Connecting Campus Infrastructures with HTCondor Services

Lincoln Bryant
Computation and Enrico Fermi Institutes
University of Chicago

Condor Week 2014
Outline

• The need for connective services for cyber infrastructure
• Technical approaches using HTCondor
• Example deployments: ATLAS Connect, Duke CI Connect
• Next steps
Resources are distributed

cloud resources + WAN resources

Distributed storage resources

Centralized campus clusters
(condo and hotel models in campus HPC centers)

Legacy campus clusters

Colleges & divisions
How can we transparently connect them to users?

Approach: work from the campus perspective considering both users and resource providers
Providers: accountable to investors

- The BIG problem:
  - Resource providers must first meet campus investor requirements
    - Sometimes with little effort to worry about connectivity to the national ecosystem
    - They are typically slow to open up resources for sharing
  - But computers depreciate in value really fast!
    - Shouldn’t we obligate ourselves to make the most out of these machines?
  - Can we do the heavy lifting for them?
Bring services directly to campus

Campus Condo Cluster or grid

Virtual Clusters

As value added services for the campus computing center

Tightly coupled ↔ serial high throughput
Minimize environment differences for users
Virtually extend capacity
Connecting Cluster Resources

- There's a large barrier to entry when we aren't even running the same scheduler software.
- The solution: Bosco
  - [http://bosco.opensciencegrid.org/](http://bosco.opensciencegrid.org/)
- Bosco provides an HTCondor interface to PBS, SGE, SLURM, etc., via BLAHP\(^1\)
  - Has end-user and multi-user (service) modes
- We can provide transparent user access to multiple resources using Bosco and direct flocking

\(^1\) See BLAHP presentation at HTCondor Week 2013
Advantages of the Bosco approach

- From the admin perspective, we only need a user account and SSH key on the remote machine.
- Since our Bosco service appears a normal user, it's trivial to apply local policy.
- If jobs get killed, we can handle that.
  - All the better if we can use an opportunistic queue!
- Bosco also lets us have pre-job wrappers that allow us to build a comfortable nest
Bosco as a standard tool for HTC

- In the reverse direction, we want users of remote clusters to be using our resources for spillover.
- This is an especially nice value-added service for HPC environments
  - We don't require allocations. We'll process your pleasingly parallel workloads for free.
- Everyone in HPC-land seems to be using Modules, so we have done some work to provide a Bosco module.
Example use case:
ATLAS Connect Service Components

- **Globus Platform**
  - Reliable file transfer to ‘scratch’ endpoints
  - Identity management
  - Groups, Projects

- **Login host**
  - Auto-provisioning, quickly validating users

- **Bosco-based Glidein Factories**
  - “Remote Cluster Connect Factories” (RCCF)
  - One instance per resource target

- **Gratia accounting & Cycle Server monitoring**

- **FAXbox storage service**
  - POSIX, http, XRootD, Globus access
Three Service Types for ATLAS

- **ATLAS Connect User**
  - HTCondor-based login machine for US ATLAS physicists to reach cycles at dedicated datacenters as well as departmental clusters.

- **ATLAS Connect Cluster**
  - Send jobs from local departmental cluster to ATLAS Connect infrastructure using HTCondor’s flocking mechanism

- **ATLAS Connect Panda**
  - Integration with ATLAS “Panda” job workflow manager.
  - Opportunistically send simulation jobs to clouds, campus clusters, HPC resource centers
ATLAS Connect User

CONNECT user

- connect.usatlas.org
- rccf.usatlas.org (glidein factories)
- login.usatlas.org

Campus Grids
Off-grid Tier3

ATLAS T1 (dev)
Tier2

TACC
Stampede (dev)

Cloud (AWS)

various campus resources

XSEDE cloud

users from 44 institutions

US ATLAS Computing Facility

Open Science Grid
Looks like a very large cluster

- Users want to see quick, immediate “local” batch service
- Most Tier 3 batch use is very spikey
- Use 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

<table>
<thead>
<tr>
<th>Pool</th>
<th>Total Slots</th>
<th>Running</th>
<th>Idle</th>
<th>Owner</th>
<th>Status</th>
<th>Detailed View</th>
<th>Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSU Fresno Factory</td>
<td>248</td>
<td>248</td>
<td>0</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
<tr>
<td>Great Lakes Tier 2 Factory</td>
<td>643</td>
<td>639</td>
<td>4</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
<tr>
<td>Midwest Tier 2 Factory</td>
<td>1992</td>
<td>1992</td>
<td>0</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
<tr>
<td>Southwest Tier 2 Factory</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
<tr>
<td>TACC Stampede</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
<tr>
<td>UC3</td>
<td>528</td>
<td>528</td>
<td>0</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
<tr>
<td>UChicago RCC Factory</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>3411</strong></td>
<td><strong>3407</strong></td>
<td>4</td>
<td>0</td>
<td>Usage</td>
<td>Jobs</td>
<td></td>
</tr>
</tbody>
</table>

- Jobs by State
- Jobs by Owner
- Slots by State
- Slots by Owner
Connect is very quick relative to grid.

Throughput: 10000 5 min jobs in 90 minutes.

Site distribution:
- mwt2.org: 14223 (42.24%)
- csufresno.edu: 14185 (41.86%)
- aglt2.org: 1514 (15.14%)
- atlas-swt2.org: 75 (0.75%)

Legend:
- Used by owner
- vpa
- rwg
- Unclaimed
ATLAS Connect Cluster

- ATLAS physicists on their institution’s “Tier 3” cluster can submit into ATLAS Connect without ever leaving home.
  - Submissions can be overflow or targeted using HTCondor class ads.
- Admins configure HTCondor to flock to the Remote Cluster Connect Factories (RCCF)
  - Configuration happens on the local HTCondor schedd
  - Firewall rules etc. opened as necessary.
- The RCCF service can reach any of the targets in the ATLAS Connect system
  - Easily reconfigured for periods of high demand
CONNECT cluster

- Tier 1
- Tier 2s
- Cloud
- Campus Grids

- RCC Factory: rccf.usatlas.org
- Local Tier3 Centers
- FAXbox: connect.usatlas.org

xrd.globus
ATLAS Connect Cluster in use

- Five clusters configured in this way so far
- Works well, very low maintenance
Early adopters ramping up
Extending ATLAS Connect to more resources

• Some of our colleagues have access to “off-grid” resources such as supercomputers and campus clusters.

• We can’t expect any of these resources to have our software prerequisites.

• By combining HTCondor and Parrot\[2\], we can run ATLAS jobs on these kinds of resources.

• Parrot allows us to:
  – Access our ATLAS software repositories
  – Play some LD_PRELOAD tricks to access system dependencies that we need

\[2\] see the Parrot homepage
ATLAS Connect Panda

- We’ve been able to integrate ATLAS Connect with Panda (ATLAS grid workflow manager)
  - ATLAS production (simulation) jobs are fairly well understood in terms of requirements.
  - A new opportunistic queue is created in the workflow system and pointed at HPC resources
  - Jobs come through AutoPyFactory\(^3\), and get locally submitted as HTCondor jobs
  - Pre-job wrappers use Parrot to set up an environment that looks like an ATLAS worker node for the jobs.

\(^3\) AutoPyFactory paper
CONNECt panDa  

Optional user inputs

XSEDE cluster

Tier1

Tier2s

XSEDE cluster

Off-grid

Campus Grids

Tier2 USERDISK
LOCALGROUPDISK

RCC Factory
rccf.usatlas.org  
+ autopyfactory

Faxbox
faxbox.usatlas.org

Local Tier3 Center

Tier2 USERDISK
LOCALGROUPDISK

connect.usatlas.org

xrd  
http  
globus

dq2

prun

ANALY
CONNECT
We’re also providing value-added services for existing campus clusters (ci-connect.net)

One of our early adopters: Duke University

- Login node and scratch storage provisioned for Duke users
- Integrated into Duke “BlueDevil Grid” campus grid.
- Also provides a submit interface into the Open Science Grid
- Bridged to opportunistic resources at the University of Chicago
Duke Grid → the OSG

(No significant effort needed by Duke SCSC staff)
Cl Connect Services in Preparation
Where it started: a login service for OSG
What does the next-generation look like?

- Why can't identity behave more like eduroam?
  - I want my login account to be my campus identity. Period. No new accounts.

- Can we take a hard look at solving some of the problems with software distribution?
  - Are virtual machines the way forward? What about containers?
  - We can play games with static compilation and LD_PRELOAD, but it sure would be nice to have something that looks like your home environment!

- Data access is still not dead simple
  - Focus on data delivery, not data management
Thank you!
Acknowledgements

• Dave Lesny – UIUC (MWT2)
• David Champion – UChicago (MWT2)
• Steve Tuecke, Rachana Ananthakrishnan – (UChicago Globus)
• Ilija Vukotic (UChicago ATLAS)
• Suchandra Thapa (UChicago OSG)
• Peter Onysis – UTexas
• Jim Basney (CI-Logon) & InCommon Federation
• & of course the HTCondor and OSG teams

osg connect