Tier3 Setup at Univ. Wisconsin-Madison
- With PROOF, PQ2, Condor

Neng Xu, Wen Guan, German Montoya, Sau Lan Wu

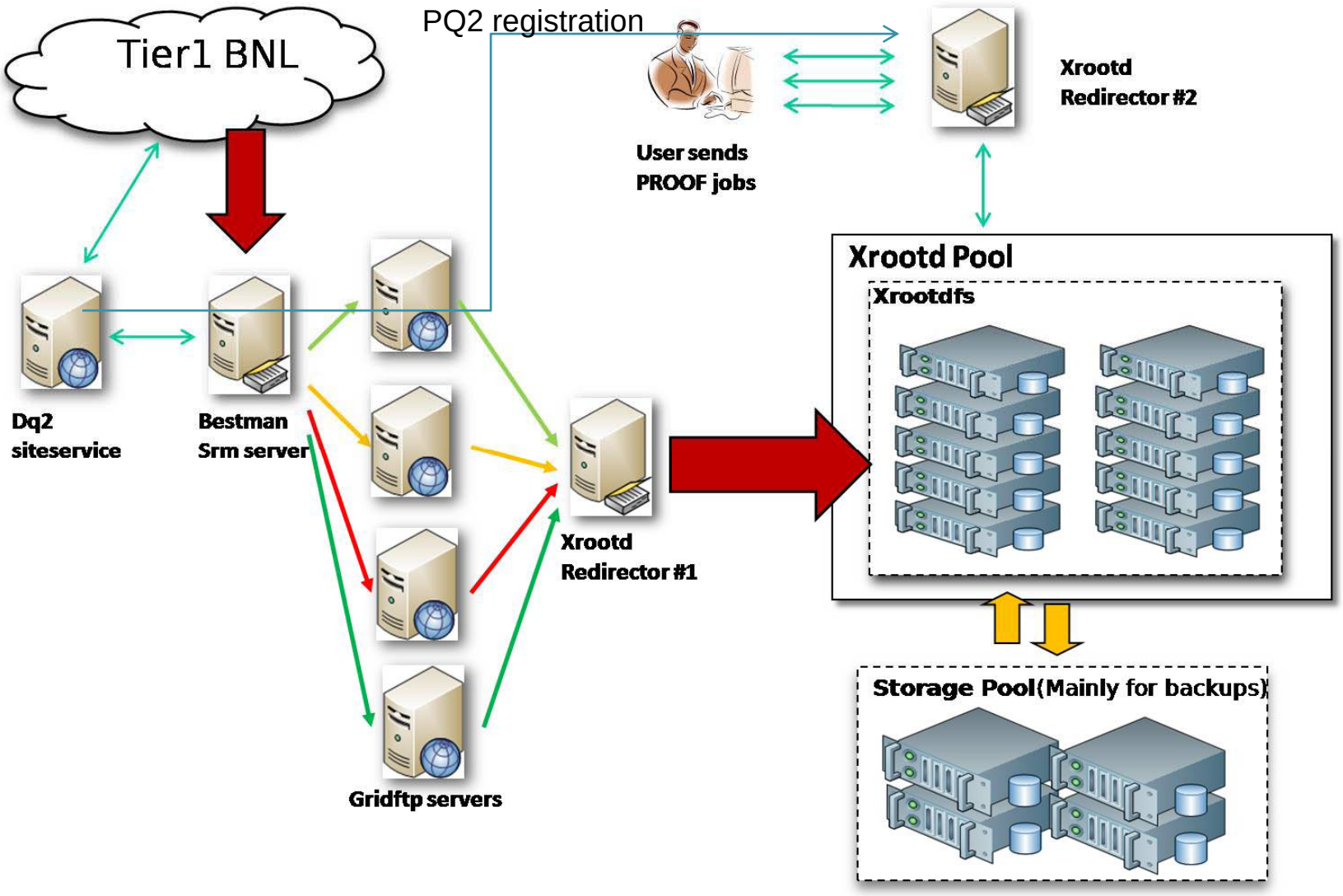
Outline

- Simple introduction of our Tier3 Facility.
- Data management with PQ2 tools.
- Our experience with PROOF
- Our experience with Condor-dagman.
- The use of ““opportunistic” CPU resources. (Run ToyMC and Analysis with Windows machine or other OS cluster.)

Simple introduction of our Tier3 Facility

- We are using BestmanSrm + Xrootd for data transfer and storage.
- Our hardware is 50x(8core+16GB+8x750GB).
- Our network is 10Gb uplink and 1Gb to each node.
- We are using PATHENA, PROOF and Condor- dagman for data analysis.
- We run Xrootd and PROOF separately but on same storage pool.
- We were using MySQL for local data management but now we are using PQ2 tools.

Our system design



Introduction of PQ2 tools

- It's a DQ2 like local data management system by using PROOF dataset management function.
- It can be used to manage local users' data and it can also synchronize with LFC and DQ2.
- File information is stored locally on PROOF master. It can easily replicate to different PROOF master.
- Users only deal with datasets for both PROOF and Condor dagman job submission.
- The files are pre-located and pre-validated. This will save lots of overhead for the PROOF sessions, especially for the large datasets.

Basic Commands

- For system admin:
  - pq2-put (Register a dataset.)
  - pq2-verify (File location check and pre-validate.)
  - pq2-info-server (list all the storage nodes' information.)
  - pq2-ls-files-server (list all the files on one of the storage nodes.)
  - pq2-rm (remove datasets)
- For users:
  - pq2-ls (List the datasets.)
  - pq2-ls-files (List the file information of a dataset.)

Where to find PQ2 tools

- PQ2 is part of ROOT now. Check out the newest SVN version.
- It's in $ROOTSYS/etc/proof/util/pq2.
- To setup the environment:
  1. Setup the ROOT.
  2. export PATH= $ROOTSYS/etc/proof/util/pq2:$PATH
  3. export PROOFURL="nengxu@atlas-bkp2.cs.wisc.edu:2093"

What we had before PQ2

- We built our own MySQL database and register all the file information to this database.
- We ran daemons on each Xrootd data node to register new files to the database.
- Users run the scripts to get the file location and information from the database before submitting analysis jobs.

PQ2 vs MySQL

- We don't need to run separate cron daemons on each storage node. We have to always worry if those scripts are running correctly or not.
- Add/remove datasets is much simpler with PQ2.
- We don't need to maintain our own scripts and don't need to maintain a MySQL database. PQ2 is part of PROOF now. Saved 1 FTE.
- It's easy to make load balance for the PROOF master.
- Users can even synchronize the file information to their desktops/laptops.

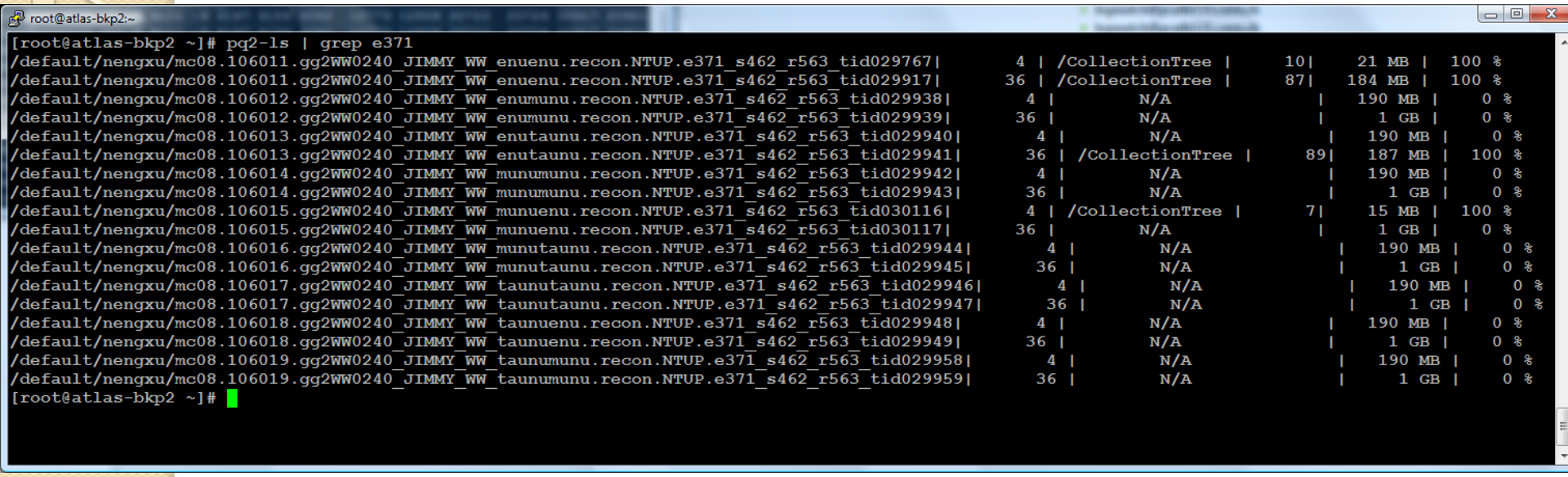
Use PQ2 tools with DQ2/LFC in Wisconsin

- Since we are running DQ2 site service there. we wrote a script which reads out all the file information from DQ2/LFC and converts srm url of PFN to xrootd url.
- For each dataset, we create a simple text file.
- pq2-put command takes those text files as inputs and registers them into PROOF.

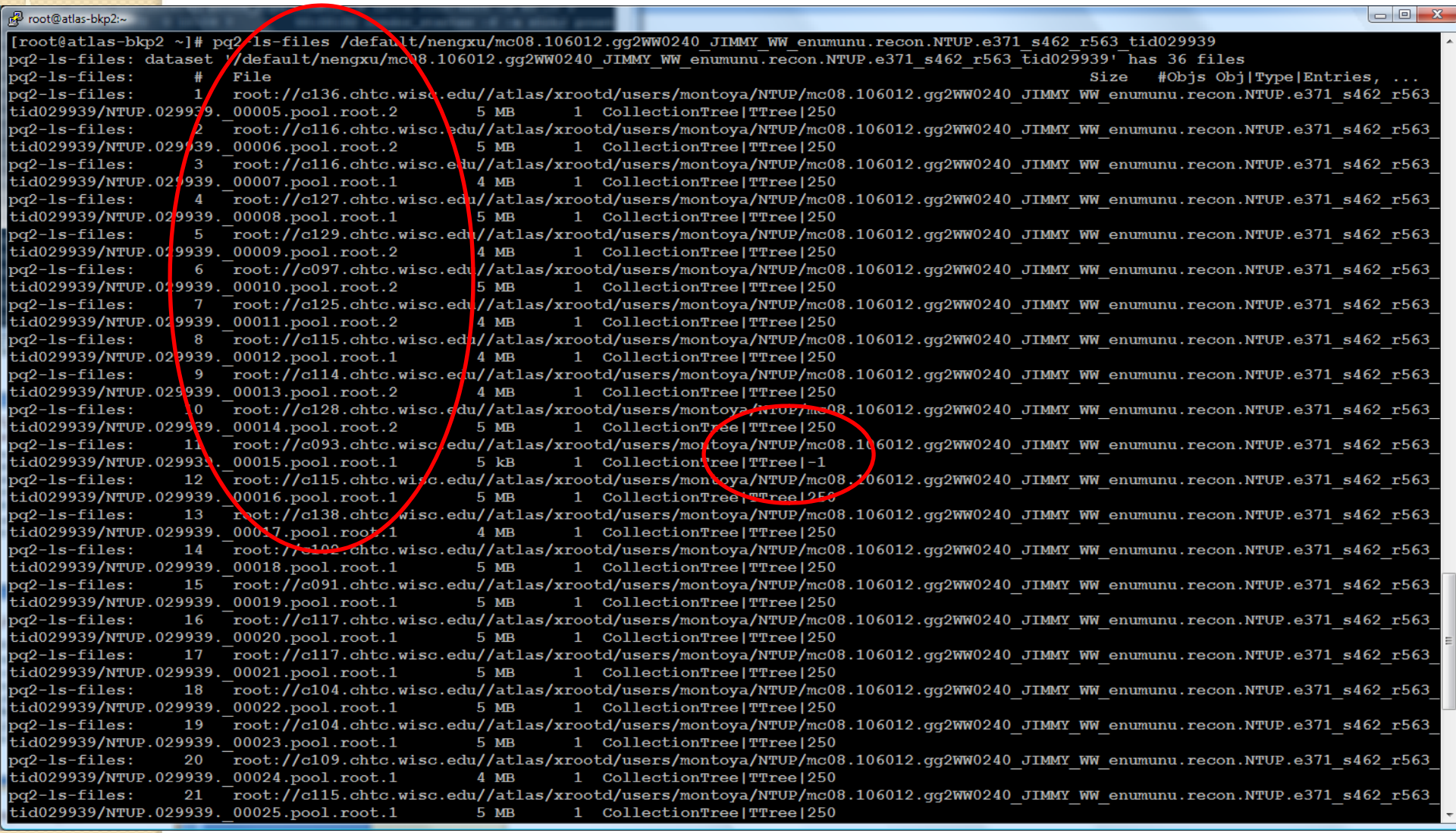
Use PQ2 tools with dq2-get at CERN

- At CERN, we have a small Xrootd/PROOF storage pool with XrootdFS setup.
- We can directly run dq2-get command over the XrootdFS system.
- For the finished datasets, we just simply create a text file with the xrootd url.
- pq2-put takes those text files to register the files into the PROOF.

Example of pq2-ls



Example of pq2-ls-files



How do users use PQ2 tools

- For PROOF users, they just need to put the name of the dataset like this:

```
p-> Process("/default/nengxu/mc08.109067.PythiaH190zz4l.merge.NTUP.e384_s462_r635_t53_tid056686",
                "CollectionTree.C+",options);
```

- For Condor-dagman users, they submit their analysis jobs like this:

```
python submit.py --initdir=/home/wguan/dag_pq2/test3 --parentScript=/home/wguan/dag_pq2/dump.sh --
    childScript=/home/wguan/dag_pq2/merge.sh --dataset=/default/nengxu/mc08.109999.PythiaQCDbb2l.fullsim.v14022303 --
    transferInput=/home/wguan/dag_pq2/input.tgz --jobsPerMachine=2
```

New features in PROOF

- Session queuing system.(Limit number of running sessions and waiting sessions.)
- PROOF-Batch. (Run PROOF as a batch system.)
- PROOF-Lite. (Make use of all your CPU cores on your desktop.)
- PQ2 tools. (New data management system with PROOF.)

Our development with Condor-dagman

- Check the machine status before submit the jobs.
- The way to setup the fast job slots.
- Setup the timeout for the child jobs.
- Limit the number of jobs to each node.
- Change the interface from MySQL to PQ2 tools.

The use of "opportunistic" resources

- Following certain steps, we found the ToyMC and Analysis code(ROOT) can be compiled and run on different operating systems.
- We can run those jobs on many Windows machines (with Condor and Cygwin installed)in the campus.
- This saves lots of CPU time for our limited SLC4 CPU resources.
- No addition software needed on the remote Windows nodes, like compilers. We just need to ship the binary file and some libraries.
- PROOF client can also run on those resources, too.

Summary

- There are many available methods to let users to run analysis jobs. PROOF, PROOF-batch, PROOF-Lite, Condor-dagman, PATHENA.
- PQ2 tools is a handy software package for Tier3s.
- ROOT works fine on many operating systems. Try to run your ToyMC and analysis jobs on Windows machines.
- Wiki pages will be available soon. If you are interested in what we are doing, you can contact me (neng.xu@cern.ch) or ROOT/PROOF team.

Thanks to

- ROOT/PROOF team.
- Condor team.
- DDM team at BNL.
- SLAC Xrootd team.