Introduction to SLAC Analysis Computing Facility

Shawfeng Dong, Yeeting Li, Wei Yang
Introduction and resources

- SLAC Unix account, as part of the benefit of becoming a SLAC Lab user.
  - SLAC is a federal government (DOE) facility, so
  - It can take days or even weeks to become a SLAC Laboratory user, so
  - act/apply as early as possible.

- Interactive Login and Remote X-window

- GPFS home directory and private data space
  - 100GB/user home, 2TB/user private data, up to 10TB/user

- RUCIO storage (for ATLAS data products)

- Batch (LSF)

- New things: JupyterLab/PyRoot, etc. Xcache to access data
  - Think of them as β test stage
Contents of next 10+ slides

After this, there will be tutorials on how to get started at BNL and SLAC Analysis Facility (aka. Shared Tier 3)

As you work with the tutorial materials, details in the next 10+ slides will serve as references to the SLAC environment

● We will quickly fly over those detail slides

For SLAC specific questions, Email to

● SLAC AF user mailing group: atlas-us-slac-acf@cern.ch
● Unix related questions: unix-admin@slac.stanford.edu
  ○ SLAC personnels are in both lists
Where to start

Become a SLAC Lab User **FIRST**, then apply a SLAC unix account

- [https://atlas.slac.stanford.edu/atlas-support-center](https://atlas.slac.stanford.edu/atlas-support-center)

**2019 SLAC Summer Institute**

**See past related SLAC events »**

---

**USER INFORMATION**

- User Registration
- Computer Account
- Using the SLAC computing resources

---

SLAC hosts a U.S. ATLAS Center for the community.

- We organize jamborees on physics analysis, analysis
- We provide hands-on experimental opportunities for

---

**Detailed computing documents**

- [https://confluence.slac.stanford.edu/display/Atlas/Computing](https://confluence.slac.stanford.edu/display/Atlas/Computing)
Interactive Login, GPFS space

```bash
ssh -Y <user>@rhel6-64.slac.stanford.edu or
ssh -Y <user>@centos7.slac.stanford.edu — preferred, ATLAS moves to CentOS 7
```

```
$ cd /gpfs/slac/atlas/fs1/u/yangw ← user “yangw” home directory
$ df -h .
    Filesystem  Size  Used  Avail  Use%  Mounted on
remote-atlas-fs1 100G  219M  100G    1%  /gpfs/slac/atlas/fs1
```

```
$ cd /gpfs/slac/atlas/fs1/d/bdouglas ← user “bdouglas” private data space
$ df -h .
    Filesystem  Size  Used  Avail  Use%  Mounted on
remote-atlas-fs1 5.0T   0  5.0T    0%  /gpfs/slac/atlas/fs1
```

In GPFS (only), this is disk quota
CVMFS, ATLAS environment

CVMFS is available on all interactive login nodes and batch nodes

- Example of a very minimum ATLAS environment setup via CVMFS:
  
  ```
  export RUCIO_ACCOUNT="yangw"
  export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
  source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh
  localSetupRucioClients # optional, setup RUCIO client tools
  localSetupPandaClient   # optional, setup Panda client tools
  ...
  ```

All nodes also have /scratch for you to store temporary data

- Create and use your own directory under /scratch!
- Clean your debris when you are done
- SLAC will periodically delete files under /scratch that are 7+ day old
IBM Platform LSF batch system

In LSF, your batch jobs inherit all of your current working environment:

- Working directory
- Unix environment variables
- But not Shell aliases

Trick on X509 proxy:

- Before submitting a job, put a X509 proxy under your home directory
  
  ```
  export X509_USER_PROXY=$HOME/.globus/x509up_u$(id -u)
  voms-proxy-init -voms atlas
  ```

- This way all of your batch jobs can access your X509 proxy
  
  ○ Note that your job may be pending in the batch queue for a while. So don’t forget to renew your X509 proxy

This is different from the Condor batch system
Your batch jobs

Submit a job

$ bsub -q atl-analq -R 'select[centos7]' < myjob.sh
Job <59535> is submitted to queue <atl-analq>.

Check the job

$ bjobs 59535

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>STAT</th>
<th>QUEUE</th>
<th>FROM_HOST</th>
<th>EXEC_HOST</th>
<th>JOB_NAME</th>
<th>SUBMIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>59535</td>
<td>yangw</td>
<td>RUN</td>
<td>atl-analq</td>
<td>cent7d</td>
<td>kiso0057</td>
<td>myjob.sh</td>
<td>Jul 31 23:49</td>
</tr>
</tbody>
</table>

$ bjobs 59535

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>STAT</th>
<th>QUEUE</th>
<th>FROM_HOST</th>
<th>EXEC_HOST</th>
<th>JOB_NAME</th>
<th>SUBMIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>59535</td>
<td>yangw</td>
<td>DONE</td>
<td>atl-analq</td>
<td>cent7d</td>
<td>kiso0057</td>
<td>myjob.sh</td>
<td>Jul 31 23:49</td>
</tr>
</tbody>
</table>

Kill the job

$ bkill 59535
Job <59535>: Job has already finished

Detail and more complex examples: https://confluence.slac.stanford.edu/display/Atlas/Run+batch+jobs
Submit a Panda/Grid Analysis Job

Panda can run many type of jobs on the Grid ← your entrance point of docs

- “prun” is one of them: **ROOT/general jobs** (Run “prun --help”)
  - General: ROOT (CINT, C++, PyROOT) jobs, python jobs or **AthenaROOTAccess jobs (ARA)**

**Very simple example ARA jobs** (from the above prun URL)

- Test on local machine: python aratest.py - AOD1.pool.root,AOD2.pool.root
- Submit to Panda: prun --exec "python aratest.py - %IN" --useAthenaPackages \\
  --inDS valid1.006384.PythiaH120gamgam.recon.AOD.e322_s412_r577/ \\
  --outDS user.me.test01 --outputs out.root \\
  Your input dataset. Panda will submit multiple Grid jobs to process all input files \\
  in the input dataset in parallel, each job creates an output file (below) \\
  --destSE SWT2_CPB_LOCALGROUPDISK \\
  Send all your entire output dataset to a US LOCALGROUPDISK, such as this one

Monitor the status of your Panda jobs: [https://bigpanda.cern.ch/dash/analysis/#cloud_US](https://bigpanda.cern.ch/dash/analysis/#cloud_US)
Playing with RUCIO

List files in a dataset:                           RUCIO Data IDentifier: **scope : file**

```
$ rucio list-files data18_13TeV:DAOD_EXOT12.14278917._000001.pool.root.1
+-----------------+----------------+-----------------+-----------------+-----------------+
| SCOPE:NAME       | GUID            | ADLER32         | FILESIZE        | EVENTS          |
+-----------------+----------------+-----------------+-----------------+-----------------+
| data18_13TeV     | F1ED3FDA-79BA-6F4E-9DAA-36151F48BD63 | ad:1bc9a126 | 17.338 MB | 363 |
+-----------------+----------------+-----------------+-----------------+-----------------+
```

Total files: 83
Total size: 9.593 GB
Total events: 274085

```
$ rucio list-file-replicas --protocol=root --PFN data18_13TeV:DAOD_EXOT12.14278917._000001.pool.root.1
```

- There are multiple copies, including two US sites: **SWT2_CPB** and **BNL**
- Note: Some of these functions are integrated into the **pnfs_ls.py** (See Shuwei’s tutorial)
RUCIO storage at SLAC

Names: SLACXRD_{DATADISK, SCRATCHDISK, LOCALGROUPDISK}

- All T1s and T2s, as well as SLAC & BNL T3/AF have them, in different prefix
  - No direct write access to RUCIO disks
- Please use R2D2 to move ATLAS data product in and out
- Or use “rucio upload” to upload your own dataset
- Or --destSE in your Panda job

To access data in RUCIO storage from SLAC interactive/batch nodes:

- Use RUCIO command to find the location:
  - root://griddev03.slac.stanford.edu:2094/xrootd/atlas/...
- If the file is at SLAC, replace the host name (above) in the URL
  - By root://atlrdr1/xrootd/atlas/...
  - Access directly using TFile *f = TFile::Open("root://atlrdr1/...")
  - copy to scratch xrdcp root://atlrdr1/.../file.root /scratch/me/file.root
Access Remote data: using Xcache

This is experimental at both SLAC and BNL

- We don’t have permanent machines to run Xcache yet
- Subject to changes

Use one of the following URLs (change the obvious mistake here of course)

   - Only work for RUCIO managed files
   - You don’t need to know where is the data source
   - Data will likely coming from SWT2_CPB because they are close to SLAC

   - Work for both RUCIO managed file and private files.
   - You do need to know where is the data source

Current Xcache hosts:
- `atlfax.slac.stanford.edu`
- `xrootd03.usatlas.bnl.gov`

Telling Xcache that this path is a global Logical Filename (gLFN)
Remote Xwindow Display

If you are close to SLAC

- `ssh -Y centos7.slac.stanford.edu /usr/bin/xeyes`

If you are far away from SLAC, use FastX to speed up remote Xwindow display


If you are roaming

- Try JupyterLab (next)
- JupyterLab also has lots of modern data science tools
JupyterLab

Experimental

- We are ready to provide these service though you will be our first real users

JupyterLab runs on precious, *shared*, powerful HW w/ GPUs

- Terminal in JupyterLab is convenient, but not a replacement of login nodes
- Don’t waste the resource for tasks that can be done elsewhere, such as “cp”
Logged in using my SLAC unix account

A list of ATLAS images can be added at here
- All are Docker images
- Better to run on dedicated ATLAS hardware

Start your Jupyter container
JupyterLab with PyROOT

PyROOT can access data via multiple protocols
- Local file (file:// protocol)
- root:// protocol (example above)
- http:// protocol
PROOF and PROOF Lite are powerful parallel data processing tools.

- Need to guide users to get started
- Learning curve is quite flat / easy
JupyterLab as interactive login node

Terminal window:
- Access to users unix home directory
- CVMFS
- Submit batch jobs
JupyterLab Integration with Google Drive

It is also possible to access BNL Box (ownCloud) from JupyterLab