



**PROOF for ATDF**

**Sergey Panitkin**

**BNL**

**ATLAS**





# Outline

- ◆ Motivation
- ◆ Short introduction to PROOF
- ◆ PROOF for Atlas T3
  - ◆ Hardware and Software Considerations
- ◆ PROOF in Atlas
  - ◆ PROOF farm(s) at BNL
    - ◆ Recent developments
    - ◆ FDR1 experience
  - ◆ PROOF farm at Wisconsin
    - ◆ Computing on Demand (COD) and PROOF
    - ◆ Multiuser PROOF
- ◆ Summary



# Introduction I

- ◆ Large datasets will be a basic feature of Atlas physics analysis
  - ◆ Expect  $\sim 2 \times 10^9$  events per year
- ◆ In the context of current T3 discussion in Atlas the main questions are:
  - ◆ What are typical use cases for T3 in Atlas ?
  - ◆ How to make Tier 3 affordable, but still effective for Atlas analysis?
  - ◆ Is “interactive” (fast turn around) analysis possible at all?



# Introduction II

- ◆ Atlas Analysis Model and Event Data Model are still evolving
- ◆ Atlas analysis data (ESD, AOD, DPD) are written in **root** files
- ◆ POOL Root files written by Athena in AOD, D<sup>1</sup>PD, D<sup>2</sup>PD (?) formats
- ◆ Most likely D<sup>3</sup>PD will be written as plain root trees
- ◆ AthenaRoot Access (ARA) provides tools for accessing POOL root data –AOD, DPDs directly in Root, without Athena framework
  
- ◆ How to analyze  $\sim 10E9$  DPD events efficiently in Root?
- ◆ **Use PROOF!**

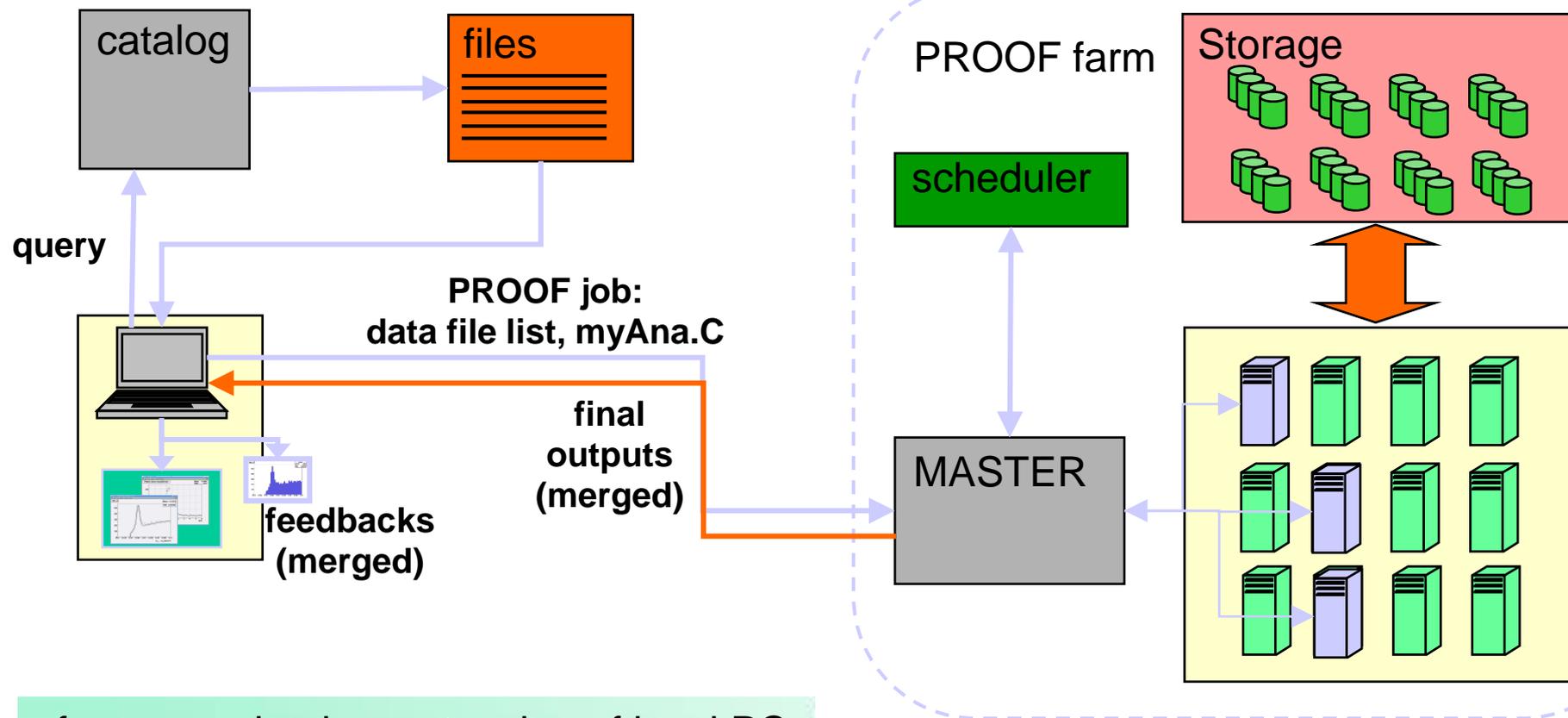


# PROOF system

- ◆ Parallel ROOT Facility, originally developed by MIT physicists
- ◆ A system for the **interactive** or **batch** analysis of **very large sets** of **Root** data files on a **cluster of computers**
- ◆ Speed up the query processing by employing inherent parallelism in event data
- ◆ Can run non-data-driven jobs (Monte Carlo simulation: Pythia, Toy MC, etc)
- ◆ The usage of PROOF is transparent for a root user . Feels like your laptop, only faster!
  - ◆ Same code can run locally and in a PROOF system (certain rules have to be followed- TSelector, etc)
  - ◆ Short learning curve
- ◆ PROOF is part of Root now. Distributed with Root
- ◆ PROOF is a plug in for Xrootd
- ◆ PROOF uses Xrootd for data discovery and file serving
- ◆ Well suited for (if not geared to) analysis farms with distributed **local** storage
  - ◆ Local data processing is encouraged – automatics matching code with data
  - ◆ Scales up trivially- just add more nodes
- ◆ Also, via Xrootd magic, can work with remote data

# PROOF processing

Jan Iwaszkiewicz, CERN



- farm perceived as extension of local PC
  - **same syntax** as in local session
- more dynamic use of resources
- real time feedback
- automated splitting and merging

- Basically “just” ROOT on a **distributed system**
  - **But** with interactive and batch commands, status information, etc.
  - Potential to interact **directly** with event data

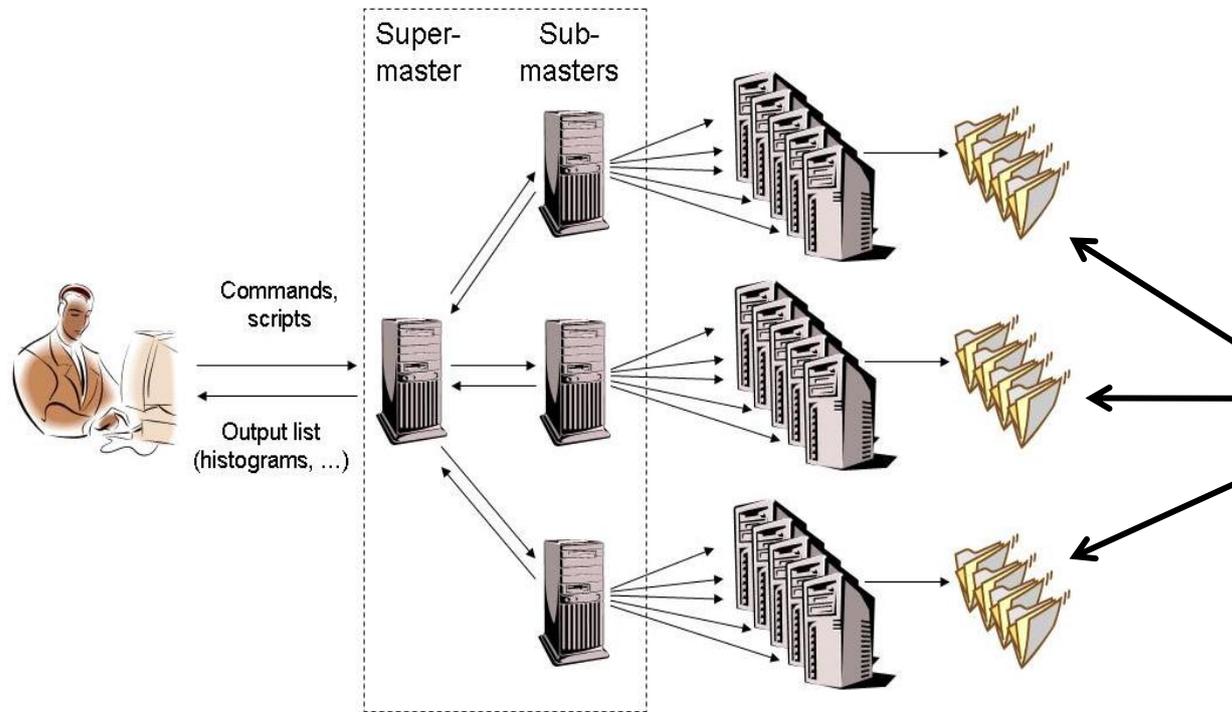
# Multi-Tier Architecture

Client

Master

Slaves

Files



Adapts to wide area  
*virtual* clusters

Geographically  
separated domains,  
heterogeneous  
machines

Super master is users' single point of entry. System complexity is hidden  
Automatic data discovery and job matching with local data  
Can be optimize for data locality or high bandwidth data server access

- ◆ **Alice**

- ◆ will run PROOF on CAF for calibrations, alignment, etc
- ◆ Large PROOF farm at GSI T2
- ◆ Various local T3s
- ◆ Integration of PROOF with AliRoot

- ◆ **Atlas**

- ◆ PROOF farm(s) at BNL T1
- ◆ PROOF farm at T2 at Munich LMU
- ◆ PROOF test farm at UTA T2
- ◆ Proof test farm at Universidad Autonoma de Madrid T2
- ◆ PROOF farm at Wisconsin T3

- ◆ **CMS**

- ◆ PROOF task force: CMS PROOF wiki
- ◆ PROOF cluster at Purdue - USCMS T2



# PROOF for Atlas T3

- ◆ We think that PROOF is an attractive technology for Atlas analysis centers. But raw performance alone is not enough for adoption.
- ◆ Important consideration in choosing PROOF for T3 will be “cost of ownership”.
- ◆ How labor intensive it is to operate PROOF farm in analysis environment?
- ◆ One needs to address questions typical for distributed environments:
  - ◆ How to manage the farm? Scalability, Load balancing, Security!?
  - ◆ What's going on? ->Monitoring, problem discovery/recovery
  - ◆ Where is my data? ->Tools for data management:
    - ◆ Farm file catalogs, etc
    - ◆ Tools for upload/download/removal/backup, etc
- ◆ One needs a suitable set of tools in order to keep farm operation workload at a reasonable level. PROOF alone does not provide all these tools!
- ◆ Integration with the rest of Atlas computing “ecosystem”
  - ◆ Interaction with T1s,T2s and other T3s centers
  - ◆ Grid tools, etc
- ◆ Experience of the first PROOF farm prototypes will be valuable for Collaboration.
  - ◆ “Know how”, recipes, mistakes, solution, etc



# General Remark

- ◆ It's difficult to give an advice on “how to install PROOF at T3”
- ◆ The pure PROOF/xrootd installation part is easy
  - ◆ Instructions are available on PROOF web page at CERN
- ◆ The difficult part is integration with existing local environment
  - ◆ Existing infrastructure
    - ◆ Existing legacy farms and SE
    - ◆ Grid tools
    - ◆ Monitoring tool
  - ◆ Administration, Management and Support customs
  - ◆ Local Security regime
- ◆ Local physics group commitments, interests and plans
  - ◆ Dedicated PROOF farm (ARA on AOD, DPD, etc)
  - ◆ Multipurpose farm, Panda site



# What do I need for PROOF farm?

## ◆ Hardware:

- ◆ Linux farm: any AMD or Intel (preferably multi-core) CPUs
- ◆ Adequate network infrastructure: 1Gb/s or better is desirable
- ◆ Storage: Local, in-node disks or centralized (dCache, NFS vaults, etc).
  - ◆ PROOF can use local disk storage very efficiently.

## ◆ Software

- ◆ Linux (SL, Debian, Ubuntu, etc..)
- ◆ Root. If you have Root, you have PROOF/Xrootd libraries installed already
  - ◆ Latest versions are recommended. Atlas rel. 14 will have Root v 5.18
- ◆ Atlas software
- ◆ Some farm management tools, if you run a farm
  - ◆ Many exist, I use *tentakel* to run commands in parallel on multiple nodes
- ◆ Some Farm Monitoring tool
  - ◆ Ganglia, Mona Lisa, XrdMon, etc
  - ◆ Software to supports monitoring: MySQL, web server, etc
- ◆ Tools for Atlas specific DDM (see previous talk by Marco)

# Hardware Considerations

- ◆ The main issue is to match I/O demand and supply
- ◆ “Rule of two thumbs”
  - ◆ One I/O bound Root job (usually 1 per core) requires ~10MB
  - ◆ One SATA HDD can sustain ~20 MB/s of random I/O (~2 jobs)
- ◆ 8 core machine with 1 SATA disk makes little sense in PROOF context. Unless you will feed PROOF jobs with data from external SE (dCache, NFS disk vault, etc)
- ◆ In the latter case same rule applies to network bandwidth. Make sure that your network infrastructure can sustain  $\sim N \cdot 10$  MB/s transfer rate from SE, where N is a number of PROOF jobs/cores.
- ◆ Solid State Disks (SSD) are ideal for PROOF (our future?). Tests started
- ◆ Memory : “The more the merrier!”
  - ◆ Xrootd makes efficient use of file caching. Atlas recommended 2 GB per core is a good ballpark number.
- ◆ **Caution:** 10MB/s figure of merit will jump to ~15+ MB/s with arrival of more powerful CPUs (Penryn, Barcelona, etc)

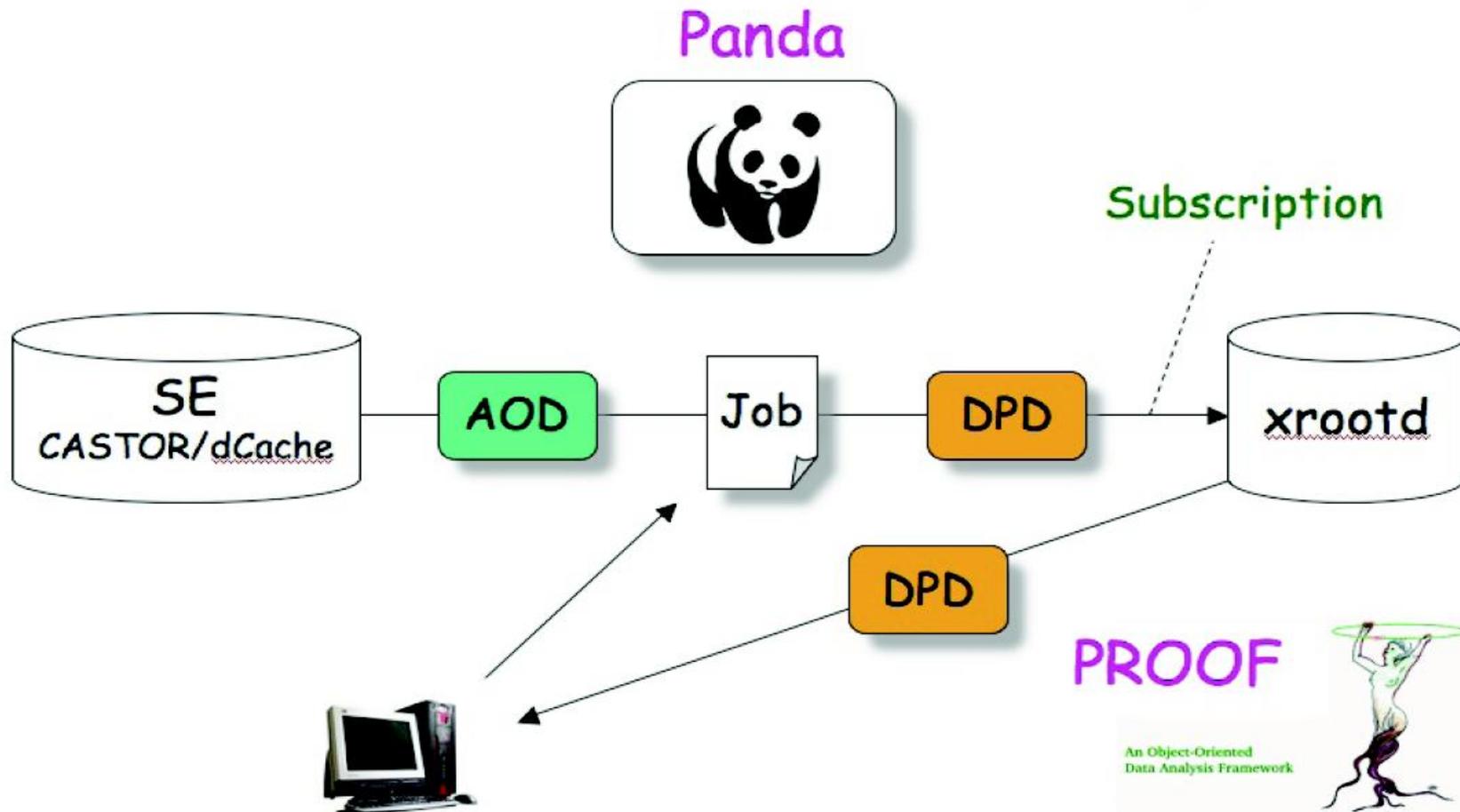


# Importing and Managing datasets

- ◆ You need to bring data to your local PROOF cluster.
  - ◆ GridFTP, etc
  - ◆ Local Grid connected SE (dCache, Xrootd, etc)
- ◆ Some integration with Atlas DDM is necessary (see Marco's talk)
  - ◆ LRC
  - ◆ dq2 tools
- ◆ Xrootd generic tool for file copy is xrdcp.
  - ◆ Load balanced file placement
  - ◆ Multi-stream file transfer
  - ◆ Checksum calculations
- ◆ XrootdFS , FUSE may be necessary for POSIX like access
  - ◆ Several groups are looking at this. SLAC, Wisconsin

# Integration with Atlas DA

Tested in 2007 by Tadashi Maeno



We can also use dCache as an end point storage element in PANDA and then pull DPDs from dCache via Xrootd door



# Integration with Atlas DDM. FDR1 Experience

- ◆ We ran BNL PROOF farm under Root v 5.14 for compatibility with rel. 13 data
  - ◆ 36 cores, ~15TB of disk space
- ◆ All the data (AODs, DPDs, etc) first arrived at dCache
- ◆ All AODs and DPDs were copied from dCache to the PROOF farm for analysis in root
- ◆ File transfer was done using custom Perl scripts
- ◆ Datasets were copied using xrdcp via Xrootd door on dCache
  - ◆ Fall back solution exists in case Xrootd door on dCache is unstable: dccp+xrdcp
  - ◆ Was used several times for files larger than 2GB
- ◆ Xrootd was “added” to LRC and ToA. Hiro Ito (SE name is BNLXRDHDD1)
- ◆ Copied datasets were registered in LRC with dq2\_rc modified by Tadashi
- ◆ AthenaRootAccess analysis was run on AODs and DPDs by S. Ye, H.Ma, A. Shibata
- ◆ Xrootd was used as SE for Athena analyses. A. Shibata and D. Adams
- ◆ **It was a useful exercise and many bugs and kinks were discovered and fixed!**

# Integration with Atlas DDM

Tested during FDR 1

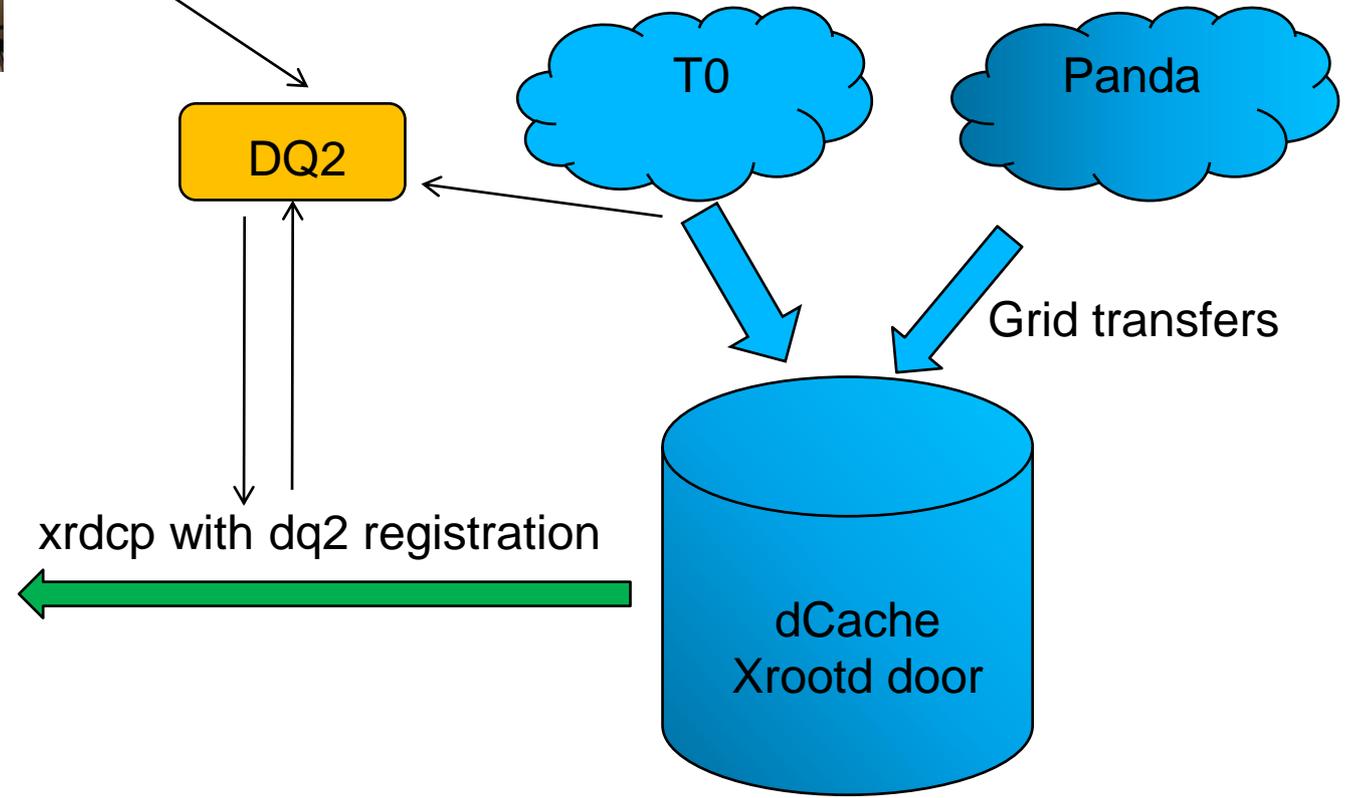
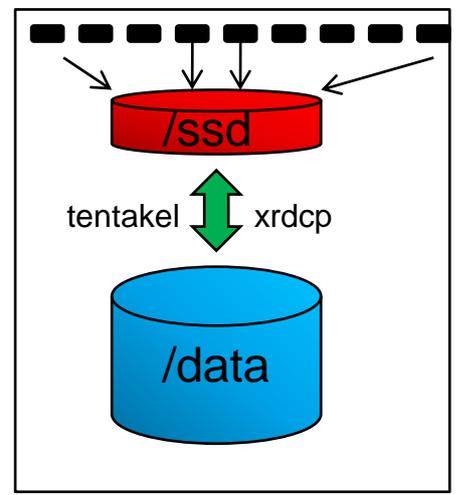
Atlas user



Command line interface:  
`dq2_ls -fp -s BNLXRDHDD1 "my_dataset"`

Analysis ↓

Xrootd/PROOF Farm



`root://acas0420//data/datasetname/filename`



# AthenaRootAcces on PROOF

- ◆ Demonstrated feasibility of running ARA analysis in PROOF (S. Ye, S. Snyder)
  - ◆ Instructions at:  
<http://www.usatlas.bnl.gov/twiki/bin/view/AtlasSoftware/ProofTestBed>
- ◆ Analysis with FDR1 AODs, DPDs performed at BNL by Hong Ma .
- ◆ Important milestone!
- ◆ Tests show that for current ARA implementation, event rate is only ~18 times faster on PROOF with 36 nodes than for single ARA job.
- ◆ Proof helps but a lot of room for improvement! Some code optimization was already made

# Farm monitoring using Ganglia



Cluster Report for Thu, 4 Oct 2007 14:38:06 -0400

Get Fresh Data

Metric mem\_report Last hour Sorted by hostname

Physical View Alerts: RSS feed

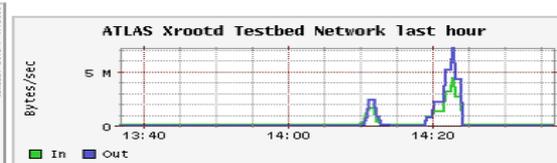
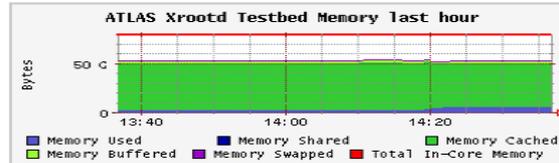
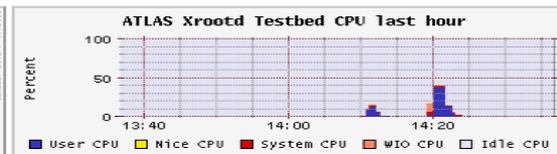
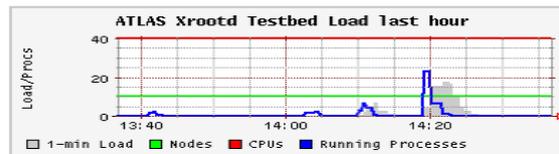
BNL ATLAS Computing Facility Grid > ATLAS Xrootd Testbed > --Choose a Node

## Overview of ATLAS Xrootd Testbed

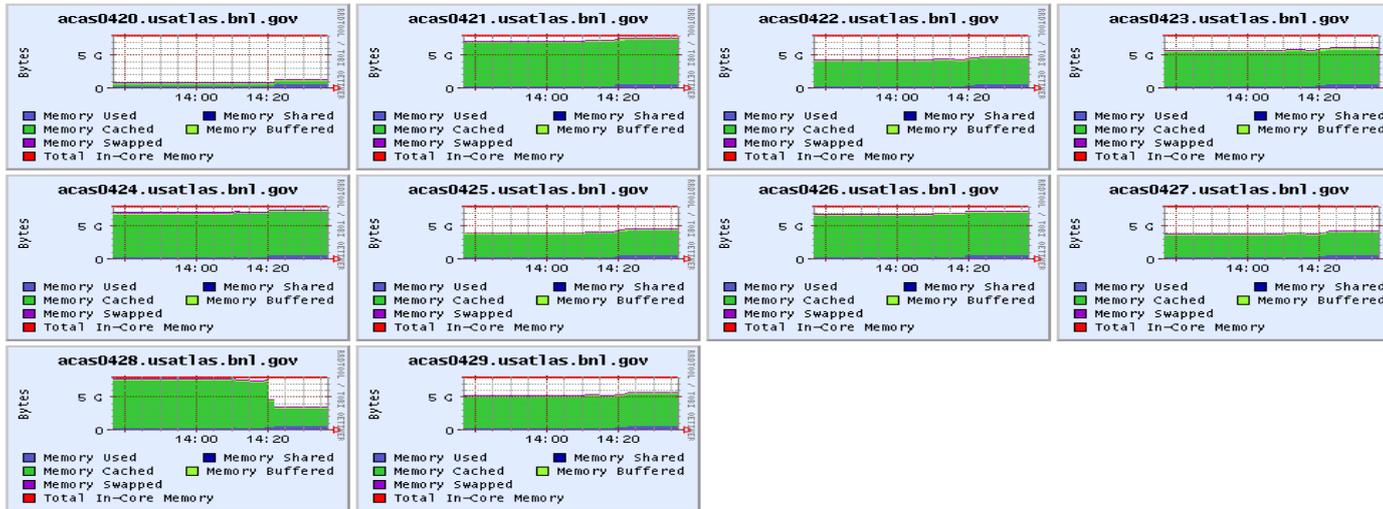
CPU's Total: 40  
Hosts up: 10  
Hosts down: 0

Avg Load (15, 5, 1m):  
2%, 3%, 4%  
Localtime:  
2007-10-04 14:36

Cluster Load Percentages  
0-25 (100.00%)



Show Hosts: yes | ATLAS Xrootd Testbed mem\_report last hour sorted by hostname | Columns 4



Farm wide view at hardware/OS level (loads, network, memory, disk space, etc)

# Farm monitoring using XrdMon

XrdMon developed in SLAC

## Xrootd data access monitoring

Basic view

Top performers

List all [users](#) [dataTypes](#) [files](#) [servers](#) [clients](#) [jobs](#)

List current [users](#) [dataTypes](#) [files](#) [servers](#) [clients](#) [jobs](#)

Query by [user](#) [dataType](#) [file](#) [server](#) [client](#) [job](#)

[Common queries](#)

[Xrootd statistics](#)

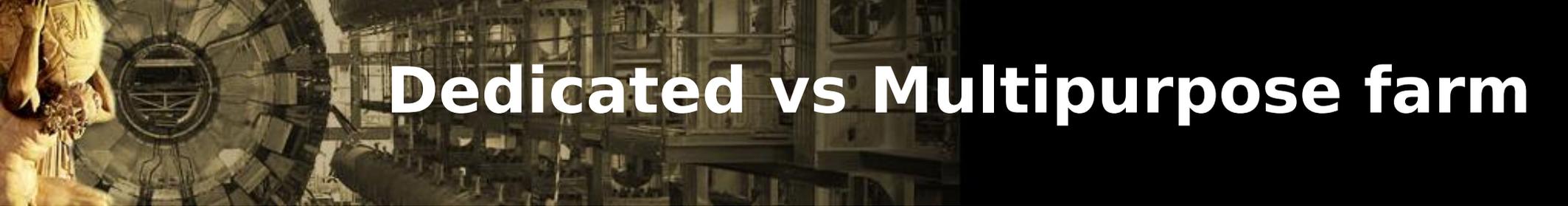
Table rows: 10 Time Period: Last Month Site: usatlas Update

Top active users				Last Month			
User Name	Number of Jobs	Number of Files	File Size [MB]	Number of Jobs ↑	Number of Files	File Size [MB]	MB Read
<a href="#">xrdadmin</a>	0	0	0	5,899	5,834	252,832	0
<a href="#">casadei</a>	0	0	0	489	2,391	131,651	37,130
<a href="#">serp</a>	22	90	0	271	1,685	44	2,555
<a href="#">tarrade</a>	0	0	0	204	223	0	4,524
<a href="#">dladams</a>	0	0	0	28	3	0	3,007
<a href="#">akira</a>	0	0	0	2	1	61	16

Hottest dataTypes				Last Month					
dataType Name	Number of Jobs	Number of Files	File Size [MB]	Number of Users	Number of Jobs ↑	Number of Files	File Size [MB]	Number of Users	MB Read
<a href="#">HPTV</a>	0	0	0	0	5,020	4,587	256,589	4	37,169
<a href="#">MUON_1</a>	0	0	0	0	997	969	0	2	3,007
<a href="#">HiggsToTauTau-00-00-44</a>	22	90	0	1	836	339	0	3	7,056
<a href="#">serp</a>	0	0	0	0	2	1,519	0	1	0

Hottest files				Now	Last Month
File Path	File Size [MB]	Number of Jobs	Number of Files	Number of Jobs	MB Read
<a href="#">/data/cache/HiggsToTauTau-00-00-44/user.TARRADEFabien.trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.A12.0.6.9.medium_cut_el.AAN.AANT1_00103.root</a>	0	7	19	110	
<a href="#">/data/cache/HiggsToTauTau-00-00-44/user.TARRADEFabien.trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.A12.0.6.9.medium_cut_el.AAN.AANT1_00114.root</a>	0	6	20	50	
<a href="#">/data/cache/HiggsToTauTau-00-00-44/user.TARRADEFabien.trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.A12.0.6.9.medium_cut_el.AAN.AANT1_00117.root</a>	0	6	47	99	
<a href="#">/data/cache/HiggsToTauTau-00-00-44/user.TARRADEFabien.trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.A12.0.6.9.medium_cut_el.AAN.AANT1_00113.root</a>	0	5	24	70	
<a href="#">/data/cache/HiggsToTauTau-00-00-44/user.TARRADEFabien.trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.A12.0.6.9.medium_cut_el.AAN.AANT1_00116.root</a>	0	4	22	60	
<a href="#">/data/cache/HiggsToTauTau-00-00-44/user.TARRADEFabien.trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.A12.0.6.9.medium_cut_el.AAN.AANT1_00107.root</a>	0	4	28	80	
<a href="#">/data/cache/HiggsToTauTau-00-00-44/user.TARRADEFabien.trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.A12.0.6.9.medium_cut_el.AAN.AANT1_00106.root</a>	0	3	22	34	

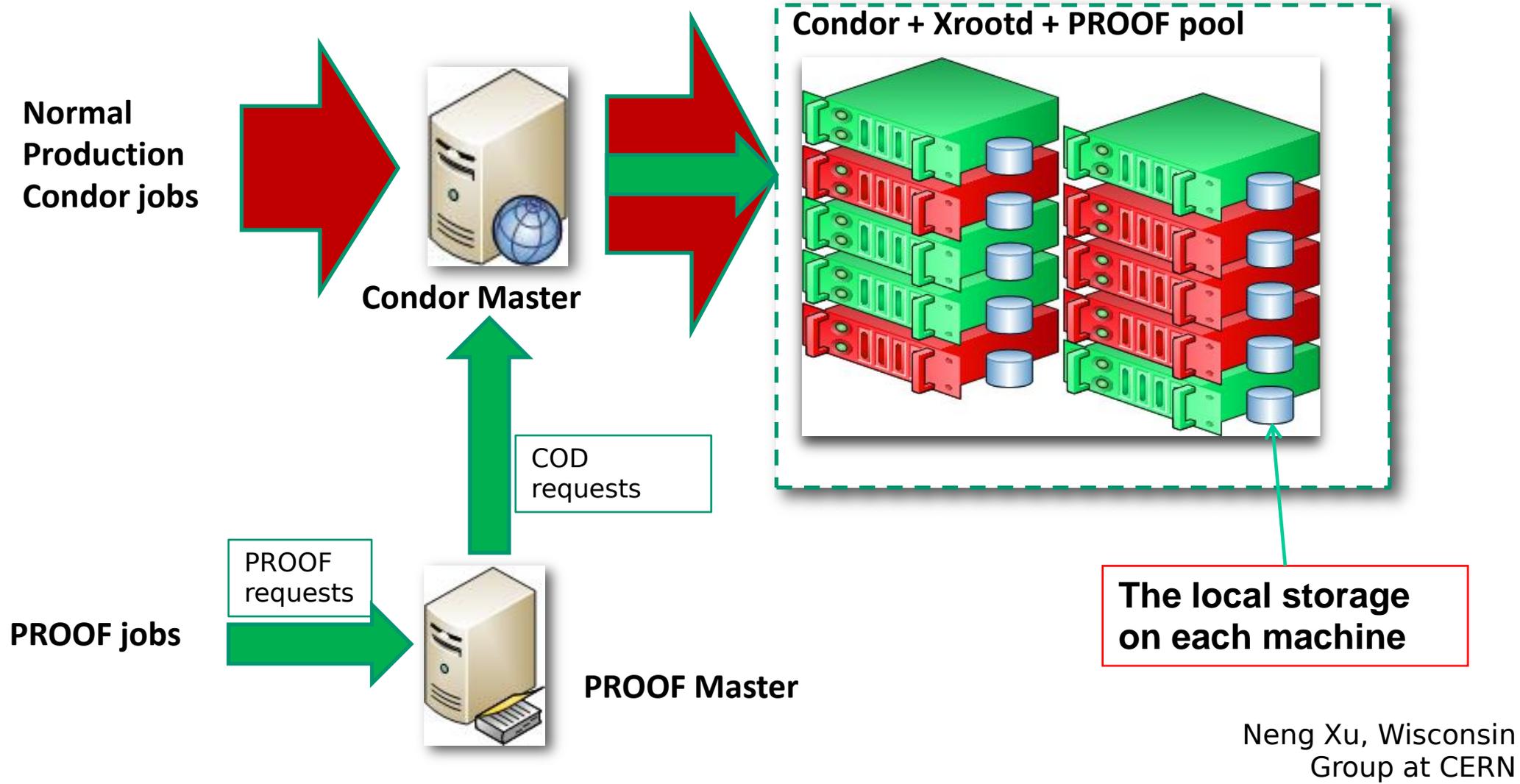
A summary of the data accessed during the last month



# Dedicated vs Multipurpose farm

- ◆ What model to choose for Atlas T3 ?
  - ◆ Pure PROOF farm
  - ◆ Pure batch farm
  - ◆ Combination of both?
- ◆ How to deal with uneven inflow of PROOF jobs?
- ◆ I think it will depend on each particular center/group goals and circumstances
- ◆ For example BNL's role and goals are different from Wisconsin's or Munich's groups.
- ◆ Combination of PROOF and Condor batch looks promising
- ◆ Wisconsin group studies of COD. Work closely with PROOF team

# The basic PROOF+COD Model

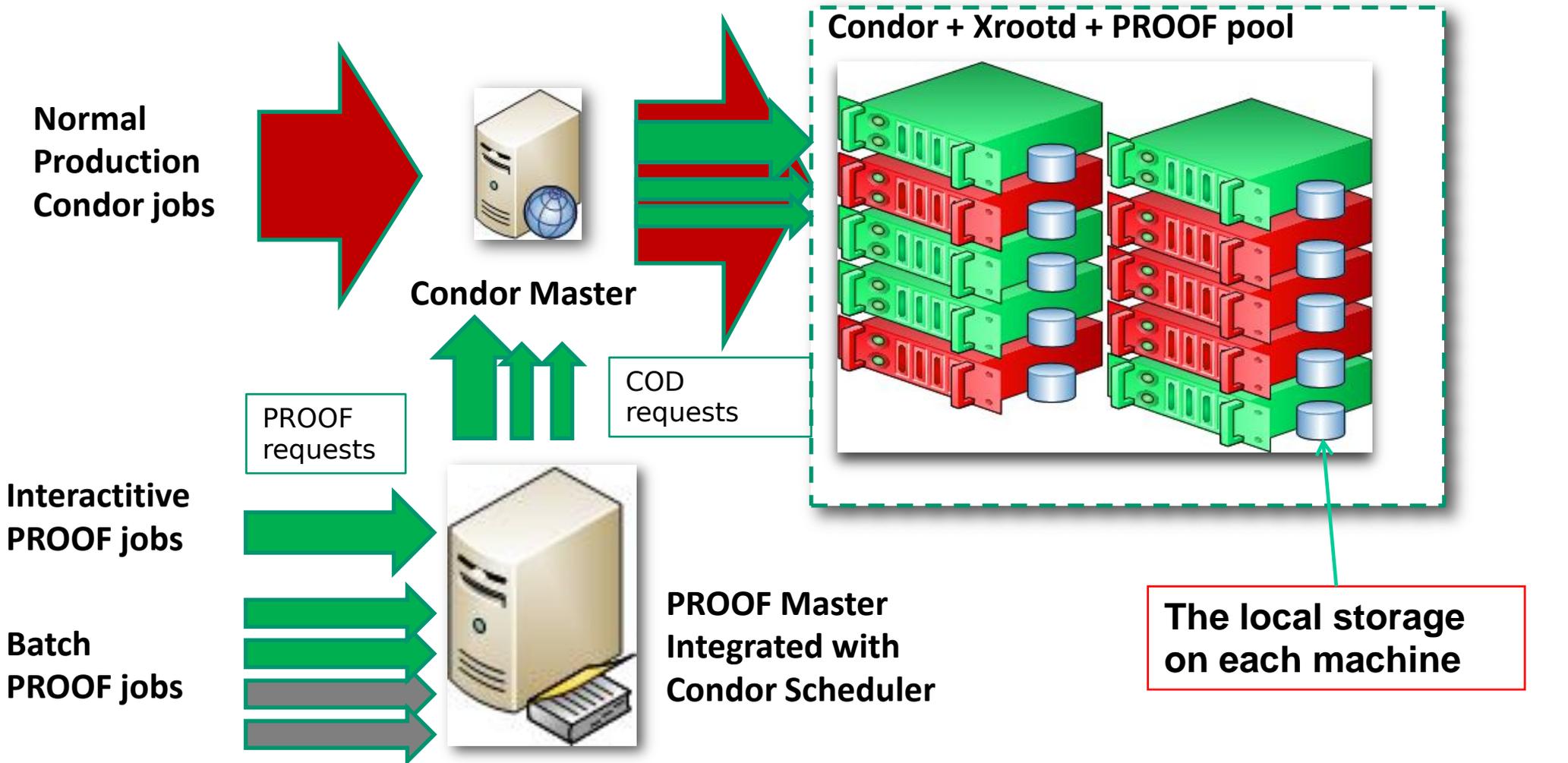




# The basic PROOF+COD Model

- ◆ Good combination of PROOF pool and Production pool.
- ◆ No empty CPU cycles.
- ◆ Production/batch jobs won't be affected.
- ◆ PROOF jobs get immediate CPU resources.
- ◆ Transparent to PROOF users.
- ◆ Good for a small Tier3 site with <10 users.

# Possible Future PROOF Model

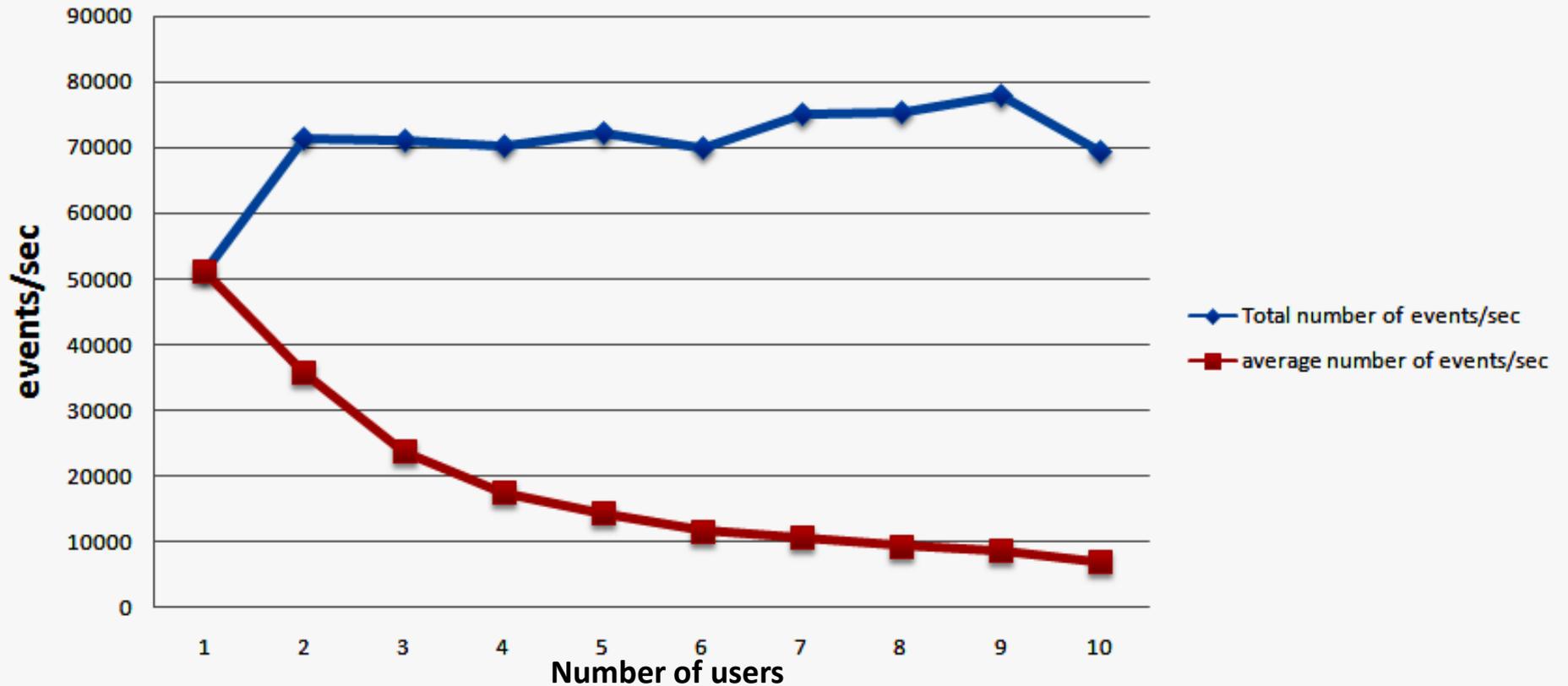


Neng Xu, Wisconsin Group at CERN

# Proof Multiuser Performance

Neng Xu  
Wisconsin

## Total Speed vs Average Speed



### Test environment settings:

- 42 nodes with 4 PROOF workers per node. Total 168 PROOF workers.
- AMD 4core, 4GB memory, 70GB dedicated data disk.
- Default scheduling setting, every session gets all the 168 workers.

Sergey Panitkin

# User support and documentation

- ◆ Main PROOF Page at CERN, PROOF worldwide forum
  - ◆ <http://root.cern.ch/twiki/bin/view/ROOT/PROOF>
- ◆ USAtlas Wiki PROOF page created (Thanks Rob!)
  - ◆ <http://www.usatlas.bnl.gov/twiki/bin/view/ProofXrootd/WebHome>
- ◆ Web page/TWIKI at BNL with general farm information, help, examples, tips, talks, links to Ganglia page, etc.
  - ◆ <http://www.usatlas.bnl.gov/twiki/bin/view/AtlasSoftware/ProofTestBed>
- ◆ Hypernews forum for Atlas PROOF users created  
[hn-atlas-proof-xrootd@cern.ch](mailto:hn-atlas-proof-xrootd@cern.ch)  
<https://hypernews.cern.ch/HyperNews/Atlas/get/proofXrootd.html>
- ◆ **PROOF tutorial is planned for the next Analysis Jamboree at BNL, in March**
- ◆ Do we need US Atlas T3 task/help force?



# Summary

- ◆ PROOF/Xrootd is an attractive technology for Atlas T3 centres
- ◆ Several PROOF test farms are operational in Atlas
- ◆ Significant experience with PROOF was gained
  - ◆ Several Atlas analysis scenarios were tested, with good results
  - ◆ AthenaRootAccess was shown to work on PROOF. Used during FDR1
  - ◆ Improved integration with Atlas DDM was demonstrated during FDR1
  - ◆ PROOF farms are used for analysis by several physicists
- ◆ Integration with Condor is being explored by Wisconsin group.
- ◆ Working prototypes/examples of management and monitoring setup exist.
- ◆ Wiki page is available for Atlas PROOF users with examples, etc
- ◆ Hypernews forum for Atlas PROOF user is set up
- ◆ Do we need US Atlas T3 task/help force?



# Acknowledgement

Many thanks to people involved in PROOF activities in Atlas!

Kevin Black, Kyle Cranmer, Michael Ernst, Tadashi Maeno,  
Robert Petkus, Ofer Rind, Akira Shibata, Fabien Tarrade,  
Torre Wenaus, Shuwei Yu

**BNL, NYU, Harvard**

Matthias Schott, Johannes Elmsheuser, Otto Schaile

**LMU-Munich**

Mengmeng Chen, Annabelle Leung, Bruce Mellado,  
Sau Lan Wu, Neng Xu

**Wisconsin-Madison**

Many thanks to Root/PROOF/Xrootd team! Rene, Fons, Gerri, Jan,  
Andy, Fabrizio



# The Packetizer

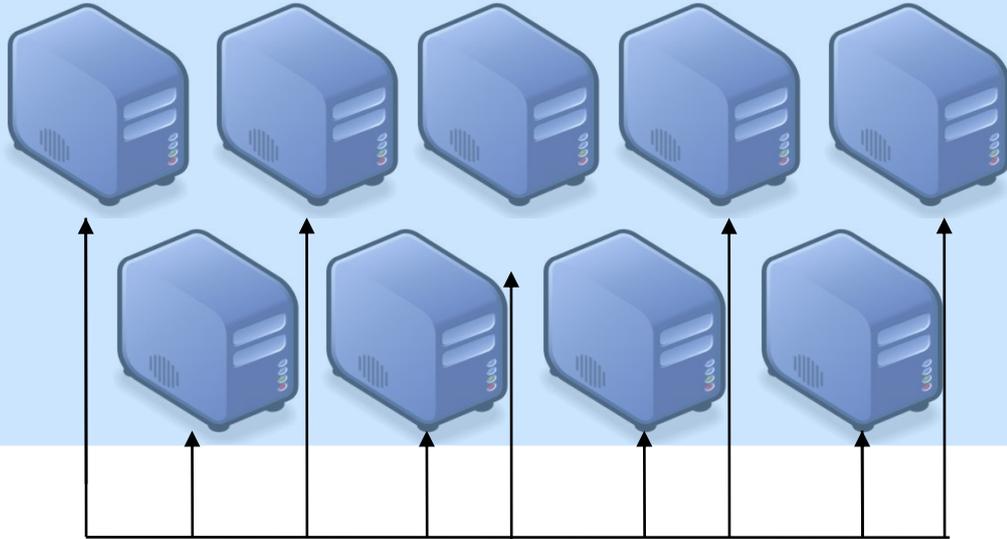
- ◆ The packetizer is the heart of the system.
- ◆ It runs on the master and hands out work to the workers
- ◆ Makes sure all workers end at the same time
- ◆ Different packetizers allow for different data access policies
  - ◆ All data on disk, allow network access
  - ◆ All data on disk, no network access
  - ◆ Data on mass storage, go file-by-file
  - ◆ Data on Grid, distribute per Storage Element
- ◆ User can choose a packetizer at the beginning of PROOF session.
  - ◆ `P->SetParameter("PROOF_Packetizer", TPacketizer);`
- ◆ So far there are:
  - ◆ **TAdaptivePacketizer** (Default one, with dynamic packet size)
  - ◆ **TPacketizer** (Optional one, with fixed packet size)
  - ◆ **TForceLocalPacketizer** (Special one, no network traffic between workers. Workers only deal with the file stored locally)

## Pull architecture

workers ask for work, no complex worker state in the master

# Current Xrootd farm configuration at BNL

Xrootd Data Servers + PROOF Workers



(9) Data Servers + PROOF Workers each with:

- (2) dual-core 1.8GHz Opteron processors
- (4) 500GB SATA disks (1.8TB) configured RAID0
- Scientific Linux 4.4
- xrd v.20070716-0300, root v5.16

(1) Redirector + PROOF Master + Xrootd monitor (Perl, MySQL) configured as above

(1) Apache Tomcat server for monitoring display (XrdMon)

Xrootd Redirector  
Xrootd Monitor  
PROOF Master



Apache Tomcat

