PROOF on Wisconsin-GLOW

W. Guan, M. Livny, B. Mellado, Sau Lan Wu, Neng Xu
University of Wisconsin-Madison

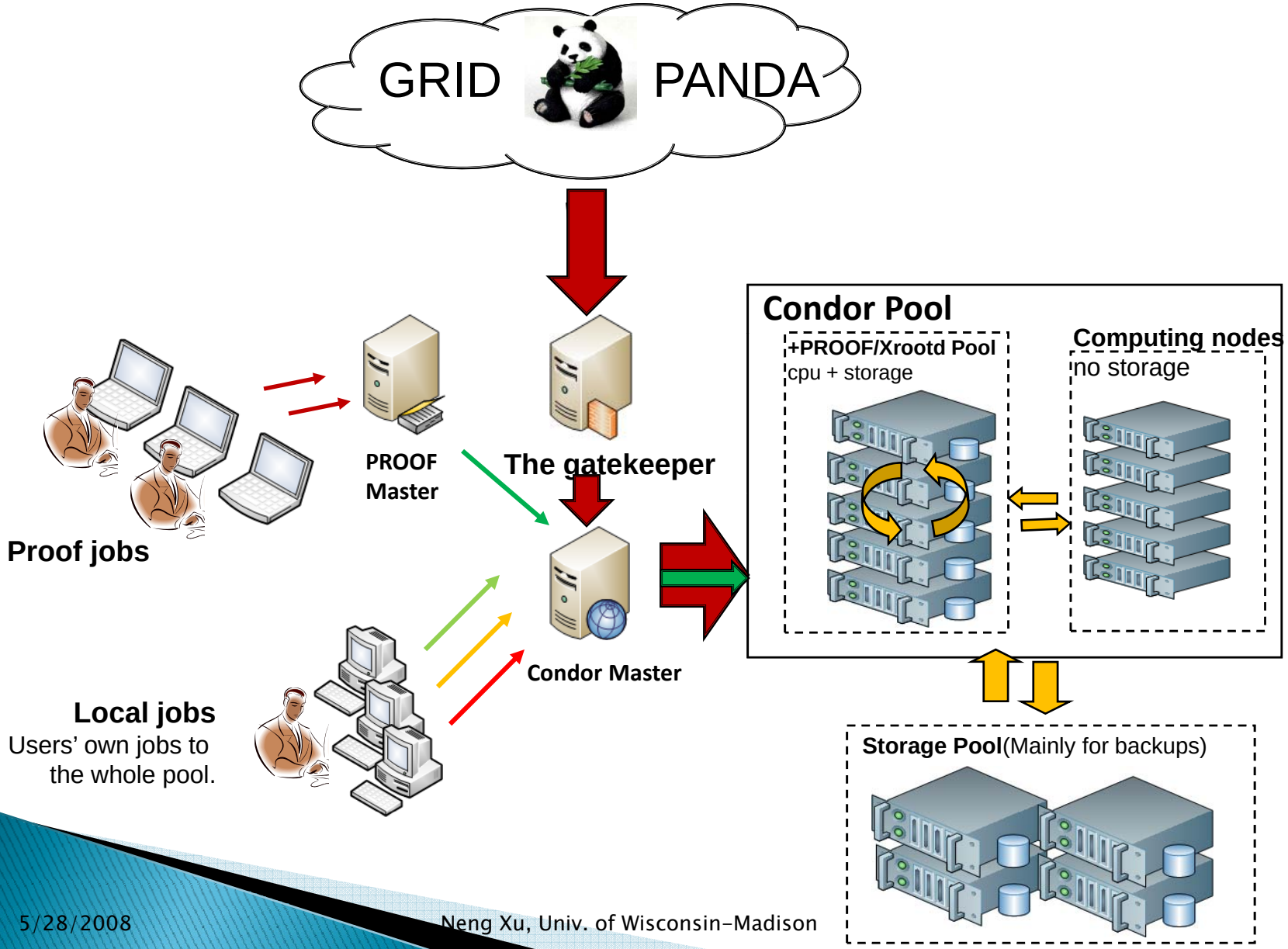
G. Ganis, J. Iwaszkiewicz, F. Rademakers
CERN/PH-SFT

Many thanks to:
M. Ernst, H. Ito, S. Panitkin, O. Rind, …

Outline

- System structure and current status of Wisconsin-GLOW.
- Current development of PROOF.

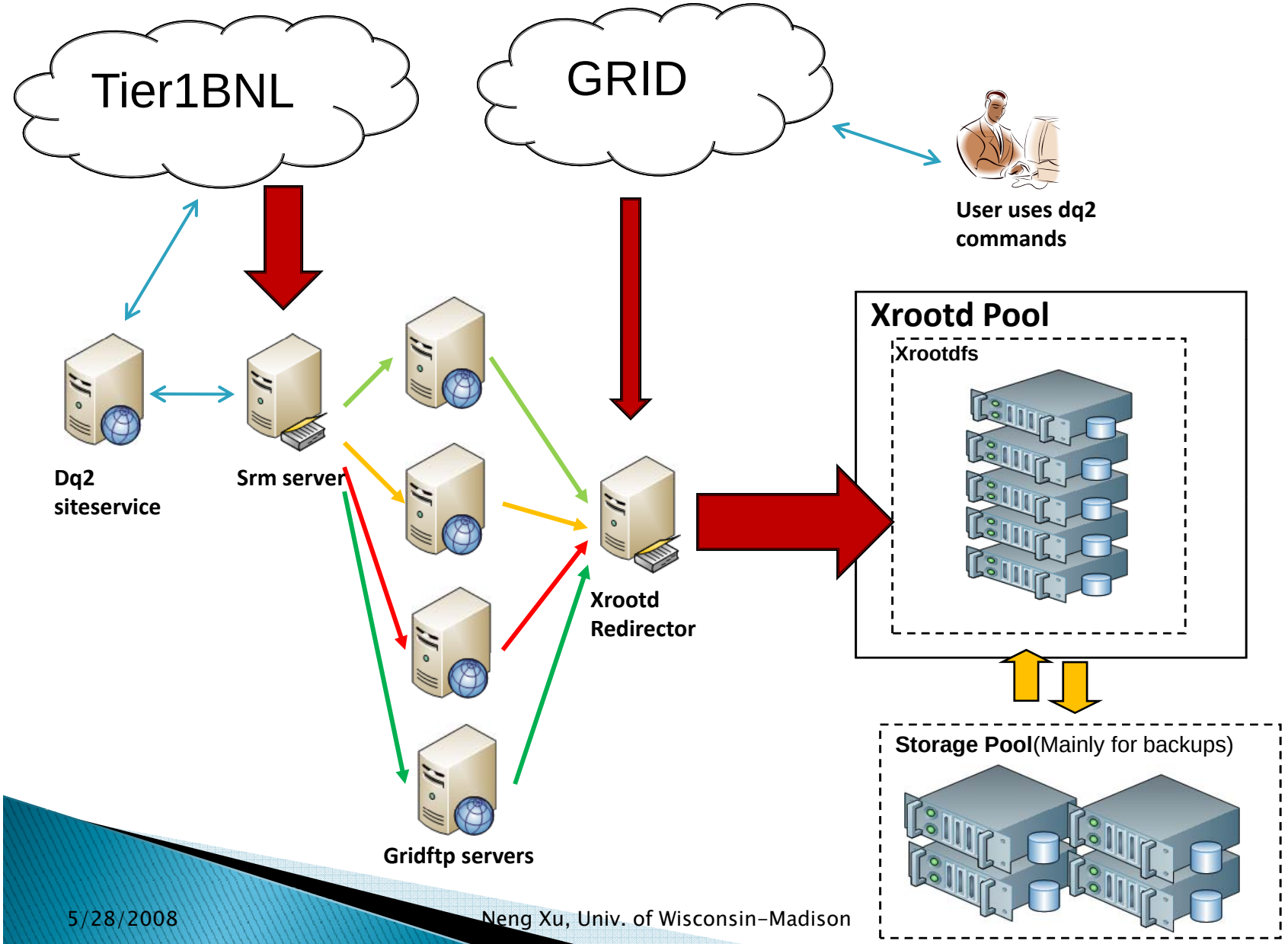
Our computing system



The computing Issues

- Computing resource management.
  - Normal batch jobs.
  - Analysis jobs (PROOF).
  - How to set a priority system which can control both of them.
- Data management.
  - How to get the data from Tier1/2(FTS, DQ2)?
  - How to manage the data(require, maintain, remove)?
  - How can user know what files are there?
  - What to do if data is lost? How to retrieve data?
- Manpower
  - How to build and maintain a system easily?
  - A complete instruction from A to Z.

Our data management system



Our hardware configuration

- CPU – Intel, 8cores, 2.66GHz
- Memory – 16GB DDR2
- System disk – 1x 160GB
- RAID Controller – 3ware 8ports hardware RAID
- Data disks – 8 x 750GB (7 disks on a RAID5 and 1 disk for hot-swap)
- NIC – 1Gb onboard

Our experience with RAID

- Software RAID
  - Software RAID on SLC4.
  - ZFS with FUSE on SLC4.
  - Open-Solaris ZFS.
- Hardware RAID
  - Single RAID array.
  - Multiple RAID arrays.
- Recommendations:
  - Software RAID has many good features but only works well when the CPU and memory are free. The CPU and memory resource for software RAID needs to be carefully considered.
  - Multiple RAID array can provide better performance but need to setup multiple Xrootd systems.

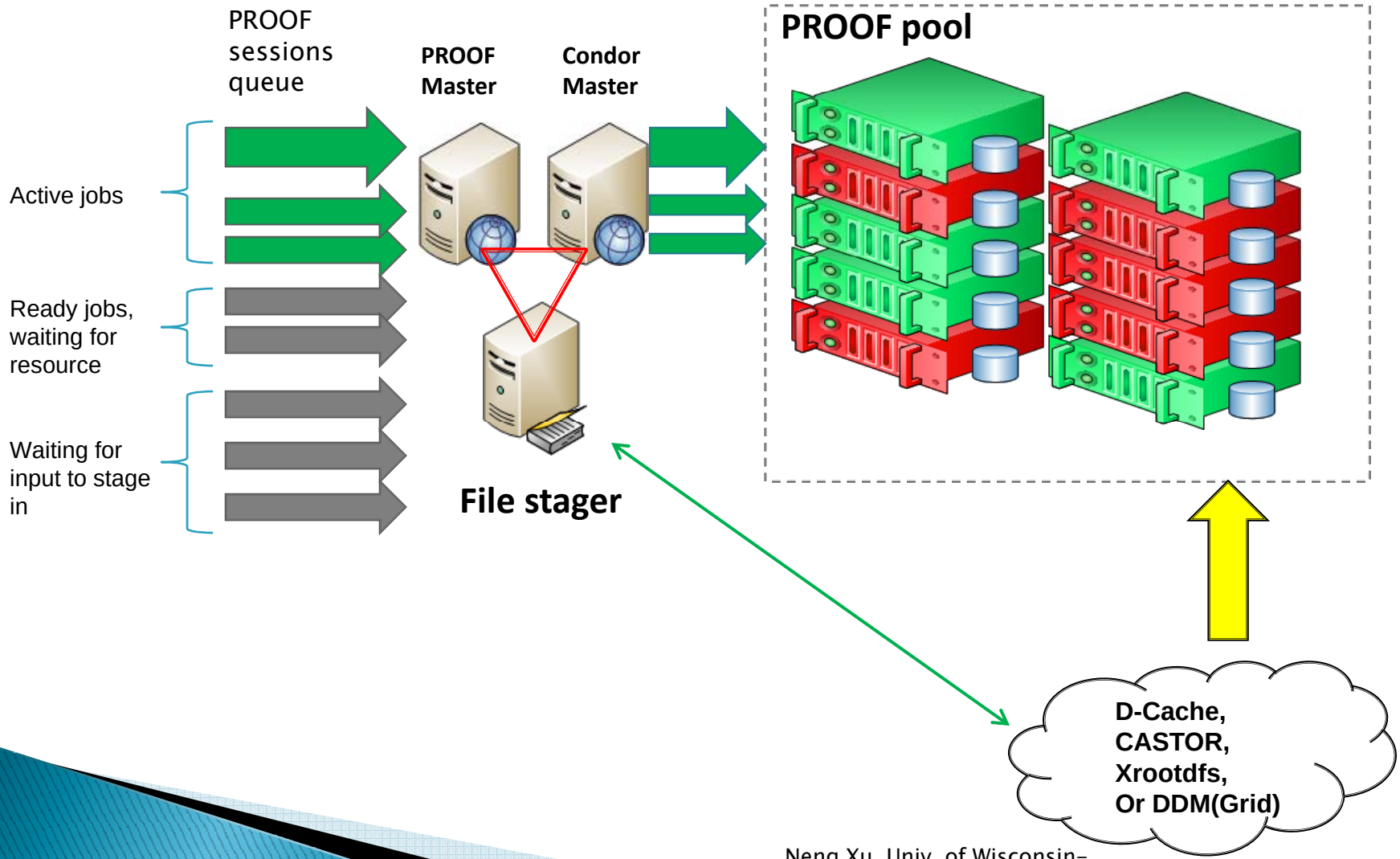
Our experience on data management

- We setup FTS between BNL and Wisconsin in 2007.
- We successfully got data from M4, M5, M6, FDR-I, CCRC, etc.
- We feel that the installation of FTS with Xrootd system is quite straightforward.
- The instruction of DQ2 site service is easy to follow.
- The maintenance of the system is quite light.
- The transfer speed increased from 2MB/s to >100MB/s. (Thanks to the taskforce lead by Shawn McKee, Jay Packard, Rob Gardner, …!)
- This enabled us to be able to run PANDA/PATHENA jobs on our own machines.
- PATHENA job output files were directly written to Xrootd pool and can be analyzed with PROOF.

New PROOF+CONDOR Model

- Started January 2008.
- Condor team joined the design.
- Mainly focus on
  - "Session level" scheduling.
  - More efficient Condor job suspension method than COD.
  - Enhanced the dataset management based on MySQL database.
  - Multi-layer file stage-in/out.
    (Tape <-> Harddrive <-> SSD <-> Memory)

Session level scheduling



Session Level Scheduling

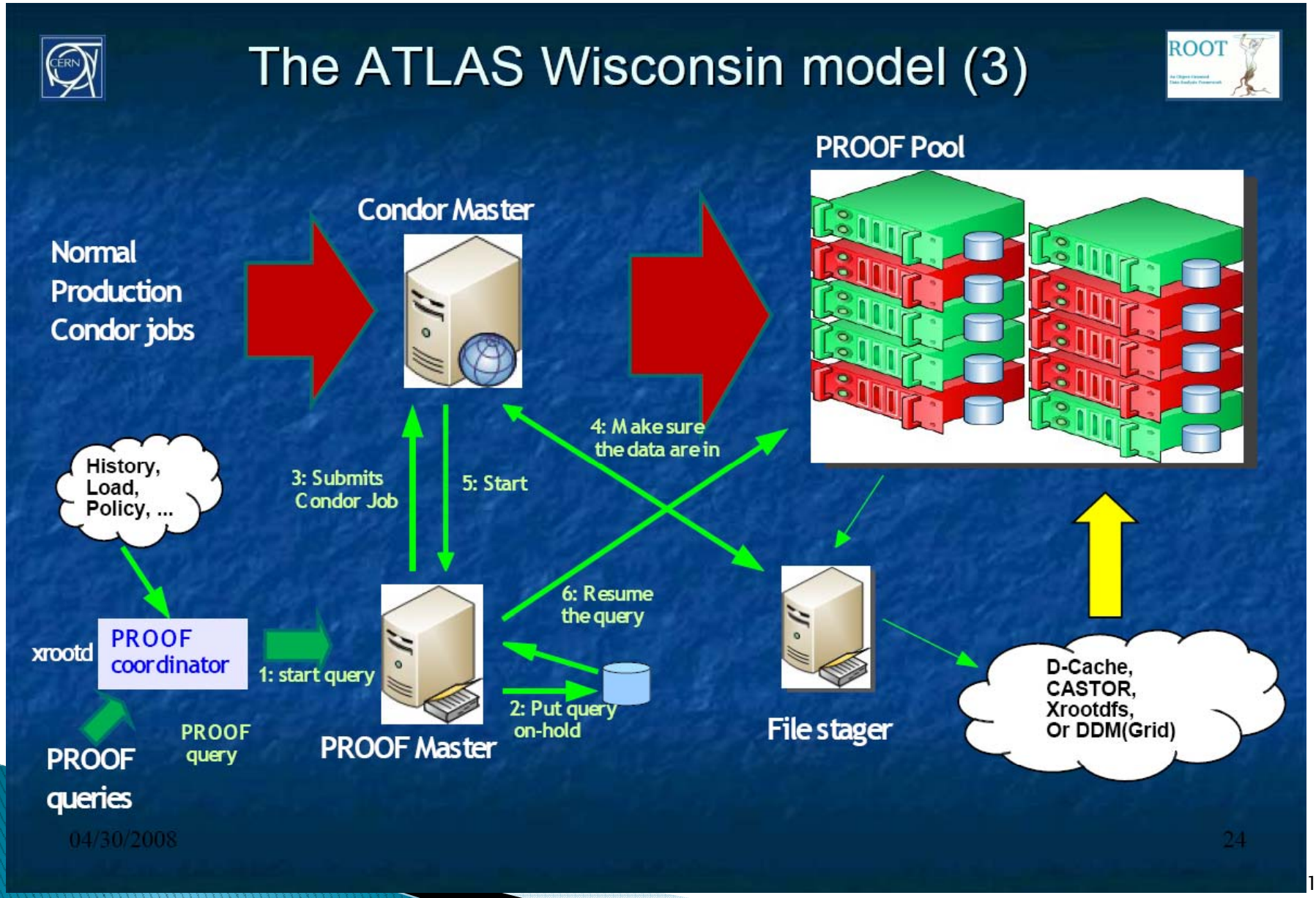
- CONDOR controls the number of running sessions based on the users' priority.
- File stager makes sure the input files are staged in to Xrootd pool. The processing won't be started until the dataset is ready.
- Condor ClassAds controls the session holding and releasing.

New Method of CPU suspension

- COD method does not consider or affect the system priority.
- CPU suspension can be easily done with job slots setting.
- PROOF can use high priority job slots which can suspend low priority job slots.
- PROOF can allocate CPU resource in a better way. (Free CPU first, suspend low priority jobs first, etc…)
- There is still some "deadtime" need to be understood.

More details (by Gerri Ganis, Condor week 2008)

http://www.cs.wisc.edu/condor/CondorWeek2008/condor_presentations/ganis_proof.pdf



New Dataset management

- PROOF provides a dataset manager which by default uses ROOT files to store information about datasets. This is used by ALICE in conjunction with a dedicated stager daemon, configured to stage out files from CASTOR and ALIEN.
- The dataset management functionality needed by PROOF has been abstracted out so that implementations for different backends can be provided.
- We plan to use the dataset manager with a MySQL backend, where we store all the relevant information about datasets (file location, date, size, status, etc…).
- Dataset stage in/out can be automated by daemon based on priority and disk quota.
- Planning to integrate into the DDM database.
- User only deal with datasets.

New Dataset management

- Multi-layer file stage-in/out
  (Tape <-> Harddrive <-> SSD <-> Memory)
- Database keeps the status and location of each dataset.
- Session scheduler will adjust session's priority based on the location of their dataset request.
- Datasets can be pre-staged in based on the priority.
- Hopefully, PROOF can work together with DQ2 to do this work.

New Dataset management

- Idea of integration with LRC database:

Tables in current LRC database

T_PFN

Id	Lfn
1	Dc3.
2	DC4.

T_LFN

Id	pfn
1	Gsiftp://
2	Gsiftp://

T_meta

MD5	size
dfsss	53242
daa	534

T_dataset

Dataset name	#of req	# of file	Last req date	Status	comment
mc08.017506.PythiaB_bbmu6mu4X.evgen.e306	2	50	2008/5/20	waiting	xx
mc08.017506.PythiaB_bbmu6mu4X.evgen.e306	1	50	2008/2/25	done	xx
mc08.017506.PythiaB_bbmu6mu4X.evgen.e306	1	500	2008/5/20	waiting	xx

The table PROOF would like to add!

Conclusion

- System structure:
  - We are satisfied with the Wisconsin-GLOW data storage and analysis model.
  - We hope the DQ2 can also work for local users' data registration.
  - We can prepare an instruction of building and maintaining the whole system for other sites if needed (RAID, Xrootd, PROOF, Condor, DQ2, PANDA, etc…).
- PROOF development:
  - The developing model is better prioritized and manageable for multi-user environment.
  - File/Dataset management will be more automated.
  - Need more interaction between PROOF team and ATLAS DDM team.