SLAC NATIONAL ACCELERATOR LABORATORY

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

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

Interactive Login, GPFS space

```
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

This is different from the Condor batch system

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

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
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
59535   yangw   RUN   atl-analq  cent7d      kiso0057    myjob.sh   Jul 31 23:49
$ bjobs 59535
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
59535   yangw   DONE  atl-analq  cent7d      kiso0057    myjob.sh   Jul 31 23:49
```

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

SLAC NATIONAL ACCELERATOR LABORATORY

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/ \
```

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)

```
     --outDS user.me.test01 --outputs out.root \
```

Your output dataset: contains all output files user.me.test01._00001.out.root, etc.

```
     --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

Playing with RUCIO

List files in a dataset:

RUCIO Data IDentifier: scope : file

```
$ rucio list-files data18_13TeV:data18_13TeV.00348885.physics_Main.deriv.DAOD_EXOT12.f937_m1972_p3553_tid14278917_00
+-------------------------------------------------------+--------------------------------------+-------------+-----------+--------+
| SCOPE:NAME                                            | GUID                                 | ADLER32     | FILESIZE  | EVENTS |
|-------------------------------------------------------+--------------------------------------+-------------+-----------+--------|
| data18_13TeV:DAOD_EXOT12.14278917._000001.pool.root.1 | 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
root://eosatlas.cern.ch:1094//eos/atlas/atlasdatadisk/rucio/data18_13TeV/da/ea/DAOD_EXOT12.14278917._000001.pool.root.1
root://lpnse1.in2p3.fr:1094//dpm/in2p3.fr/home/atlas/atlasdatadisk/rucio/data18_13TeV/da/ea/DAOD_EXOT12.14278917._000001.pool.root.1
root://gk06.atlas-swt2.org:1094//xrd/atlasscratchdisk/rucio/data18_13TeV/da/ea/DAOD_EXOT12.14278917._000001.pool.root.1
root://dcgftp.usatlas.bnl.gov:1094//pnfs/usatlas.bnl.gov/LOCALGROUPDISK/rucio/data18_13TeV/da/ea/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

Current Xcache hosts:

- **atlfax.slac.stanford.edu**
- **xrootd03.usatlas.bnl.gov**

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

1. `root://atlfax.slac.stanford.edu//atlas/rucio/data18_13TeV:DAOD_EXOT12.14278917._000001.pool.root.1`
   - 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
2. `root://atlfax.slac.stanford.edu//root://dcgftp.usatlas.bnl.gov:1094//pnfs/usatlas.bnl.gov/LOCALGROUPDISK/rucio/data18_13TeV/da/ea/DAOD_EXOT12.14278917._000001.pool.root.1`
   - Work for both RUCIO managed file and private files.
   - You do need to know where is the data source

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

- https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=205985167

If you are roaming

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

JupyterLab

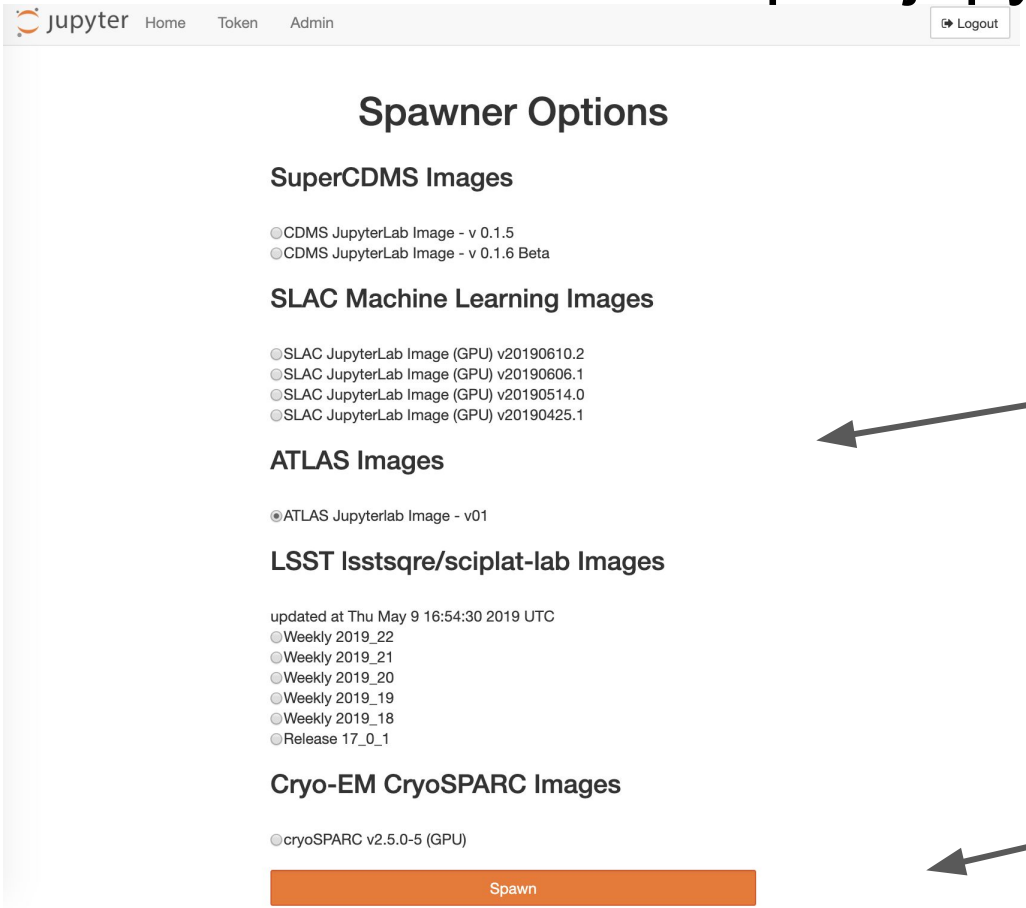
Experimental

- 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”

Entrance Screen https://jupyter.slac.stanford.edu



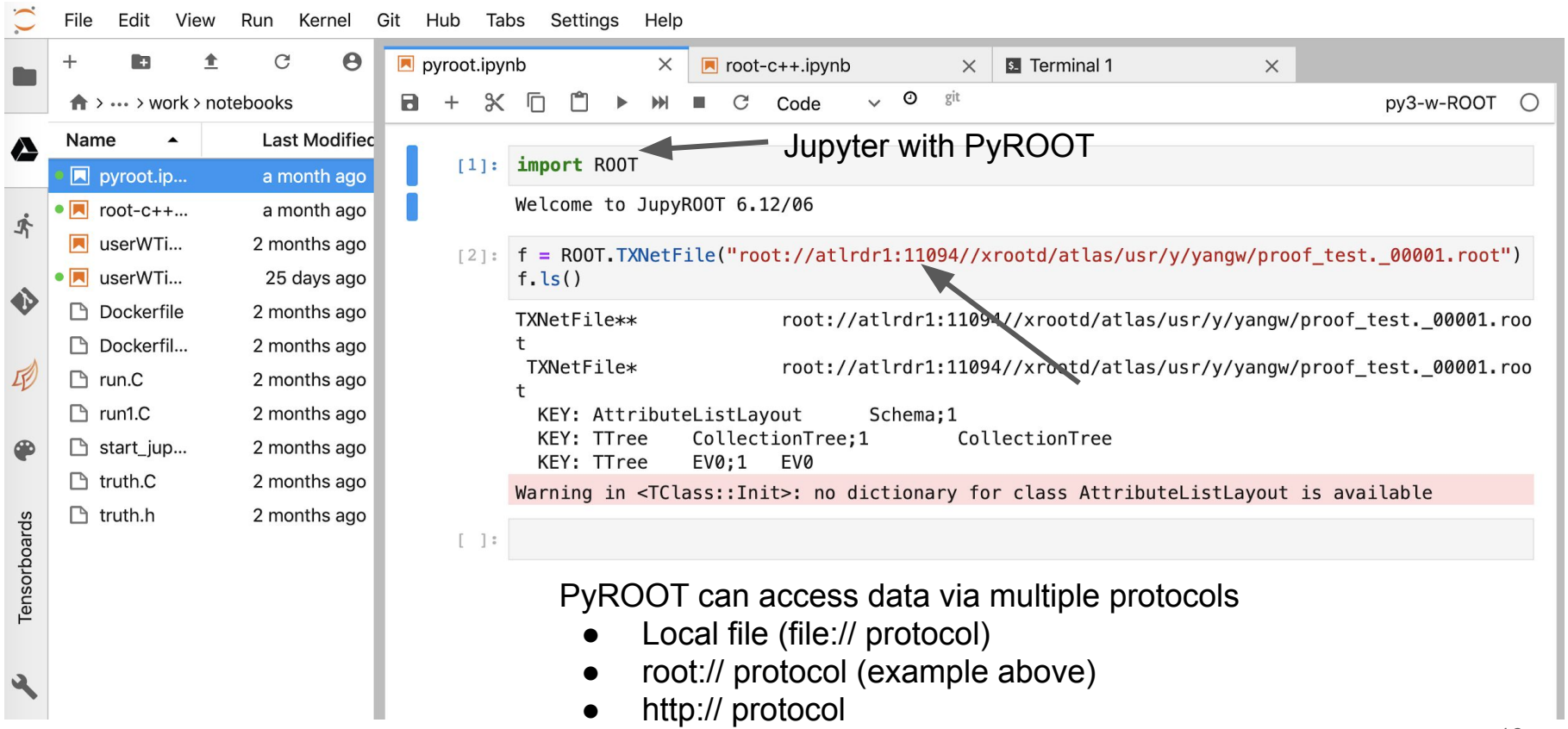
Logged in using my SLAC unix account

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

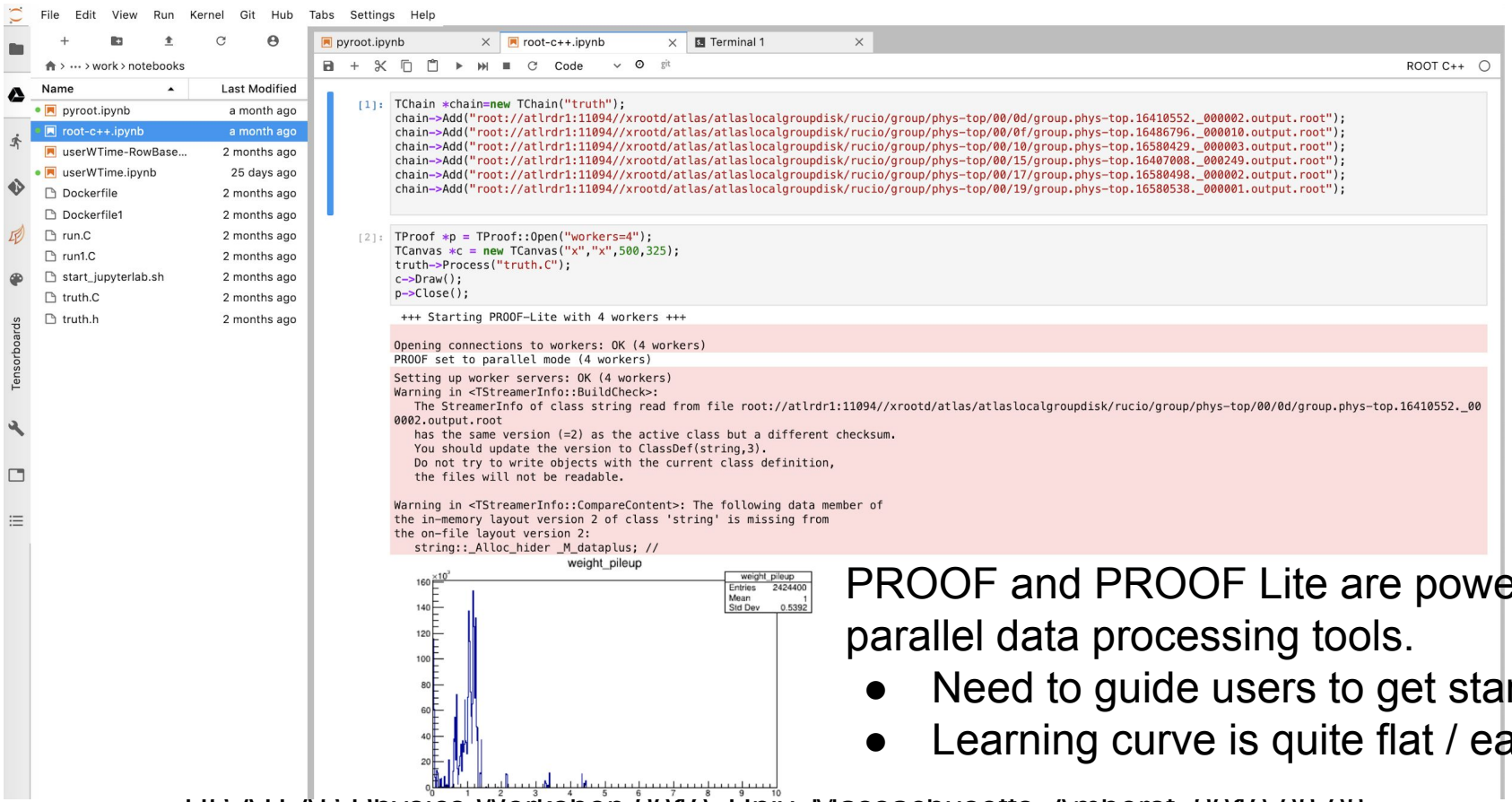
JupyterLab with PyROOT



PyROOT can access data via multiple protocols

- Local file (file:// protocol)
- root:// protocol (example above)
- http:// protocol

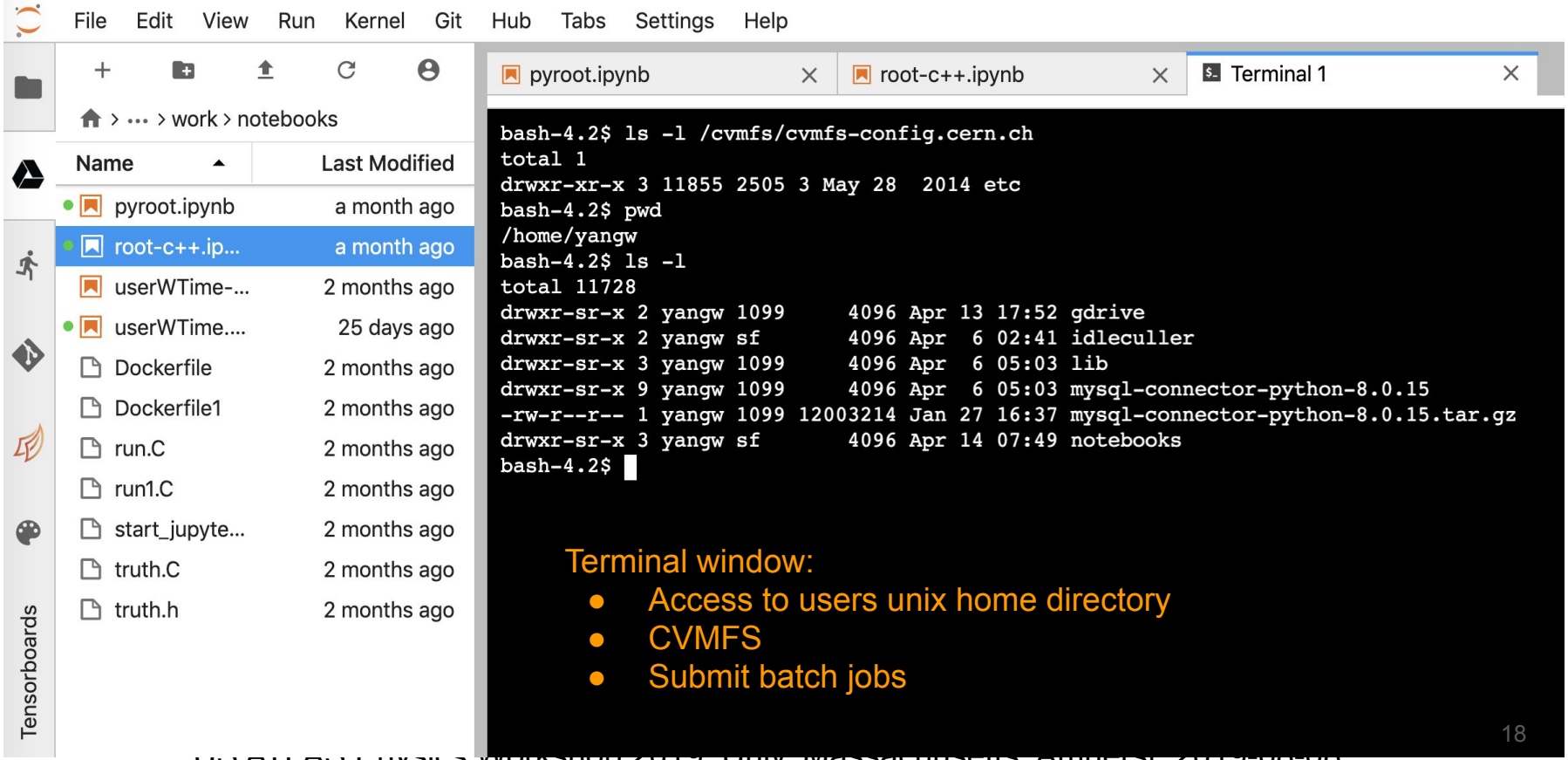
ROOT C++ notebook with PROOF Lite



PROOF and PROOF Lite are powerful parallel data processing tools.

- Need to guide users to get started
- Learning curve is quite flat / easy.

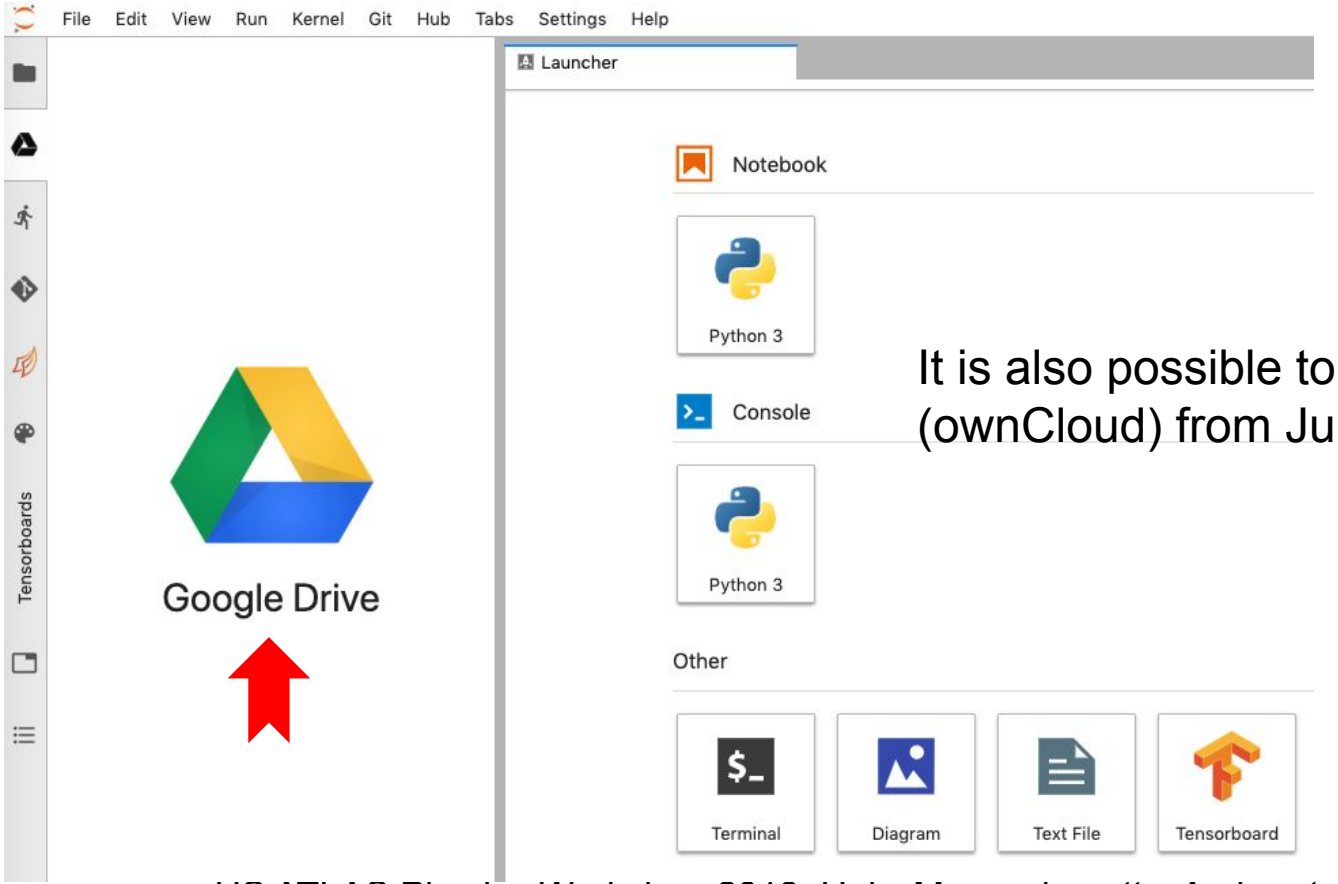
JupyterLab as interactive login node



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