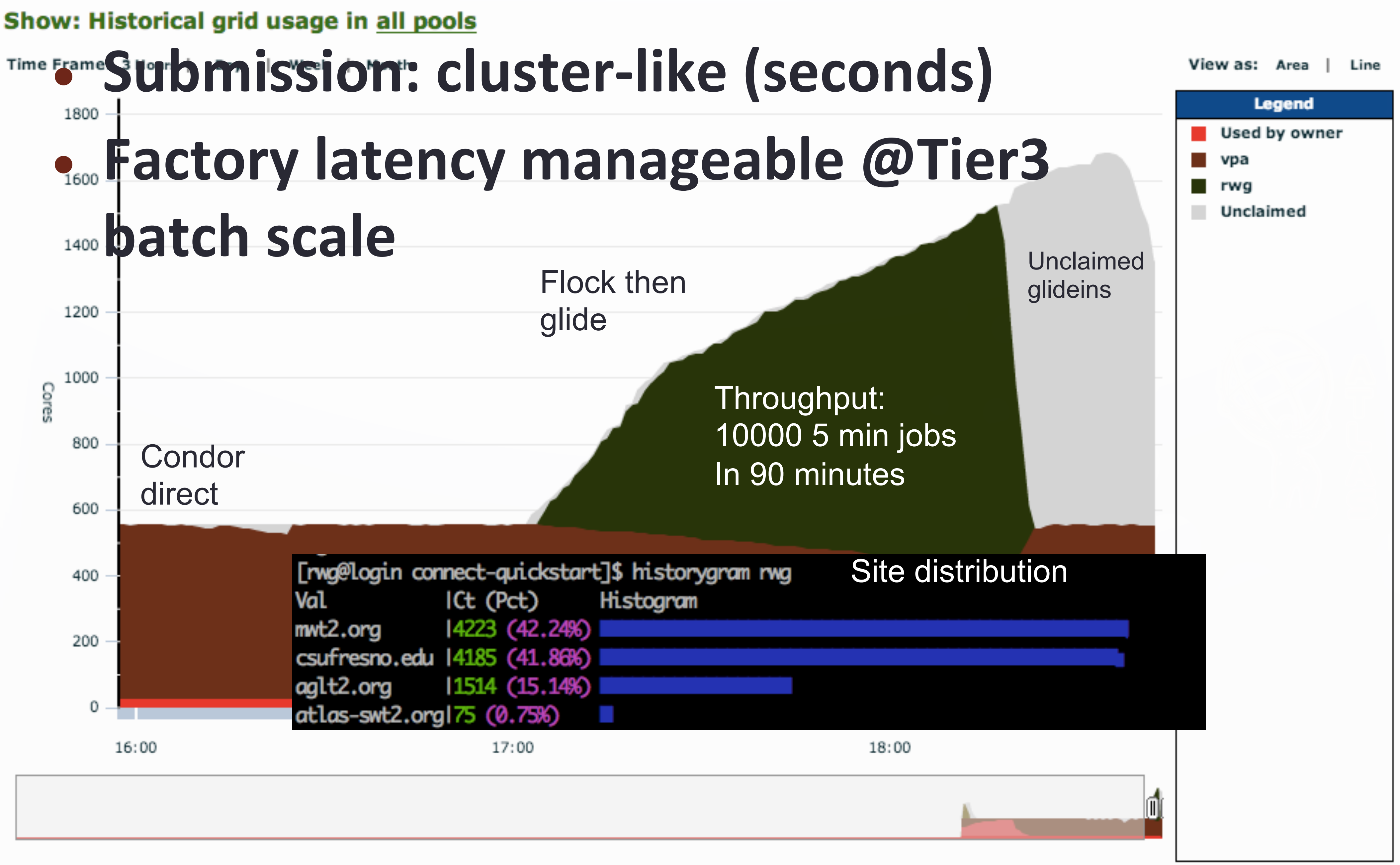


Pool Summary

Pool	Total Slots	Running	Idle	Owner	Status	Detailed View	
CSU Fresno Factory (Off-grid Tier3)	248	248	0	0		Usage	Jobs
Great Lakes Tier 2 Factory	643	639	4	0		Usage	Jobs
Midwest Tier 2 Factory	1992	1992	0	0		Usage	Jobs
Southwest Tier 2 Factory	0	0	0	0		Usage	Jobs
TACC Stampede (XSEDE)	0	0	0	0		Usage	Jobs
UC3 (UC Campus grid)	528	528	0	0		Usage	Jobs
UChicago RCC Factory (UC computing center)	0	0	0	0		Usage	Jobs
Total	3411	3407	4	0		Usage	Jobs

Connect is very quick relative to grid

- Submission: cluster-like (seconds)
- Factory latency manageable @Tier3 batch scale



Condor direct

Flock then glide

Throughput:
10000 5 min jobs
In 90 minutes

Unclaimed glideins

Site distribution

```
[rwg@login connect-quickstart]$ historygram rwg
Val            |Ct (Pct)       Histogram
mwt2.org       |4223 (42.24%)
csufresno.edu  |4185 (41.86%)
aglt2.org      |1514 (15.14%)
atlas-swt2.org|75 (0.75%)
```

Transient User Storage: **FAXbox**

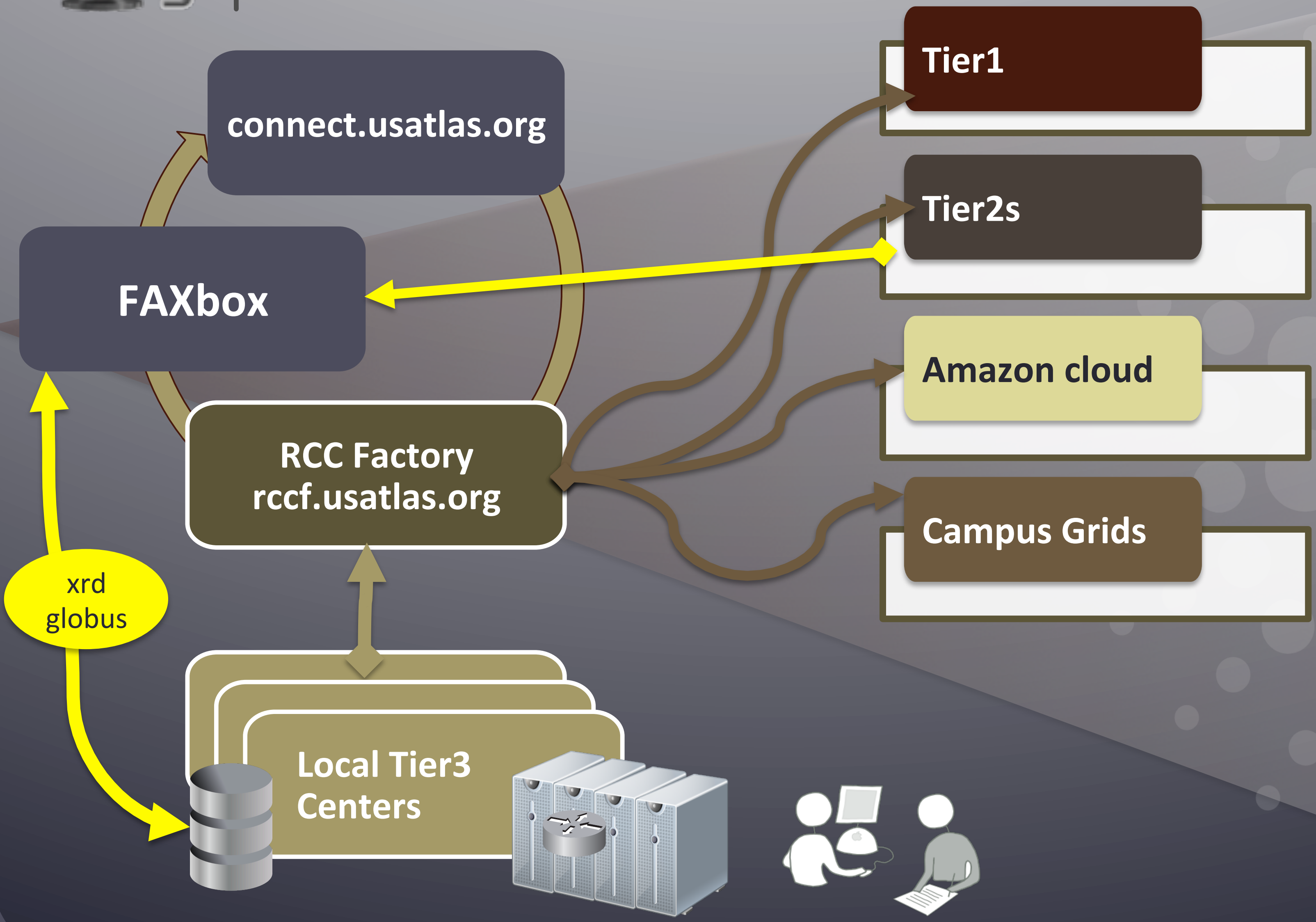
Similar to OSG Stash

- Assist ATLAS Connect User and flocked jobs via ATLAS Connect Cluster
  - Pre-stage data, write outputs for later use, etc.
- Use standard Xrootd tools and protocol
    - root://faxbox.usatlas.org/user/netID/file
    - Therefore read from anywhere, even a prun job
    - Will include a user quota system, and monitoring tools
- POSIX, Globus Online and http access too
- User managed, not ADC managed
- KIS

Tier3 to Tier2 flocking

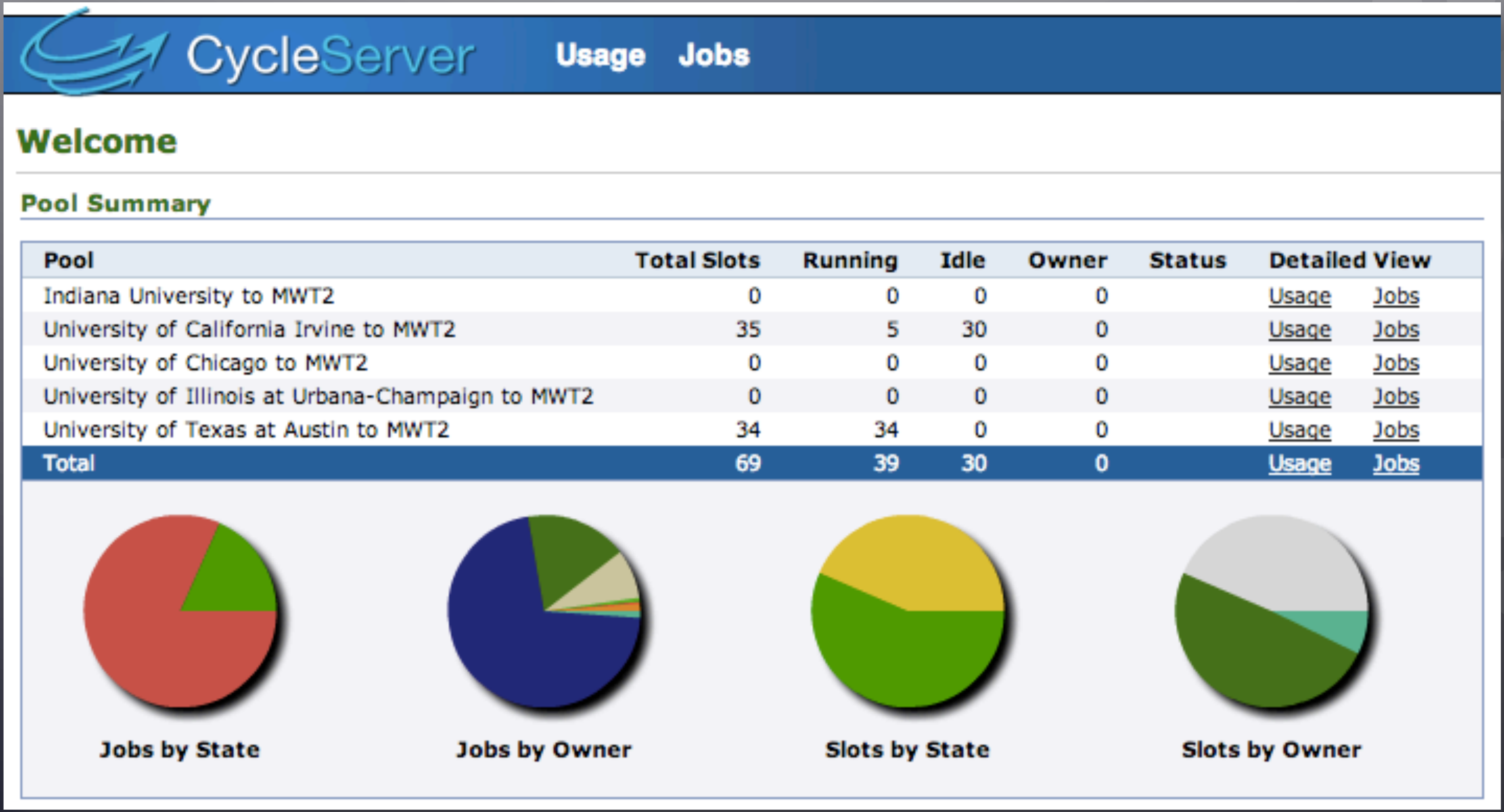
- This is ATLAS Connect Cluster
- Tier3 HTCondor as the local scheduler
  - Configure schedd to flock to the RCCF service
  - The RCCF service can reach any of the targets in the ATLAS Connect system
    - But for simplicity we configure it to submit to a large “nearby” Tier2 which has plenty of slots for T3 demand
    - Easily reconfigure for periods of high demand

CONNECT cluster



Tier3 to Tier2 flocking via ATLAS Connect

- Five Tier3 clusters configured in this way so far
- Works well, very low maintenance



Welcome

Pool Summary

Pool	Total Slots	Running	Idle	Owner	Status	Detailed View	
Indiana University to MWT2	0	0	0	0		Usage	Jobs
University of California Irvine to MWT2	35	5	30	0		Usage	Jobs
University of Chicago to MWT2	0	0	0	0		Usage	Jobs
University of Illinois at Urbana-Champaign to MWT2	0	0	0	0		Usage	Jobs
University of Texas at Austin to MWT2	34	34	0	0		Usage	Jobs
Total	69	39	30	0		Usage	Jobs

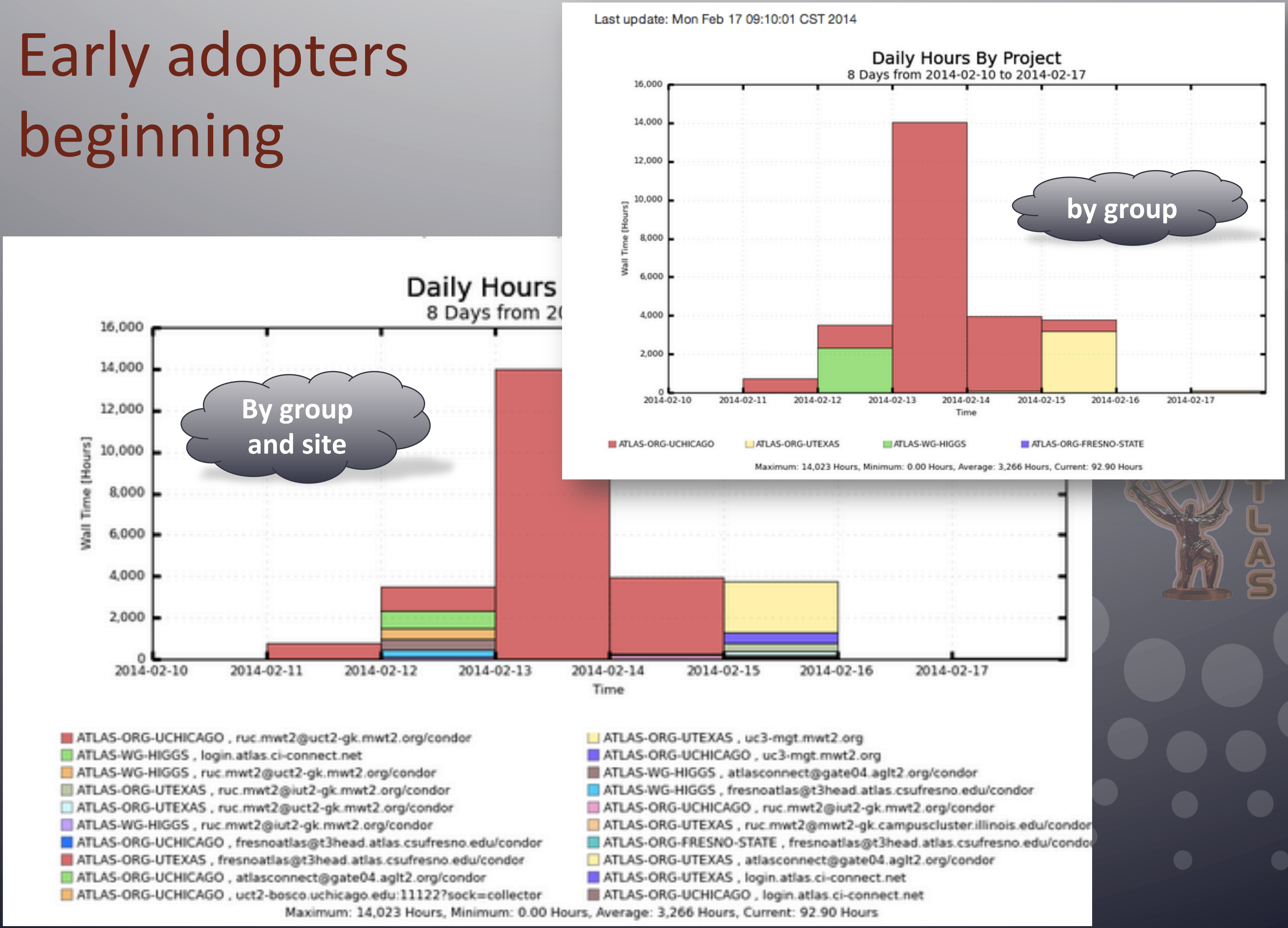
Yes, DHTC is a mode shift for local cluster users

- Users should not expect their home directories, NFS shares, or to even run jobs as their own user.
- Instead, HTCondor transfer mechanisms, FAX for data access, CVMFS for software
- Make use of ATLAS LOCALGROUPDISK's
- Smaller outputs (on the order of 1GB) can be handled by Condor's internal mechanisms
- Need to develop best practices and examples
  - Collect at http://connect.usatlas.org/ handbook

Integrating Off-Grid resources

- ATLAS Connect can be used to connect to off-grid resources:
  - Accessible from ATLAS Connect User, Cluster or even Panda
- “Wrap” campus clusters and big targets such as from XSEDE
- XSEDE-Stampede, UC-Midway
- Minimize local IT support
  - Ideally, a user account, ssh tunnel is needed
  - A local squid helps but not required (use a nearby squid in US ATLAS)

Early adopters beginning



Last update: Mon Feb 17 09:10:01 CST 2014

by group

By group and site

Maximum: 14,023 Hours, Minimum: 0.00 Hours, Average: 3,266 Hours, Current: 92.90 Hours

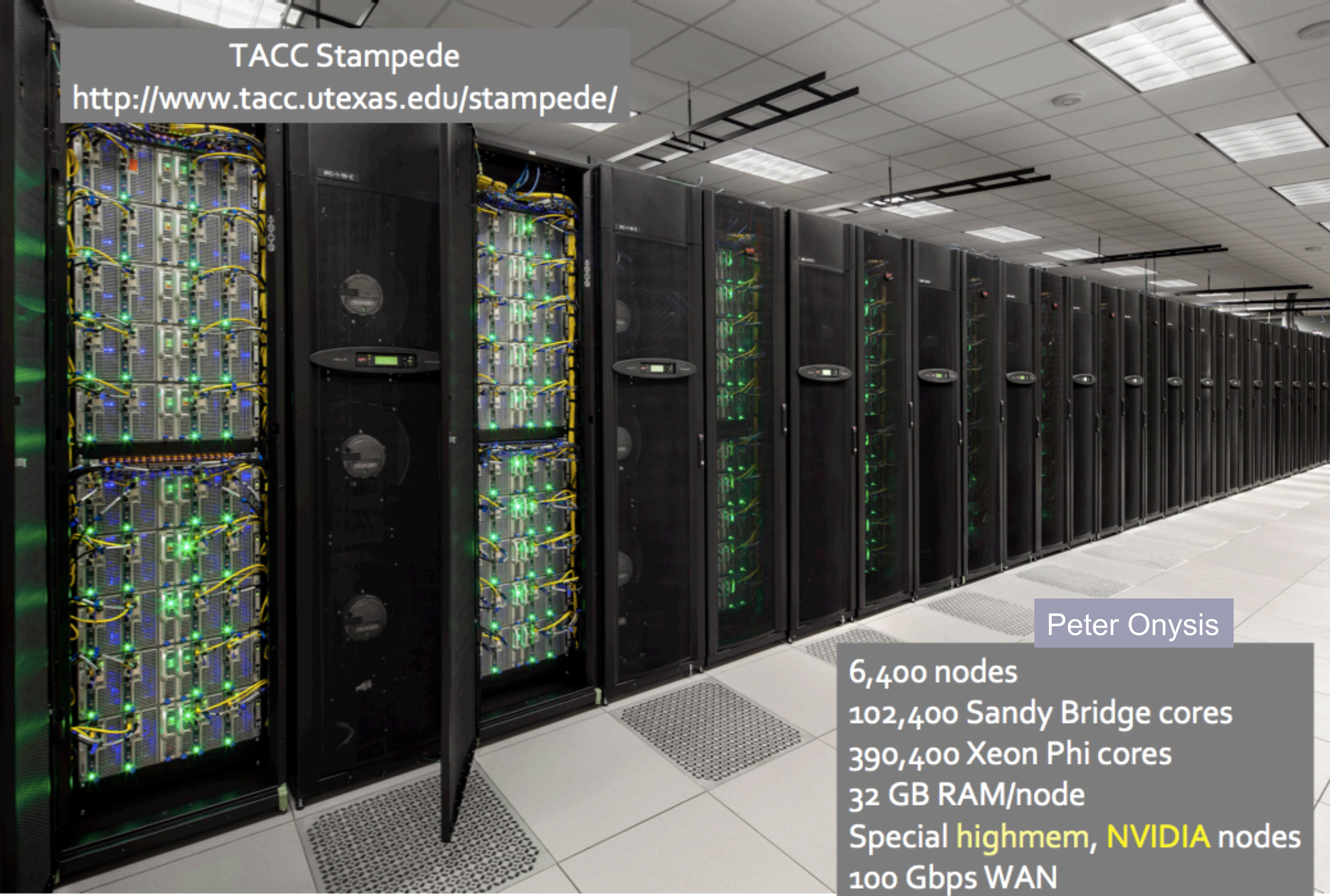
Maximum: 14,023 Hours, Minimum: 0.00 Hours, Average: 3,266 Hours, Current: 92.90 Hours

ATLAS and XSEDE

- Project to directly connect the ATLAS computing environment to TACC Stampede
- Central component of ATLAS Connect
  - Users (user login) ← from 44 US ATLAS institutions
  - Clusters (Tier3 flocking with HTCondor)
  - Central production from CERN (PanDA pilots)
- Integrating with a variety of tools
- Organized as XSEDE Science Gateway
  - Project Name: **ATLAS CONNECT** (TG-PHY140018) startup allocation
  - Principal Investigator: Peter Onyisi, University of Texas at Austin
  - Gateway team: Raminder Jeet, Suresh Maru, Marlon Pierce, Nancy Wilkins-Diehr
  - Stampede: Chris Hempel
  - US ATLAS Computing management: M. Ernst, R. Gardner
  - ATLAS Connect tech team: D. Lesny, L. Bryant, D. Champion

TACC Stampede

http://www.tacc.utexas.edu/stampede/



Peter Onysis

- 6,400 nodes
- 102,400 Sandy Bridge cores
- 390,400 Xeon Phi cores
- 32 GB RAM/node
- Special highmem, NVIDIA nodes
- 100 Gbps WAN
- 48 hr max job runtime

Approach

- Key is minimizing Stampede admin involvement while hiding complexity for users
  - Simple SSH to Stampede SLURM submit node
  - ATLAS software mounted using CVMFS and Parrot
  - ATLAS squid cache configured nearby
  - Wide area federated storage access
  - Maintain similar look and feel as native ATLAS nodes
- Leverages HTCondor, Glidein Factory, CCTools, OSG accounting and CI Connect technologies
  - User data staging + access, Unix accounts, groups, ID management → all handled outside XSEDE

ATLAS + XSEDE Status

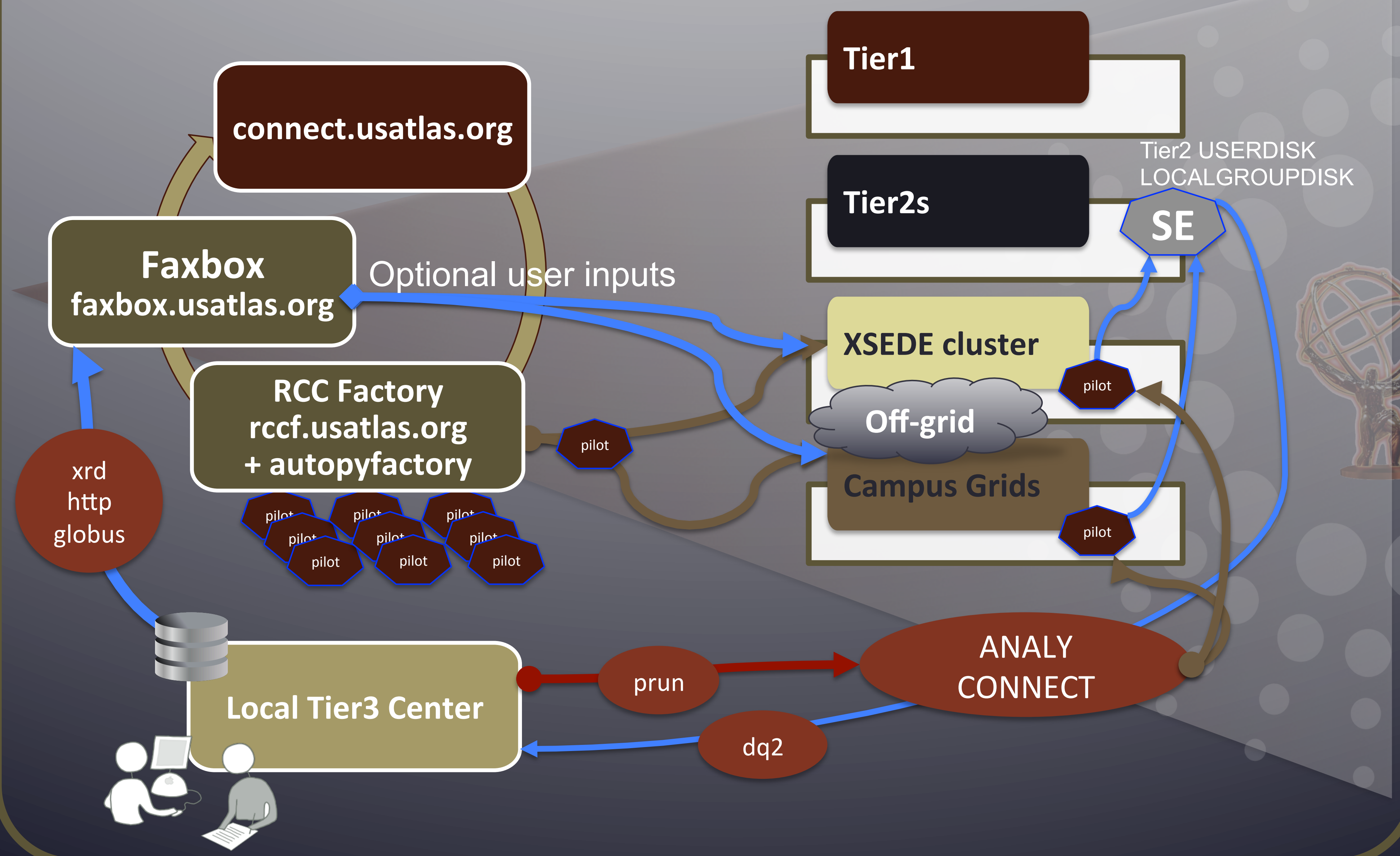
- Using SHERPA HEP Monte Carlo event generator and ROOT analysis of ATLAS data as representative applications
- Solution for scheduling multiple jobs in single Stampede job slot (16 cores)
- Using same approach for campus clusters
  - Useful for OSG Connect, campus grids, campus bridging

ATLAS + XSEDE Status: Panda CONNECT

- CONNECT queue created and configured
- APF deployed, working
- APF flocking to RCCF (glidein factory) tested and works well as expected
- Parrot wrappers to mount CVMFS repos
- Compatibility libraries needed on top of SL6 are provided by custom images created with fakeroot/fakechroot
- Race condition with Parrot under investigation by CCTools team at Notre Dame

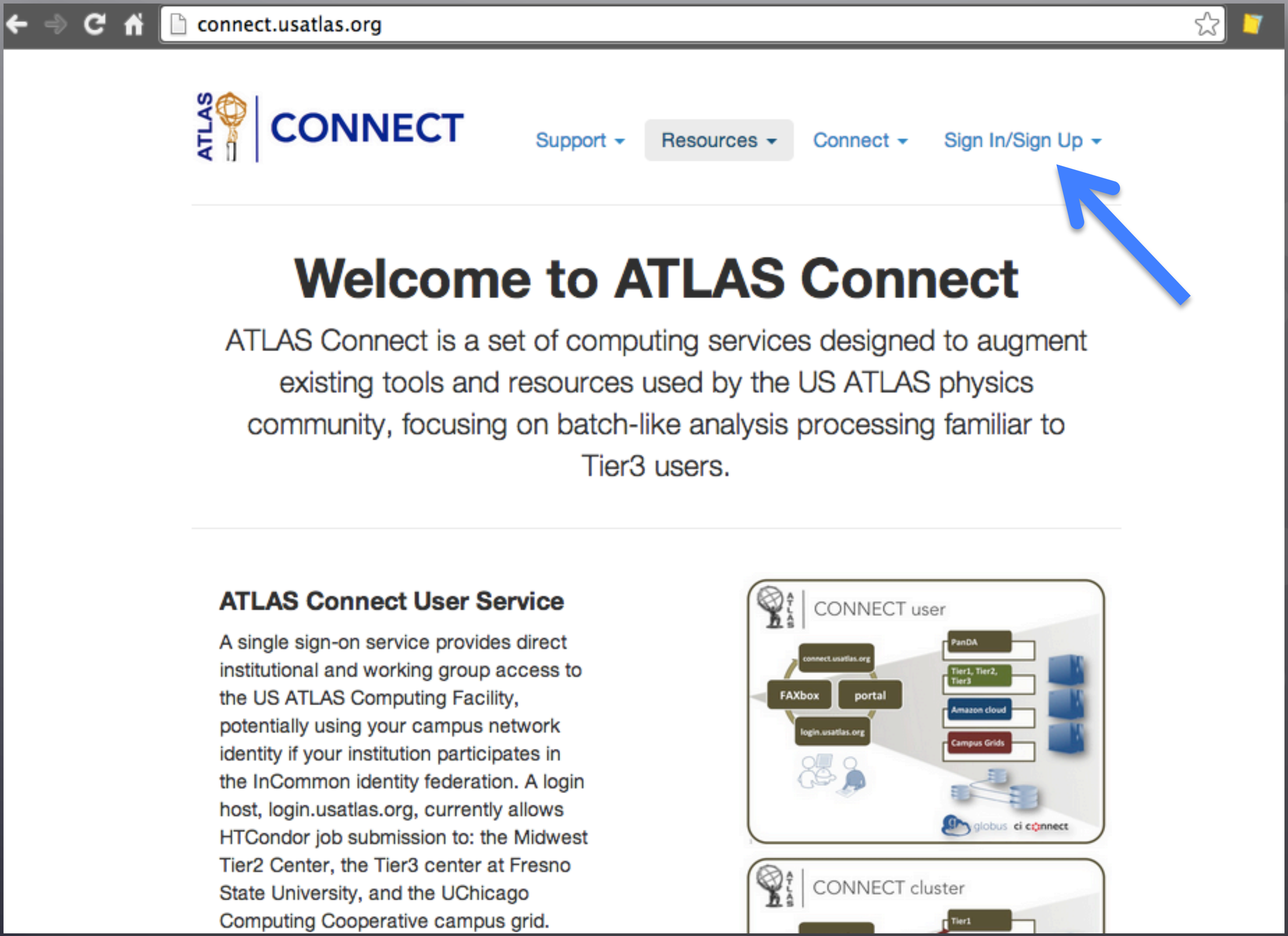
CONNECT panda

Analysis example



Login to the US ATLAS Computing Facility++

- Go to website, sign up with **your** campus ID*



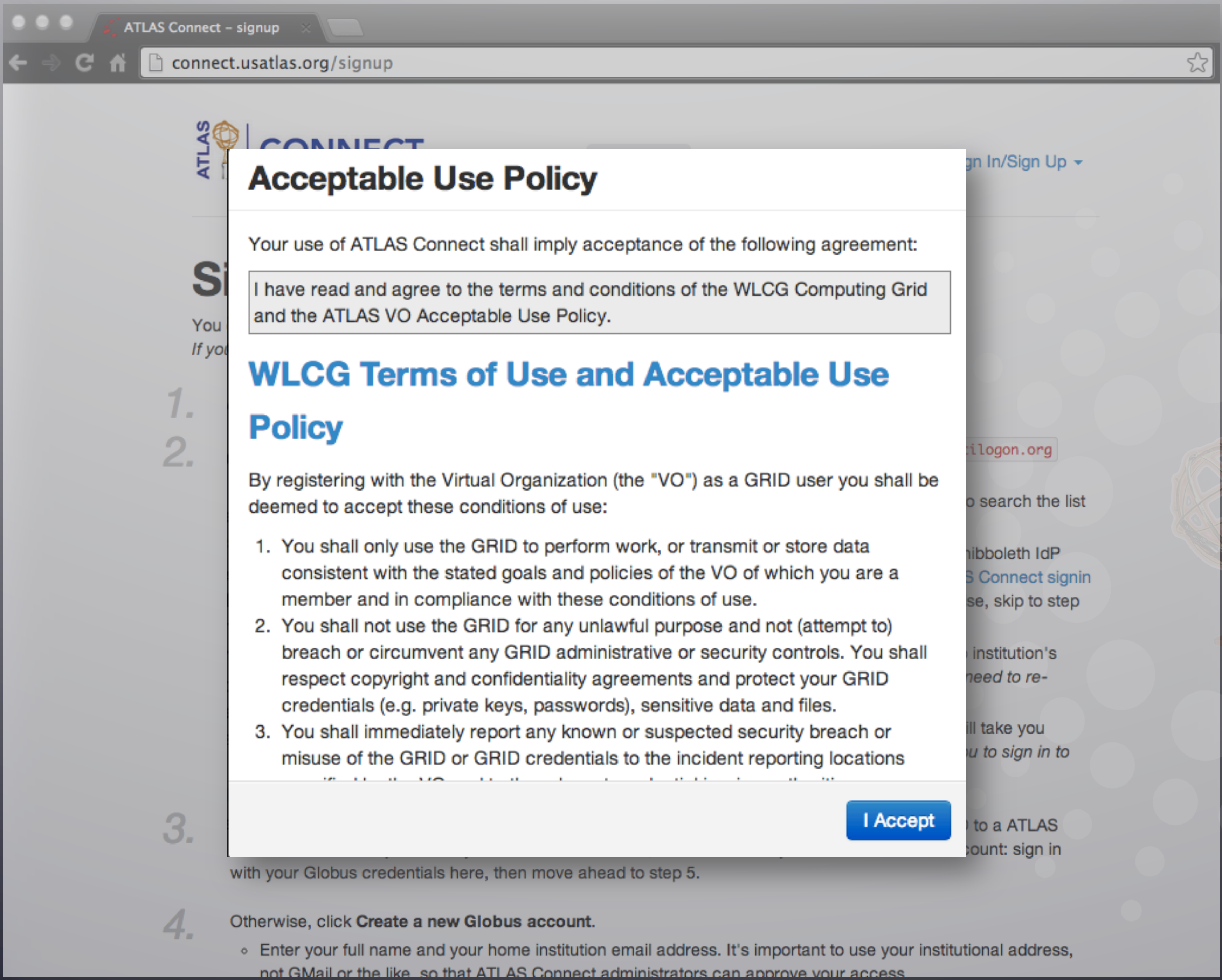
Welcome to ATLAS Connect

ATLAS Connect is a set of computing services designed to augment existing tools and resources used by the US ATLAS physics community, focusing on batch-like analysis processing familiar to Tier3 users.

ATLAS Connect User Service

A single sign-on service provides direct institutional and working group access to the US ATLAS Computing Facility, potentially using your campus network identity if your institution participates in the InCommon identity federation. A login host, login.usatlas.org, currently allows HTCondor job submission to: the Midwest Tier2 Center, the Tier3 center at Fresno State University, and the UChicago Computing Cooperative campus grid.

(*) Or Globus ID, or Google ID



Acceptable Use Policy

Your use of ATLAS Connect shall imply acceptance of the following agreement:

I have read and agree to the terms and conditions of the WLCG Computing Grid and the ATLAS VO Acceptable Use Policy.

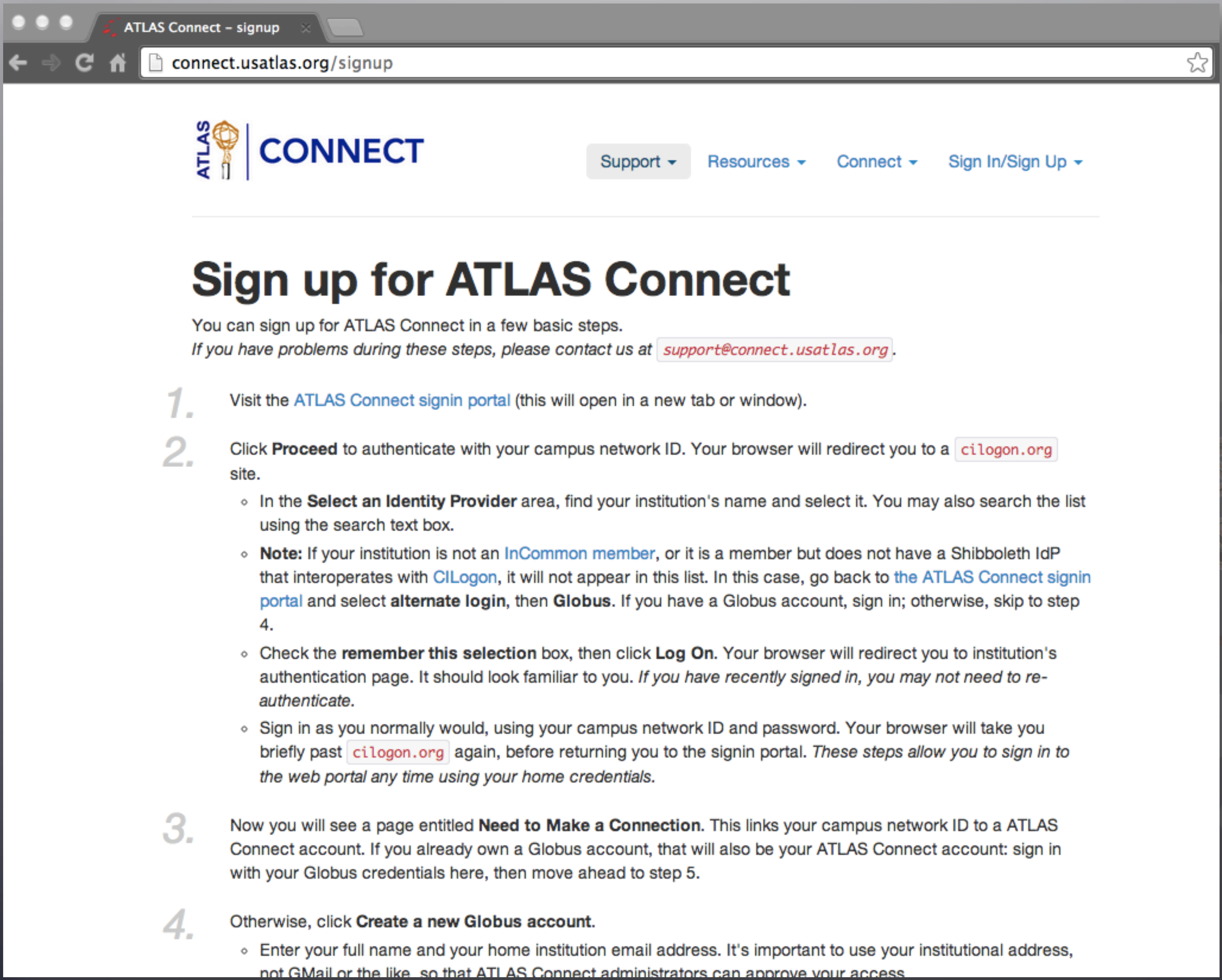
WLCG Terms of Use and Acceptable Use Policy

By registering with the Virtual Organization (the "VO") as a GRID user you shall be deemed to accept these conditions of use:

1. You shall only use the GRID to perform work, or transmit or store data consistent with the stated goals and policies of the VO of which you are a member and in compliance with these conditions of use.
2. You shall not use the GRID for any unlawful purpose and not (attempt to) breach or circumvent any GRID administrative or security controls. You shall respect copyright and confidentiality agreements and protect your GRID credentials (e.g. private keys, passwords), sensitive data and files.
3. You shall immediately report any known or suspected security breach or misuse of the GRID or GRID credentials to the incident reporting locations

with your Globus credentials here, then move ahead to step 5.

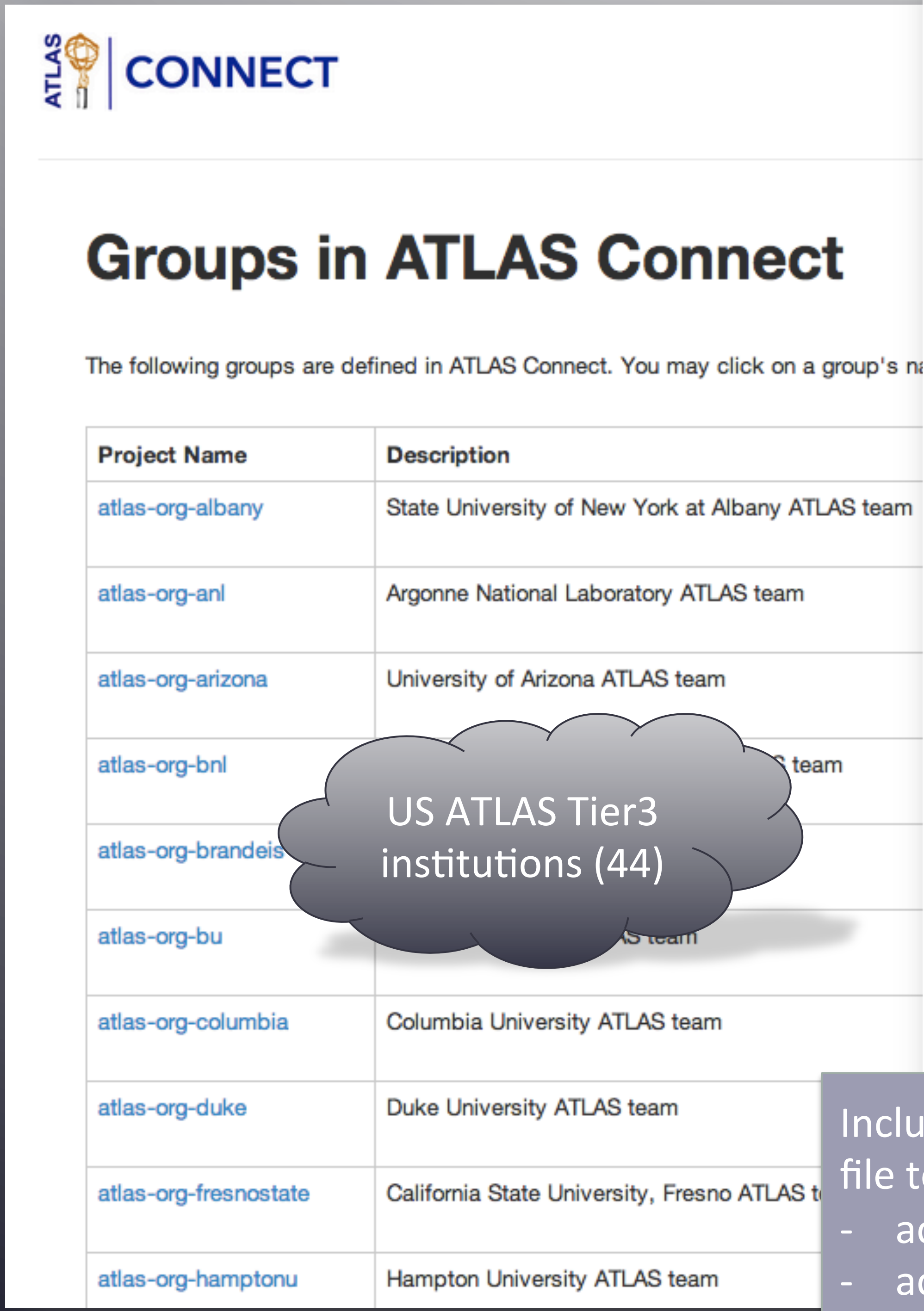
4. Otherwise, click **Create a new Globus account.**
   - Enter your full name and your home institution email address. It's important to use your institutional address, not GMail or the like, so that ATLAS Connect administrators can approve your access



Sign up for ATLAS Connect

You can sign up for ATLAS Connect in a few basic steps.
If you have problems during these steps, please contact us at `support@connect.usatlas.org`.

1. Visit the ATLAS Connect signin portal (this will open in a new tab or window).
2. Click **Proceed** to authenticate with your campus network ID. Your browser will redirect you to a `cilogon.org` site.
   - In the **Select an Identity Provider** area, find your institution's name and select it. You may also search the list using the search text box.
   - **Note:** If your institution is not an InCommon member, or it is a member but does not have a Shibboleth IdP that interoperates with CILogon, it will not appear in this list. In this case, go back to the ATLAS Connect signin portal and select **alternate login**, then **Globus**. If you have a Globus account, sign in; otherwise, skip to step 4.
   - Check the **remember this selection** box, then click **Log On**. Your browser will redirect you to institution's authentication page. It should look familiar to you. *If you have recently signed in, you may not need to re-authenticate.*
   - Sign in as you normally would, using your campus network ID and password. Your browser will take you briefly past `cilogon.org` again, before returning you to the signin portal. *These steps allow you to sign in to the web portal any time using your home credentials.*
3. Now you will see a page entitled **Need to Make a Connection**. This links your campus network ID to a ATLAS Connect account. If you already own a Globus account, that will also be your ATLAS Connect account: sign in with your Globus credentials here, then move ahead to step 5.
4. Otherwise, click **Create a new Globus account**.
   - Enter your full name and your home institution email address. It's important to use your institutional address, not GMail or the like, so that ATLAS Connect administrators can approve your access.

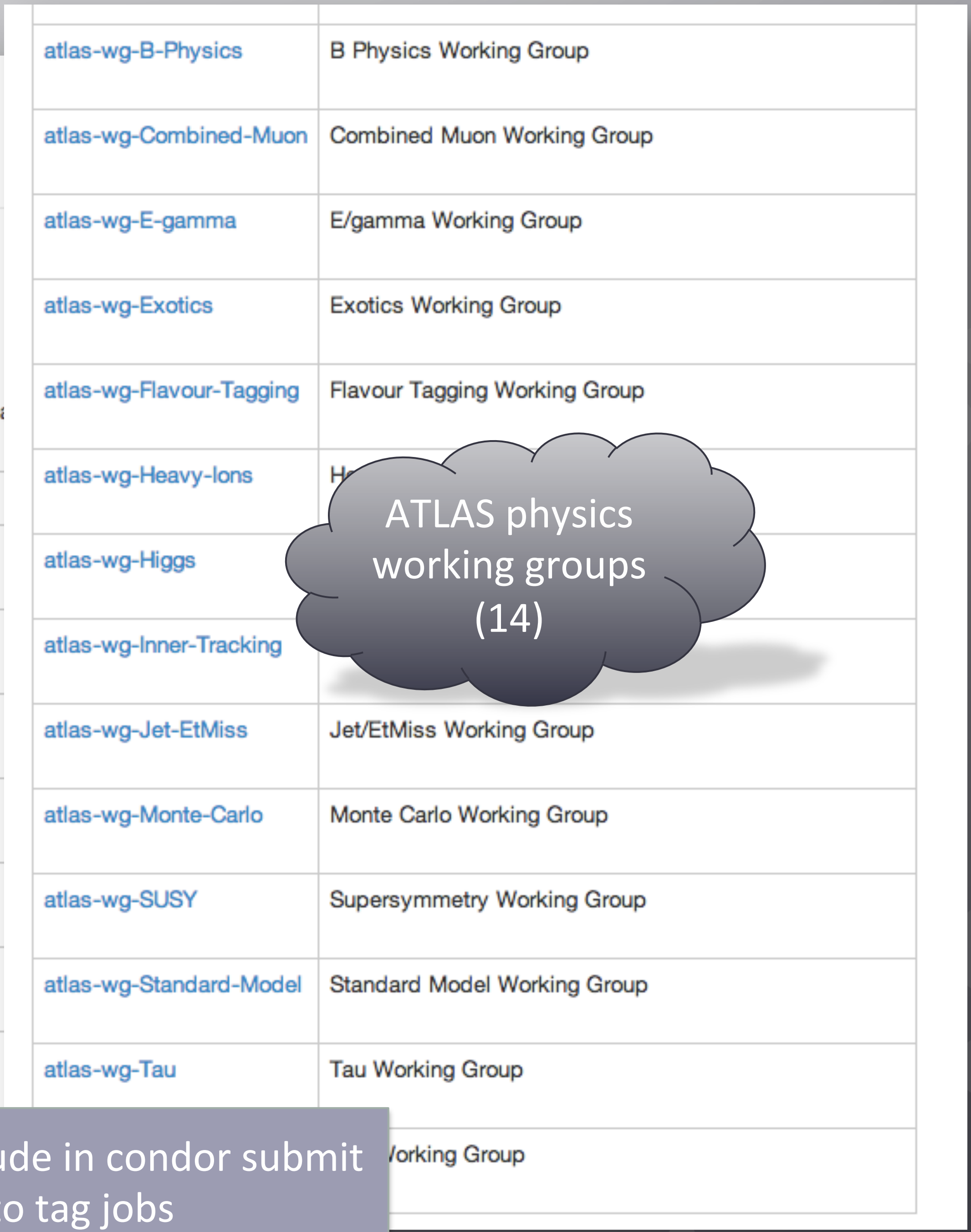


Groups in ATLAS Connect

The following groups are defined in ATLAS Connect. You may click on a group's na

Project Name	Description
atlas-org-albany	State University of New York at Albany ATLAS team
atlas-org-anl	Argonne National Laboratory ATLAS team
atlas-org-arizona	University of Arizona ATLAS team
atlas-org-bnl	team
atlas-org-brandeis	
atlas-org-bu	S team
atlas-org-columbia	Columbia University ATLAS team
atlas-org-duke	Duke University ATLAS team
atlas-org-fresnostate	California State University, Fresno ATLAS t
atlas-org-hamptonu	Hampton University ATLAS team

US ATLAS Tier3 institutions (44)

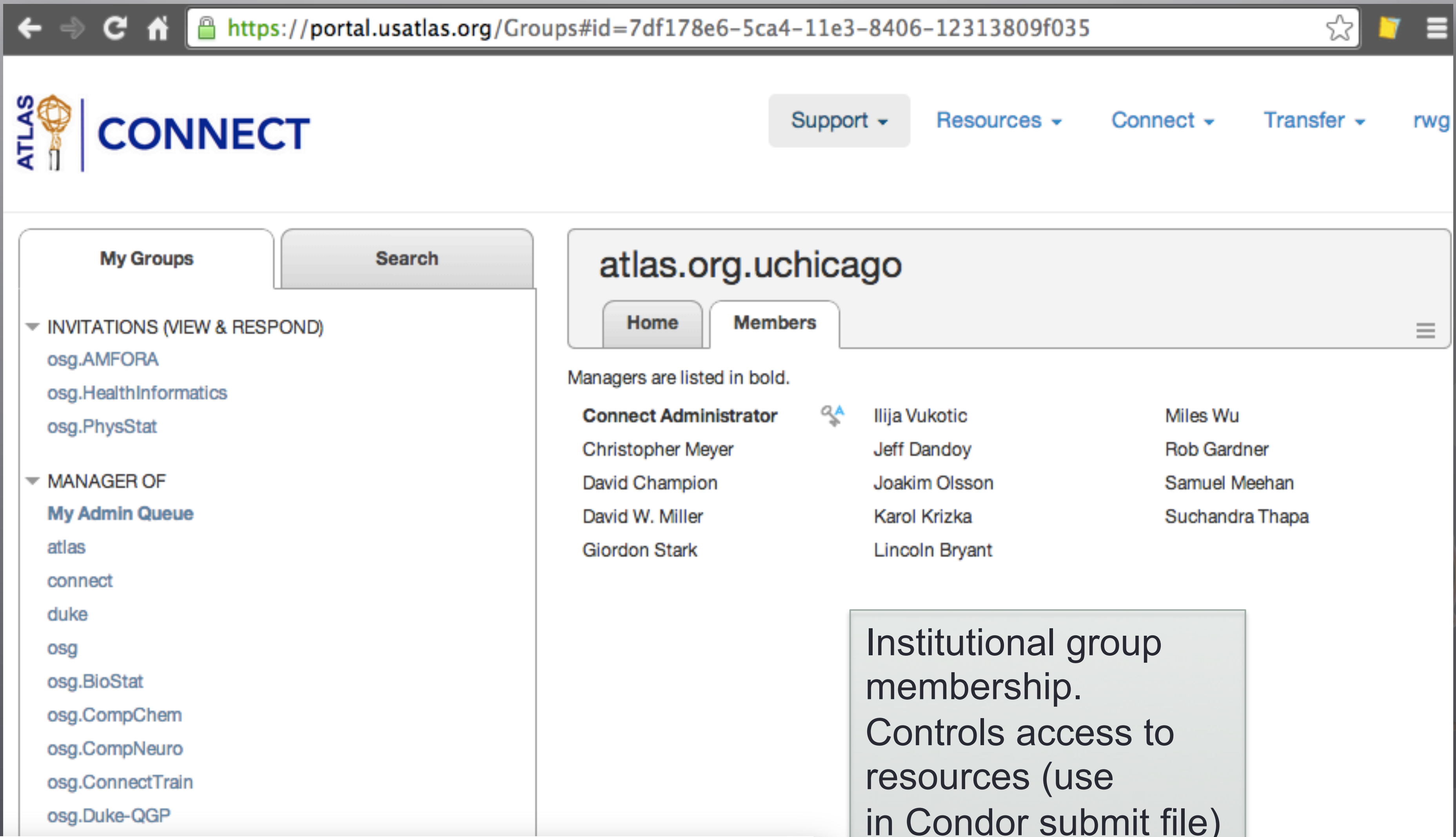


atlas-wg-B-Physics	B Physics Working Group
atlas-wg-Combined-Muon	Combined Muon Working Group
atlas-wg-E-gamma	E/gamma Working Group
atlas-wg-Exotics	Exotics Working Group
atlas-wg-Flavour-Tagging	Flavour Tagging Working Group
atlas-wg-Heavy-Ions	H
atlas-wg-Higgs	
atlas-wg-Inner-Tracking	
atlas-wg-Jet-EtMiss	Jet/EtMiss Working Group
atlas-wg-Monte-Carlo	Monte Carlo Working Group
atlas-wg-SUSY	Supersymmetry Working Group
atlas-wg-Standard-Model	Standard Model Working Group
atlas-wg-Tau	Tau Working Group
	/orking Group

ATLAS physics working groups (14)

Include in condor submit file to tag jobs

- access control
- accounting



https://portal.usatlas.org/Groups#id=7df178e6-5ca4-11e3-8406-12313809f035

INVITATIONS (VIEW & RESPOND)

- osg.AMFORA
- osg.HealthInformatics
- osg.PhysStat

MANAGER OF

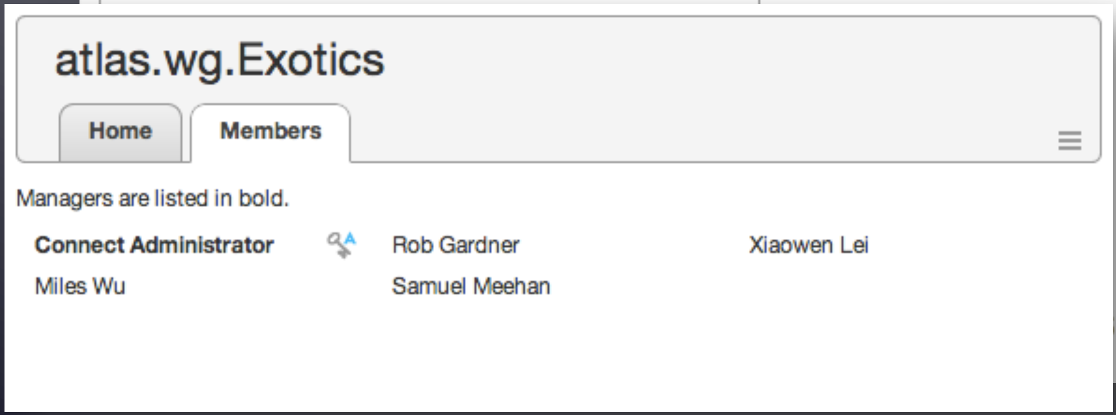
- **My Admin Queue**
- atlas
- connect
- duke
- osg
- osg.BioStat
- osg.CompChem
- osg.CompNeuro
- osg.ConnectTrain
- osg.Duke-QGP

atlas.org.uchicago

Managers are listed in bold.

Connect Administrator	Ilija Vukotic	Miles Wu
Christopher Meyer	Jeff Dandoy	Rob Gardner
David Champion	Joakim Olsson	Samuel Meehan
David W. Miller	Karol Krizka	Suchandra Thapa
Giordon Stark	Lincoln Bryant	

Institutional group membership. Controls access to resources (use in Condor submit file)

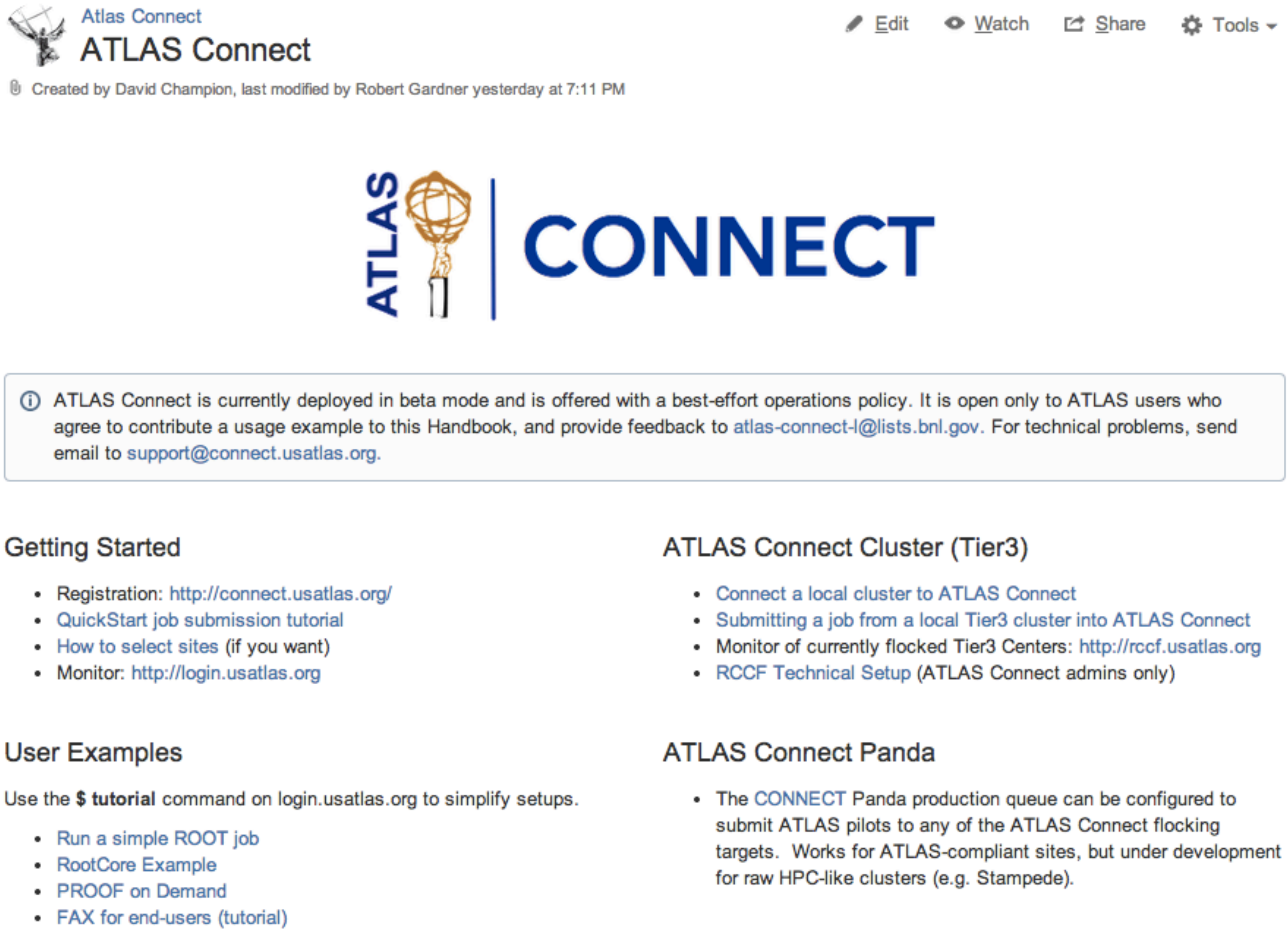


atlas.wg.Exotics

Managers are listed in bold.

Connect Administrator	Rob Gardner	Xiaowen Lei
Miles Wu	Samuel Meehan	

← Also have ATLAS physics working group tags.



Atlas Connect

ATLAS Connect

Created by David Champion, last modified by Robert Gardner yesterday at 7:11 PM

ATLAS Connect is currently deployed in beta mode and is offered with a best-effort operations policy. It is open only to ATLAS users who agree to contribute a usage example to this Handbook, and provide feedback to atlas-connect-l@lists.bnl.gov. For technical problems, send email to support@connect.usatlas.org.

Getting Started

- Registration: http://connect.usatlas.org/
- QuickStart job submission tutorial
- How to select sites (if you want)
- Monitor: http://login.usatlas.org

User Examples

Use the **$ tutorial** command on login.usatlas.org to simplify setups.

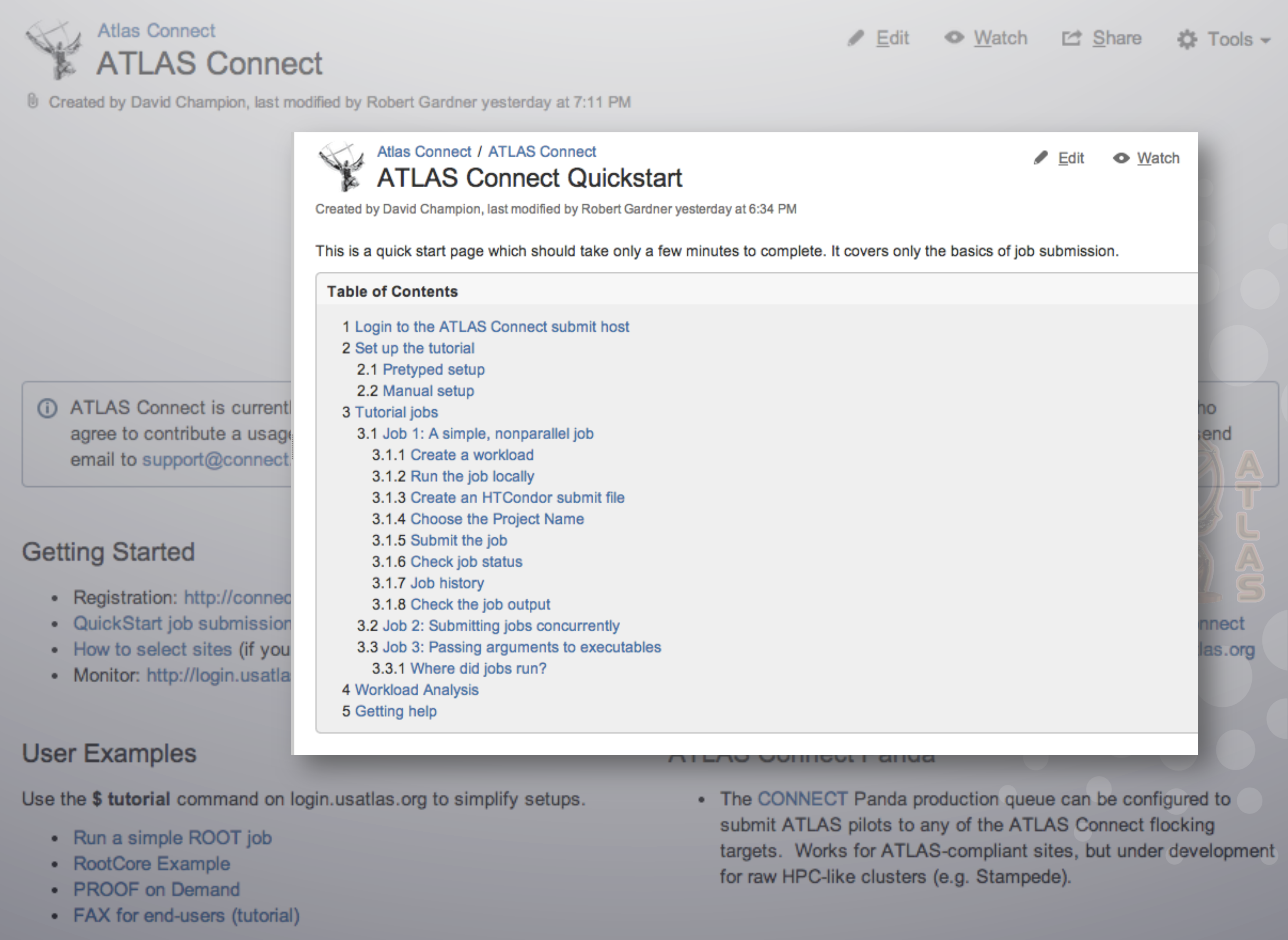
- Run a simple ROOT job
- RootCore Example
- PROOF on Demand
- FAX for end-users (tutorial)

ATLAS Connect Cluster (Tier3)

- Connect a local cluster to ATLAS Connect
- Submitting a job from a local Tier3 cluster into ATLAS Connect
- Monitor of currently flocked Tier3 Centers: http://rccf.usatlas.org
- RCCF Technical Setup (ATLAS Connect admins only)

ATLAS Connect Panda

- The CONNECT Panda production queue can be configured to submit ATLAS pilots to any of the ATLAS Connect flocking targets. Works for ATLAS-compliant sites, but under development for raw HPC-like clusters (e.g. Stampede).



Atlas Connect

ATLAS Connect

Created by David Champion, last modified by Robert Gardner yesterday at 7:11 PM

Atlas Connect / ATLAS Connect

ATLAS Connect Quickstart

Created by David Champion, last modified by Robert Gardner yesterday at 6:34 PM

This is a quick start page which should take only a few minutes to complete. It covers only the basics of job submission.

Table of Contents

ATLAS Connect is current ho
agree to contribute a usag end
email to support@connect

Getting Started

- Registration: http://connec
- QuickStart job submission
- How to select sites (if you
- Monitor: http://login.usatla

nnect

las.org

User Examples

Use the **$ tutorial** command on login.usatlas.org to simplify setups.

- Run a simple ROOT job
- RootCore Example
- PROOF on Demand
- FAX for end-users (tutorial)

ATLAS Connect Panda

- The CONNECT Panda production queue can be configured to submit ATLAS pilots to any of the ATLAS Connect flocking targets. Works for ATLAS-compliant sites, but under development for raw HPC-like clusters (e.g. Stampede).

Acknowledgements

USATLAS MIDWEST TIER2

- Dave Lesny – UIUC (MWT2)
- Lincoln Bryant, David Champion – UChicago (MWT2)
- Steve Teucke, Rachana Ananthakrishnan – (UChicago Globus)
- Ilija Vukotic (UChicago ATLAS)
- Suchandra Thapa (UChicago OSG)
- Peter Onysis – UTexas
- Jim Basney (CI-Logon) & InCommon Federation

Extras

AtlasTier1,2 vs Campus Clusters

Tier1,2 targets are known and defined

- CVMFS is installed and working
- Atlas repositories are configured and available
- Required Atlas RPMs are installed on all compute nodes

Campus Clusters are different

- CVMFS most likely not installed
- No Atlas repositories
- Unlikely that ATLAS required compatibility RPMs are installed

We could “ask” that these pieces be added, but we prefer to be unobtrusive

rccf.usatlas.org – Multiple Single User Bosco Instances

Remote Cluster Connect Factory (RCCF) or Factory

Single User Bosco Instance running as an RCC User on a unique SHARED_PORT

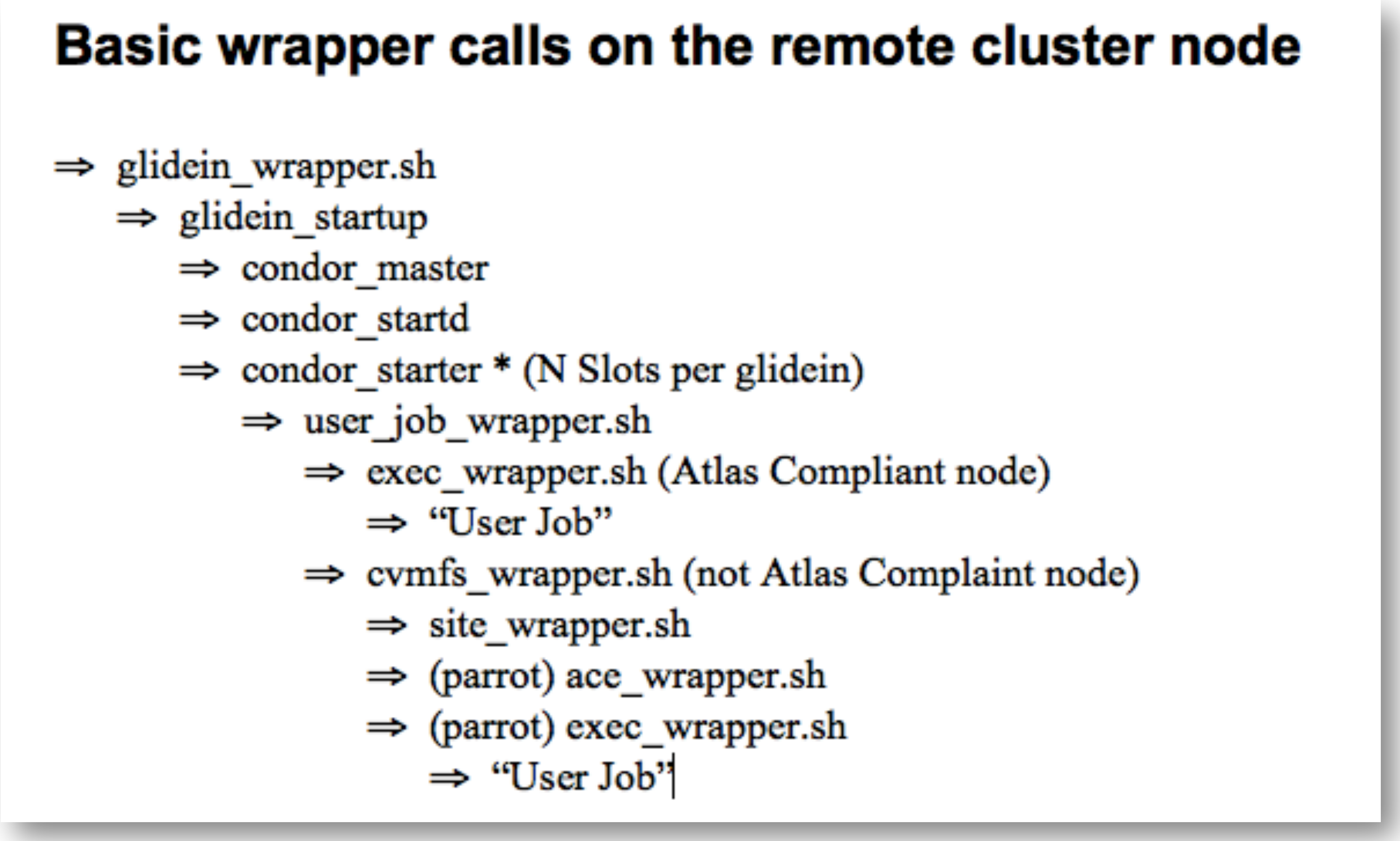
- Each RCCF is a separate Condor pool with a SCHEDD/Collector/Negotiator
- The RCCF injects glideins via SSH into a Target SCHEDD at MWT2
- The glidein creates a virtual job slot from Target SCHEDD to the RCCF
- Any jobs which are in that RCCF then run on that MWT2 Condor pool
- Jobs are submitted to the RCCF by flocking from a Source SCHEDD
- The RCCF can inject glideins to multiple Target SCHEDD hosts
- The RCCF can accept flocked jobs from multiple Source SCHEDD hosts
- Must have open bidirectional access to at least one port on Target SCHEDD
- Firewalls can create problems – SHARED_PORT makes it easier (single port)

Bosco modifications

- SSH to alternate ports (such as 2222)
- Multiple Bosco installations in the same user account on a remote target cluster
- Glidein support for the ACE (ATLAS Compatible Environment)
- Alternate location for “user job” sandbox on remote target cluster (ie /scratch)
- Slots per glidein and cores per slot
- Support for ATLAS Pilots in “native” and ACE
- Support for ClassAds such as HAS_CVMFS and IS_RCC
- Tunable Bosco parameters such as max idle glideinx, max running jobs, etc.
- Condor tuning for large large number of job submissions

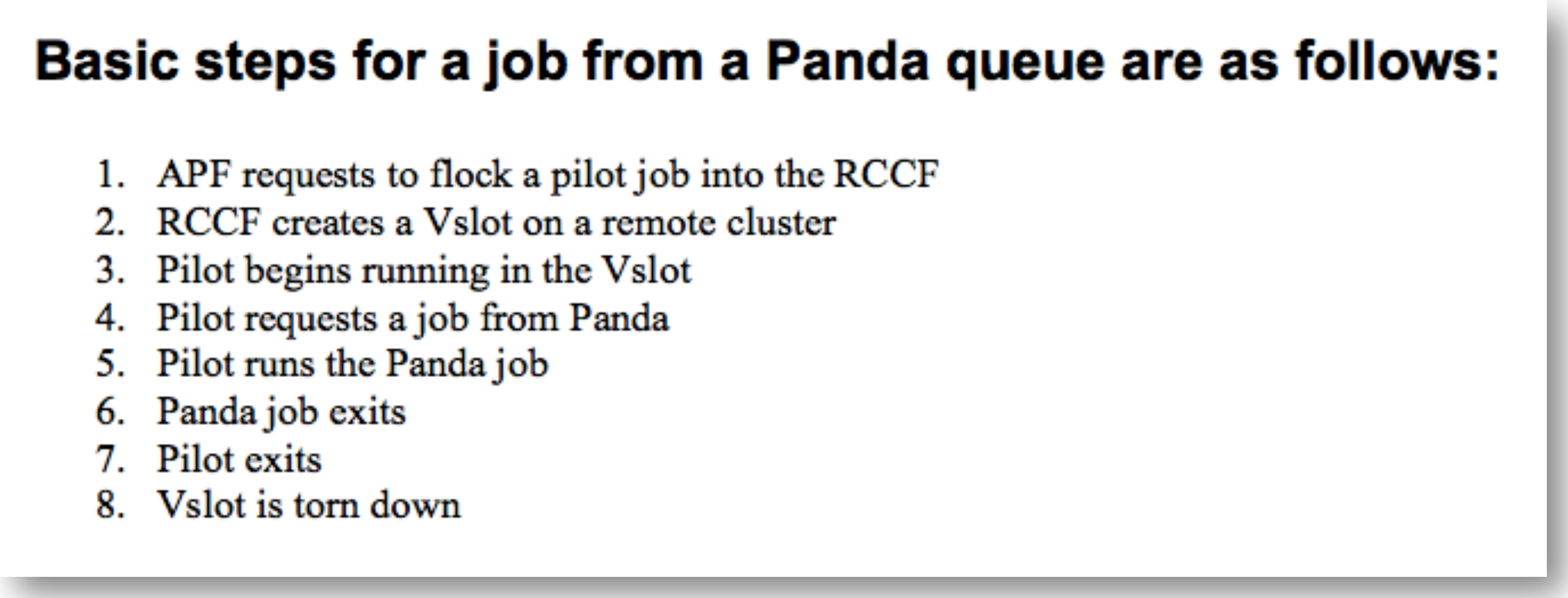
Basic job flow steps

- RCCF (Bosco) receives a request to run a user job from three flocking sources
  - Flock from the ATLAS Connect login host
  - Flock from an authorized Tier3 cluster
  - Flock from AutoPyFactory
  - Direct submission (testing only)
- RCCF creates a virtual slot(s) (Vslot) on a remote cluster
  - Running under a given user account
  - Number of Vslots site parameter (1, 2, 16) and cores (threads) per slot (1, 8)
  - If this is a not an ATLAS compliant cluster, an ACE Cache is created
- RCCF starts the job within the created Vslot running on the remote cluster



Basic wrapper calls on the remote cluster node

- ⇒ glidein_wrapper.sh
  - ⇒ glidein_startup
    - ⇒ condor_master
    - ⇒ condor_startd
    - ⇒ condor_starter * (N Slots per glidein)
      - ⇒ user_job_wrapper.sh
        - ⇒ exec_wrapper.sh (Atlas Compliant node)
          - ⇒ “User Job”
        - ⇒ cvmfs_wrapper.sh (not Atlas Complaint node)
          - ⇒ site_wrapper.sh
          - ⇒ (parrot) ace_wrapper.sh
          - ⇒ (parrot) exec_wrapper.sh
            - ⇒ “User Job”



Basic steps for a job from a Panda queue are as follows:

1. APF requests to flock a pilot job into the RCCF
2. RCCF creates a Vslot on a remote cluster
3. Pilot begins running in the Vslot
4. Pilot requests a job from Panda
5. Pilot runs the Panda job
6. Panda job exits
7. Pilot exits
8. Vslot is torn down

Source is always HTCondor

User submitting jobs always uses HTCondor submit files regardless of the target

User does not have to know what scheduler is used at the target (can add requirements if there is a preference)

```
Universe                  = Vanilla

Requirements              = ( IS_RCC ) && ((Arch == "X86_64") || (Arch == "INTEL"))

+ProjectName              = ”altas-org-utexas"

Executable                = gensherpa.sh

Should_Transfer_Files     = IF_Needed
When_To_Transfer_Output   = ON_Exit
Transfer_Output           = True
Transfer_Input_Files      = 126894_sherpa_input.tar.gz,MC12.126894.Sherpa_CT10_llll_ZZ.py
Transfer_Output_Files     = EVNT.pool.root
Transfer_Output_Remaps    = "EVNT.pool.root=output/EVNT_$(Cluster)_$(Process).pool.root"

Arguments                 = "$(Process) 750"

Log                       = logs/$(Cluster)_$(Process).log
Output                    = logs/$(Cluster)_$(Process).out
Error                     = logs/$(Cluster)_$(Process).err

Notification              = Never

Queue 100
```

Provide CVMFS via Parrot/CVMFS

Parrot/CVMFS (CCTools) has the ability to get all these missing elements

- CCTools, job wrapper and environment variables in a single tarball
- Tarball uploaded and unpacked on target as part of virtual slot creation
- Package only used on sites without CVMFS (Campus Clusters)

Totally transparent to the end user

- The wrapper executes the users job in the Parrot/CVMFS environment
- Atlas CVMFS repositories are available then available to the job
- With CVMFS we can also access the MWT2 CVMFS Server

CVMFS Wrapper Script

The CVMFS Wrapper Script is the glue that binds

- Defines Frontier Squids (site dependant list) for CVMFS
- Sets up access to MWT2 CVMFS repository
- Runs the users jobs in the Parrot/CVMFS environment

One missing piece remains to run Atlas jobs – Compatibility Libraries

HEP_OSlibs_SL6

Dumped all dependencies listed in HEP_OSlibs_SL6 1.0.15-1

Fetch all RPMS from Scientific Linux server

Many of these are not relocatable RPMs so used cpio to unpack

```
rpm2cpio $RPM| cpio --quiet --extract --make-directories --unconditional
```

Also added a few other RPMs not currently part of HEP_OSlibs

This creates a structure which looks like

```
drwxr-xr-x  2 ddl mwt2 4096 Feb 17 22:58 bin
drwxr-xr-x 15 ddl mwt2 4096 Feb 17 22:58 etc
drwxr-xr-x  6 ddl mwt2 4096 Feb 17 22:58 lib
drwxr-xr-x  4 ddl mwt2 4096 Feb 17 22:58 lib64
drwxr-xr-x  2 ddl mwt2 4096 Feb 17 22:58 sbin
drwxr-xr-x  9 ddl mwt2 4096 Feb 17 22:57 usr
drwxr-xr-x  4 ddl mwt2 4096 Feb 17 22:57 var
```

New: now providing this separately as bundle to avoid CVMFS conflicts

User Job Wrapper

- Setup a minimum familiar environment for the user
- We are not trying to create a pilot
- Print a job header to help us know when and where the job ran
  - Date, User and hostname the job is running on
  - Should we put other information into the header?
- Define some needed environment variables
  - $PATH – System paths (should we add /usr/local, etc)
  - $HOME – Needed by ROOT and others
  - $XrdSecGSISRVNAME – Works around a naming bug
  - $IS_RCC=True
  - $IS_RCC_<factory>=True
- Exec the user “executable”

User Job Wrapper – Internal Vars

- Other variables a user might want
  - $_RCC_Factory=<factory>
  - $_RCC_Port=<RCC Factory Port>
  - $_RCC_MaxIdleGlideins=nnn
  - $_RCC_IterationTime=<minutes>
  - $_RCC_MaxQueuedJobs=nnn
  - $_RCC_MaxRunningJobs=nnn
  - $_RCC_BoscoVersion=<bosco version>

Puppet Rules

- bosco_factory – Create a RCC Factory
    - Define the user account and shared port factory runs in
    - Other parameters to change max glideins, max running, etc
    - User account must exist on uct2-bosco (puppet rule)
    - Installs bosco, modifies some files, copies host certificate

- bosco_cluster – Create a Bosco Cluster to a Target SCHEDD
    - Creates Bosco Cluster to Target SCHEDD
    - User account must exist at Target and have SSH keys access
    - User account can be anything Target SCHEDD admin allows
    - Pushes job wrapper, condor_submit_attributes, etc

Puppet Rules

- bosco_flock – Allow a Source SCHEDD to flock to this Factory
  - Source SCHEDD FDQN
  - For GSI – DN of the Source SCHEDD node

- bosco_require – Add a “Requirement” (classAD) to a slot
  - Allows one to add a classAD to a slot
  - For example - HAS_CVMFS
  - Two classADs added to a factory by default
    - IS_RCC = True
    - IS_RCC_<factory nickname> = True
  - Remote Users can use these in their Condor submit file

Tier3 Source SCHEDD Condor Requirements

- We prefer to use GSI Security
- Source SCHEDD must have a working Certificate Authority (CA)
- Source SCHEDD must have a valid host certificate key pair
- Use the FQDN and DN of the Source SCHEDD in the bosco_flock
- If a site cannot use GSI for some reason we can use CLAIMTOBE
- Host based security not as secure (man in the middle attack)

Tier3 Source SCHEDD Condor Configuration Additions

```
# Setup the FLOCK_TO the RCC Factory
FLOCK_TO = $(FLOCK_TO), uct2-bosco.uchicago.edu:<RCC_Factory_Port>?sock=collector

# Allow the RUC Factory server access to our SCHEDD
ALLOW_NEGOTIATOR_SCHEDD = $(CONDOR_HOST), uct2-bosco.uchicago.edu

# Who do you trust?
GSI_DAEMON_NAME = $(GSI_DAEMON_NAME), /DC=com/DC=DigiCert-Grid/O=Open Science Grid/OU=Services/CN=uct2-bosco.uchicago.edu
GSI_DAEMON_CERT = /etc/grid-security/hostcert.pem
GSI_DAEMON_KEY = /etc/grid-security/hostkey.pem
GSI_DAEMON_TRUSTED_CA_DIR = /etc/grid-security/certificates

# Enable authentication from tne Negotiator (This is required to run on glidein jobs)
SEC_ENABLE_MATCH_PASSWORD_AUTHENTICATION = TRUE
```

Performance

- Jobs will run the same no matter how they arrive on an MWT2 worker node
- Submission rates (Condor submit to Execution) are the key
- Local submission involves only local SCHEDD/Negotiator/Collector
- Remote Flocking has multiple steps
  1. Local submission with SCHEDD and Negotiator
  2. Local SCHEDD contacts RUC Factory Negotiator
  3. RCC Factory Negotiator matches jobs to itself and they flock
  4. Factory SSH into an MWT2 SCHEDD and creates a virtual slot
  5. Job begins execution in a free virtual slot on the MWT2 worker node

Performance

- Step 4 takes the longest time, but may not always happen
  1. SSH to SCHEDD
  2. Wait for a job slot to open in this SCHEDD Condor pool
  3. Create virtual slot from a worker node back to the RCC Factory

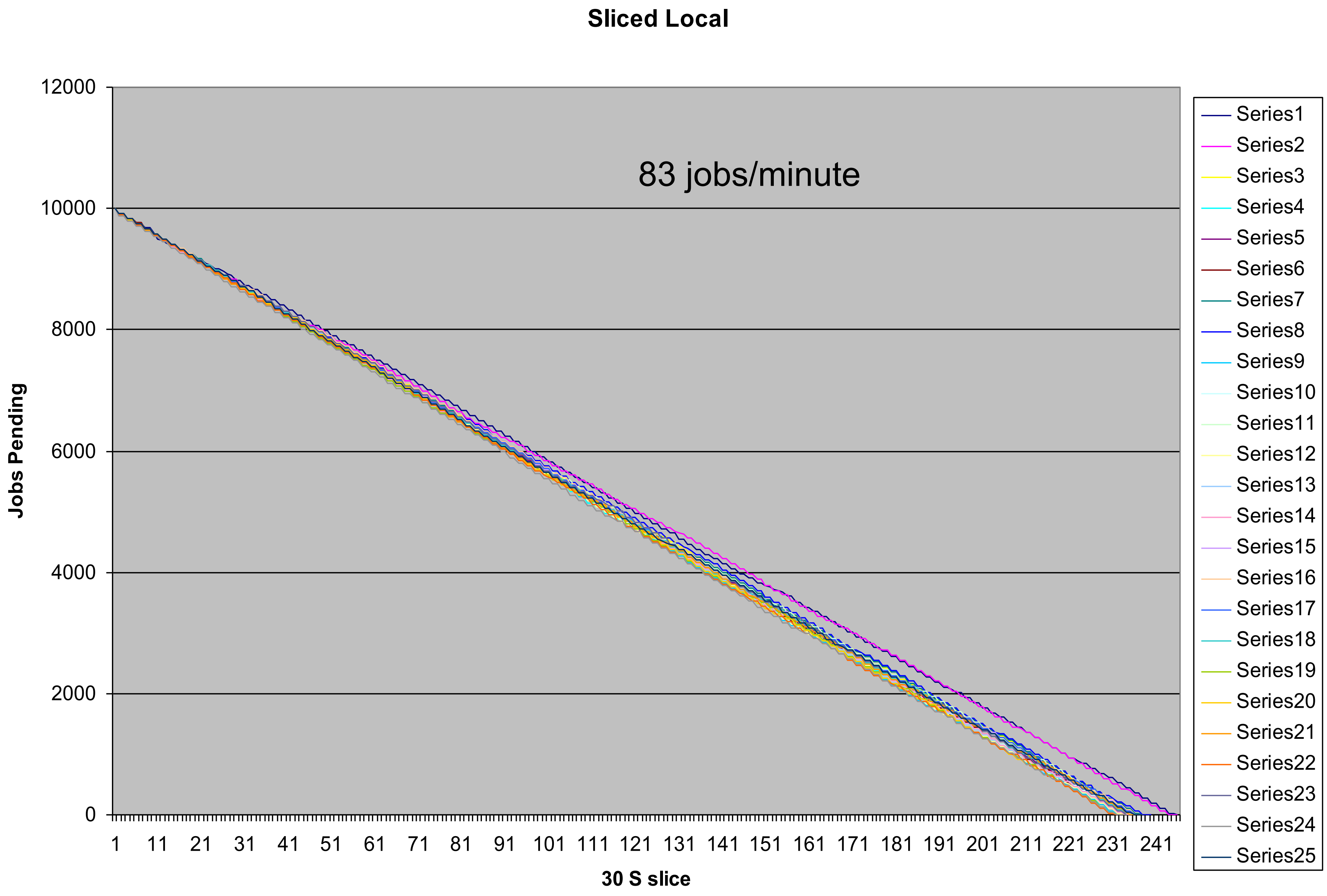
- Virtual slots remain for sometime in an “Unclaimed” state
- Unclaimed virtual slots are unused resources at MWT2
- Cannot keep them open forever or these resources are wasted

Performance

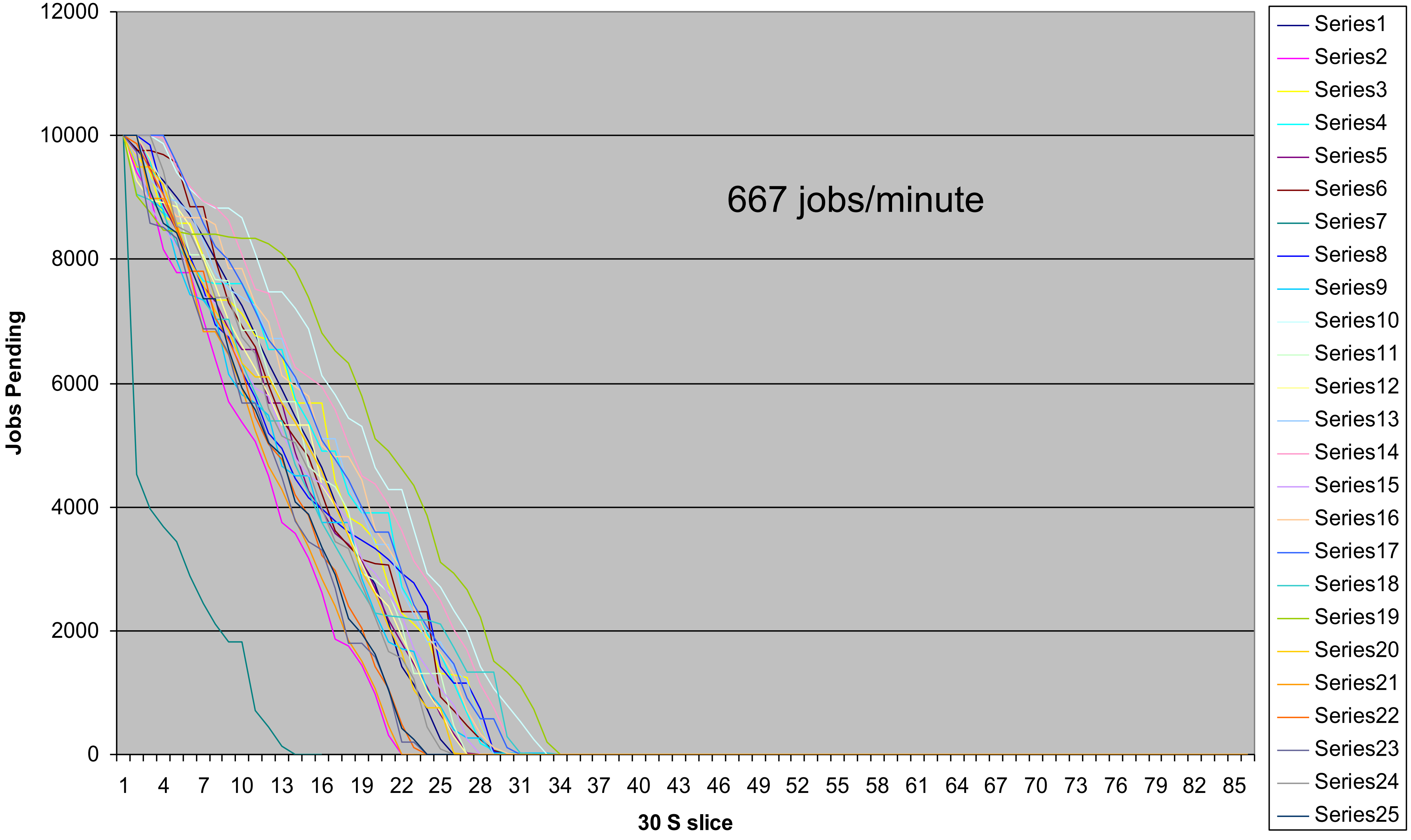
- To test submission rates, the earlier given simple submission is used
- Submit 10000 jobs to both the local Condor pool and RUC Factory
- Start the clock after the “condor_submit” with a “Queue 10000”
- Loop checking when all jobs have completed with “condor_q”
- Jobs are only “/bin/hostname” so they exit almost immediately
- Wall clock time between start and end will be a10K submission rate
- Difference between local and RCC test should show the overhead
- However, its not quite that simple
- Local rate dependent on number of local jobs slots
- Negotiator cycle time (60 seconds) also plays a big role

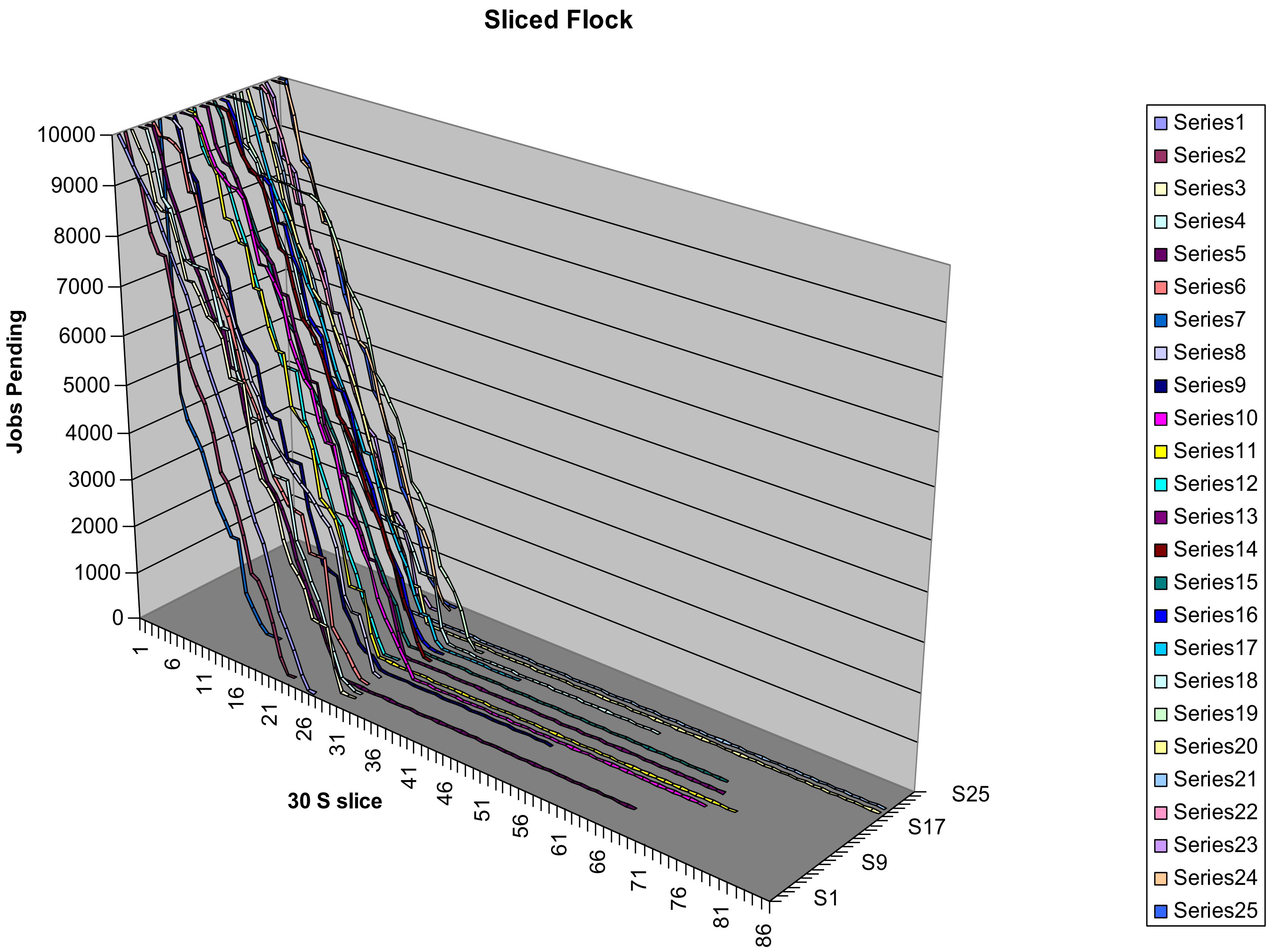
Performance

- Local rates depend on number of job slots and Negotiator rate
  - Used LX Tier3 cluster at Illinois
  - 62 empty job slots
  - Default Negotiator cycle (60 seconds)
  - All slots empty
  - Use 10K submissions to remove bias of a small sample
  - Value under 60 can happen within seconds to just over 60
- Remote test dependent on how quickly slots become available
- Ran 25 tests
- 30 second samples


Sliced Local
83 jobs/minute
Jobs Pending
12000
10000
8000
6000
4000
2000
0
30 S slice
1
11
21
31
41
51
61
71
81
91
101
111
121
131
141
151
161
171
181
191
201
211
221
231
241
Series1
Series2
Series3
Series4
Series5
Series6
Series7
Series8
Series9
Series10
Series11
Series12
Series13
Series14
Series15
Series16
Series17
Series18
Series19
Series20
Series21
Series22
Series23
Series24
Series25

Sliced Flock




Sliced Flock
Jobs Pending
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
30 S slice
1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
76
81
86
S1
S9
S17
S25
Series1
Series2
Series3
Series4
Series5
Series6
Series7
Series8
Series9
Series10
Series11
Series12
Series13
Series14
Series15
Series16
Series17
Series18
Series19
Series20
Series21
Series22
Series23
Series24
Series25

Glossary

- CONNECT
  - The overall umbrella of this project (also a Panda Queue name)
- RCCF
  - Remote Cluster Connect Facility (Multiple installations of Bosco)
- RCC Factory
  - Bosco instance installed with a unique port and user account
- Vslot
  - Virtual Slot created on a remote node by the RCCF
- ACE
  - Atlas Compliant Environment
- ACE Cache
  - Collection of all the components needed to provide an ACE
- Parrot
  - Component of cctools use to provide CVMFS access in the ACE
- Parrot Cache
  - Parrot and CVMFS caches (on the worker node)
- APF
  - AutoPyFactory – Used to inject ATLAS Pilots into the RCCF