Stampede and Panda

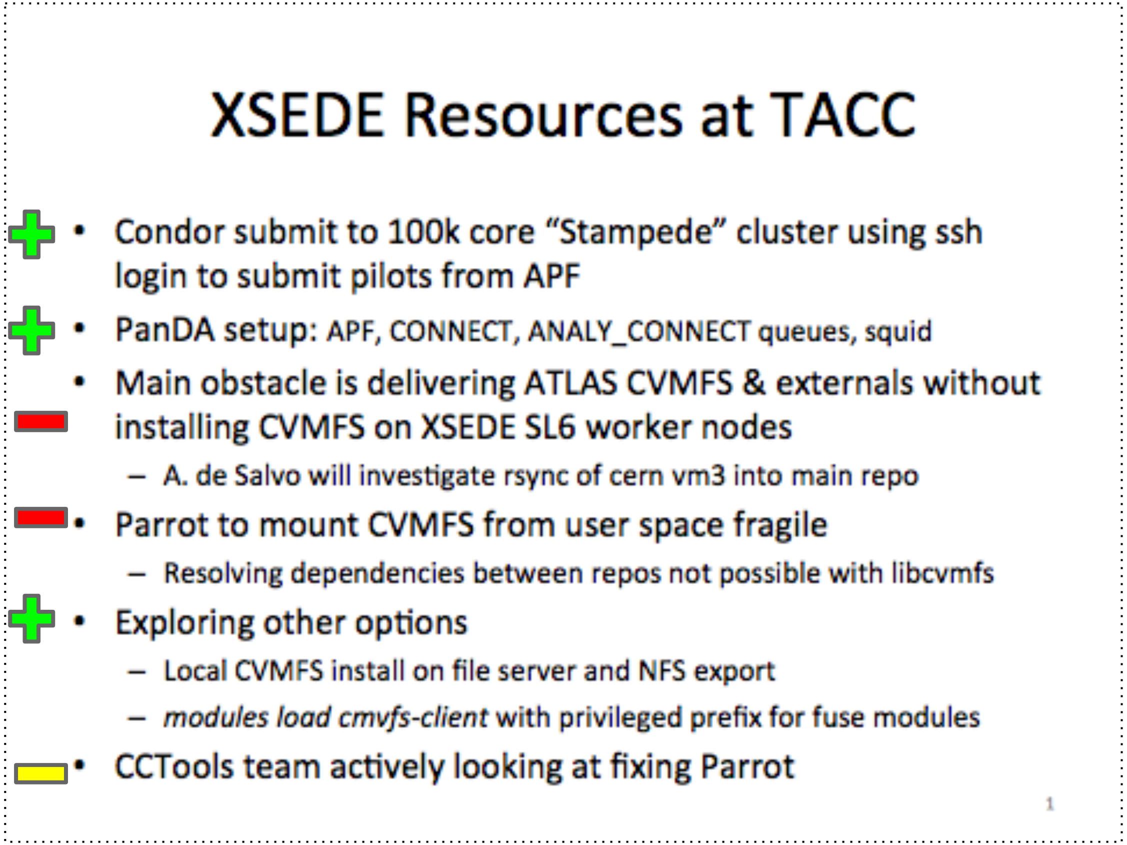
David Lesny, Lincoln Bryant, Rob Gardner,
Peter Onyisi

8/21/14

Setup

From last time:

Software access (CVMFS) remains the stumbling block



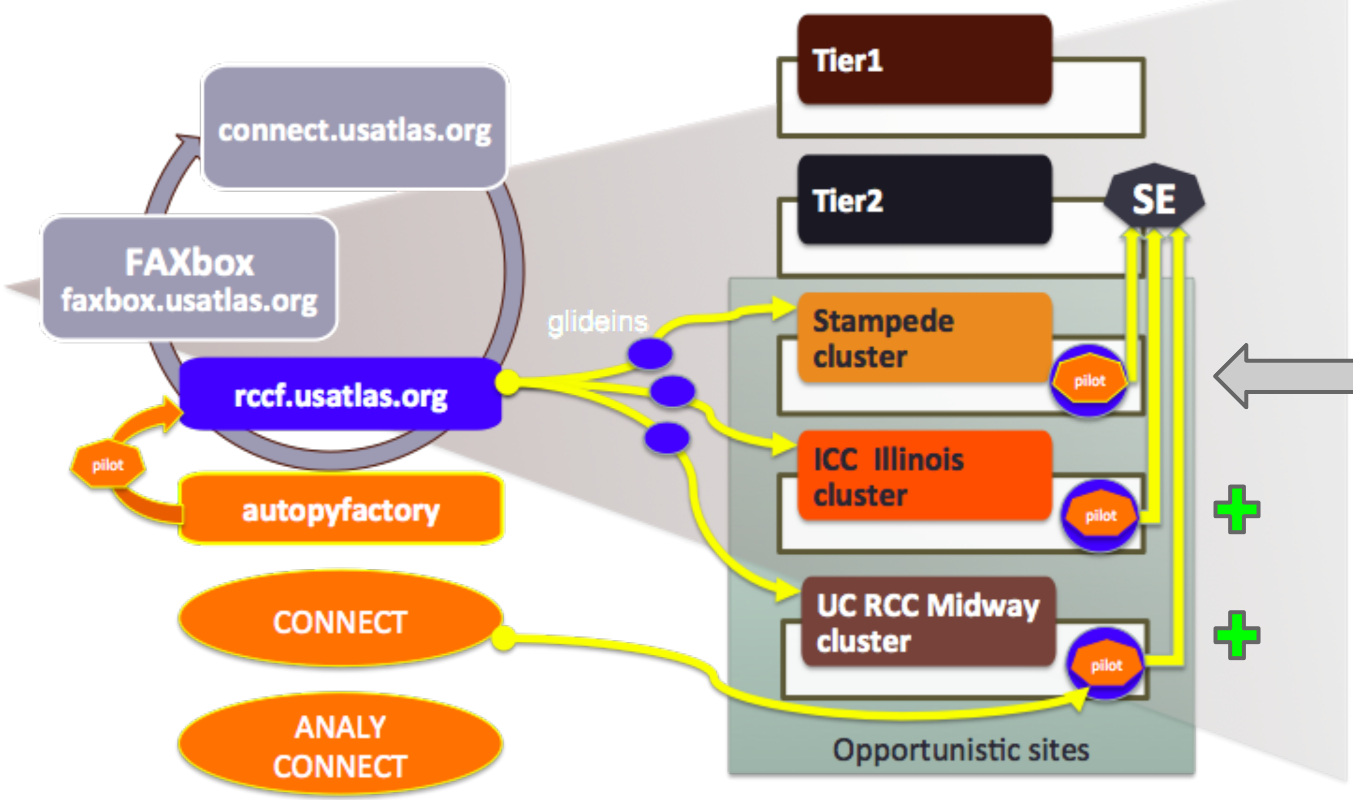
XSEDE Resources at TACC

- ➕ Condor submit to 100k core “Stampede” cluster using ssh login to submit pilots from APF
- ➕ PanDA setup: APF, CONNECT, ANALY_CONNECT queues, squid
- 🟥 Main obstacle is delivering ATLAS CVMFS & externals without installing CVMFS on XSEDE SL6 worker nodes
  - A. de Salvo will investigate rsync of cern vm3 into main repo
- 🟥 Parrot to mount CVMFS from user space fragile
  - Resolving dependencies between repos not possible with libcvmfs
- ➕ Exploring other options
  - Local CVMFS install on file server and NFS export
  - *modules load cmvfs-client* with privileged prefix for fuse modules
- 🟨 CCTools team actively looking at fixing Parrot

1

CONNECT panda

Glidein factory, APF, queues all setup and operational.



For Stampede, the issue is CVMFS

Six ways to CVMFS

- **NativeCVMFS:** Install CVMFS on every node
  a. RPMS are installed on every node by site administrators (standard for a WLCG site)
  b. Best performance; also requires compatibility libraries over base SL6.x
  c. Needs some local disk for the cache
  d. Configure for ATLAS, OASIS and MWT2 repositories

the standard

- **ParrotCVMFS:** I/O trap and redirect to a CVMFS Alien Cache
  - Emulates a NativeCVMFS installation
  - No changes required by remote site administrators
  - Performance hit: 30% and up depending on application
  - Still problems generically running all Atlas code
- **nfsCVMFS**: Access CVMFS repositories via an NFS server
  a. Good performance
  b. Only 1 mount on each worker node
  c. No need for local disk
  d. Unknown scalability (network and NFS server load)

Deployed and operational on UC RCC Midway

Six ways to CVMFS (continued)

- **PortableCVMFS:** User job mounts all repositories
  1. Bring a CVMFS client with the job
  2. Need to install FUSE and fuse kernel module
  3. Needs some local disk for the cache
  4. Can use a mount root other than “/cvmfs”, but not supported by Atlas as yet
  5. Same performance as NativeCVMFS

Deployed and operational on Illinois Campus Cluster (ICC)

- **ReplicaCVMFS:** Replicate all repositories to a local Linux file system
  1. Dump all repositories to a local disk via “rsync”
  2. Slow process to unpack, replicate and keep updated on a local disk
  3. Can speed up the process if using a local Stratum-1 and DIRECT (no proxy)
  4. Need a fair amount of disk for the S1 repositories replicas and “rsync” replica
  5. Rsync target need to be a common file system so all worker node have access
- **Dependency bundling**
  1. Use tools to gather dependencies and place into a package, for execution on remote sites: auditing step.
  2. CVMFS is only needed on an “auditing” host, not on the compute node

ParrotCVMFS (tested & paused)

Parrot can be used to provide access to CVMFS repositories without any changes to the system

Brought with the job as part of the wrapper

- Parrot traps all I/O calls with PTRACE and redirects them to libcvmfs if accessing “/cvmfs”
- Parrot versions 4.1.3 and 4.1.4rc5 worked on some sites, would cause hangs on others
  - ⇒ Sensitive to the kernel version
- Current release 4.2 works better
  - Still have problems running any “java” code

nfsCVMFS (CVMFS via NFS)

- Build a standard **NFS server** on an EL6 platform (Use RPCNFSDCOUNT=128)
  - Install CVMFS Client 2.1.19 (or later), CVMFS init scripts and CVMFS keys (from CERN). Do NOT setup to use autofs (/etc/auto.cvmfs)
  - Install OASIS and MWT2 repositories (scripts and keys)
  - Configure “default.local” for repositories, squids, cache location/size, etc
  - Configure “default.local” to use NFS

    ```
    CVMFS_NFS_SOURCE=yes
    CVMFS_MEMCACHE_SIZE=256
    CVMFS_MAX_RETRIES=2
    ```

  - Statically mount all repositories (10 total) at “/cvmfs/xxx” via /etc/fstab

    ```
    atlas.cern.ch                /cvmfs/atlas.cern.ch           cvmfs   defaults 0 0
    atlas-condb.cern.ch           /cvmfs/atlas-condb.cern.ch         cvmfs   defaults 0 0
    ```

  - Add all repositories (10 total) to /etc/exports along with the “/cvmfs” in a crossmnt

```
/cvmfs -ro,sync,no_root_squash,no_subtree_check,insecure,fsid=100,crossmnt xx.xx.xx.xx
/cvmfs/atlas.cern.ch -ro,sync,no_root_squash,no_subtree_check,insecure,fsid=101,nohide   xx.xx.xx.xx
/cvmfs/atlas-condb.cern.ch  -ro,sync,no_root_squash,no_subtree_check,insecure,fsid=102,nohide     xx.xx.xx.xx
```

- On **worker node**, only need to mount the “/cvmfs” in /etc/fstab

```
uct2-int.mwt2.org:/cvmfs   /cvmfs    nfs ro,nfsvers=3,noatime,nodiratime,ac,actimeo=60,lookupcache=all 0 0
```

PortableCVMFS

Portable CVMFS is brought with the job to the worker node.

FUSE is used to mount the CVMFS repositories

- On worker node fuse must be installed, module loaded **and user in “fuse” group**

```
     yum install fuse fuse-libs
     modprobe fuse
```

- User can then mount the repositories with

```
     cvmfs2 -o config=${_CVMFS_CONF_atlas_cern_ch}     atlas.cern.ch
${CVMFS_MOUNT} /atlas.cern.ch
```

- Can umount with

```
     fusermount -u ${CVMFS_MOUNT}/atlas.cern.ch
```

ReplicaCVMFS (testing)

- For sites that want to use a project area on a shared filesystem like Lustre or GPFS
- Replicating CVMFS repositories to a Linux file system via “rsync” is not an option
  - Very slow - network latency
  - Generates load on Squid proxies and Stratum-1
- Idea: Build a local Stratum-1 to bypass network, squid and keep overhead local
  - Use “cvmvs_server snapshot” to create a Stratum-1 replication (takes days)
  - Use snapshot to incrementally update
  - Install CVMFS client to use only this Stratum-1 as its source
  - Use “DIRECT” for Squid proxy
  - All I/O restricted to local disk only
  - rsync from “/cvmfs” to local Linux file system
  - Still slow for atlas and atlas-nightlies but can fine tune what to update daily
- Linux File System should be common to all worker nodes
  - Link “/cvmfs” to the location of replicated repositories
  - Jobs can then access all repositories from the local copy

Dependency bundling (testing)

- As part of DASPOS, we are testing two options for gathering dependencies and placing into a package (or Linux container)
- A configuration of Parrot and a tool called PTU are being tested. Both work the same way:
  - An auditing step is performed, and selected libraries are placed in a self-contained package
  - Deliver the package with the job for execution on sites without CVMFS
- Tested with derivation transform

Status

TACC admins have agreed to try the portableCVMFS on a test node

Meanwhile we explore the ReplicateCVMFS option