Parallel ROOT Facility Status and Plans

Application Area Internal Review

Fons Rademakers

Outline of Presentation

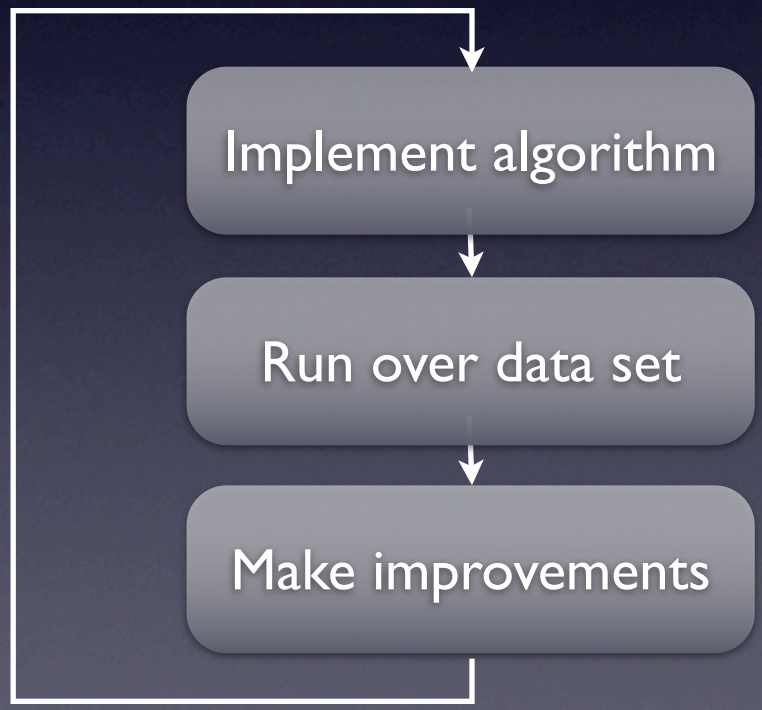
- PROOF motivation
- PROOF features and status
- PROOF testing and deployment
- PROOF development plans
- Conclusions

The PROOF Team

- Maarten Ballintijn
- Bertrand Bellenot
- Leandro Franco
- Gerri Ganis
- Jan Iwaszkiewicz
- Andreas Peters
- Fons Rademakers

Motivation

- Typical HEP analysis needs a continuous algorithm refinement cycle



HEP Final Analysis

- Ranging from I/O bound to CPU bound
- Need many disks to get the needed I/O rate
- Need memory to cache as much as possible
- Need many CPUs for processing

Data Analysis Hardware

- Aim for the highest possible I/O rate per CPU
- Use local disks or make sure to have high bandwidth to remote storage
- A good amount of RAM for efficient data caching

Some ALICE Numbers

- 1.5 PB of raw data per year
- 360 TB of ESD+AOD per year (20% of raw)
- One pass using 400 disks at 15 MB/s will take 16 hours
- Using parallelism is the only way to analyze this amount of data in a reasonable amount of time

PROOF Design Goals

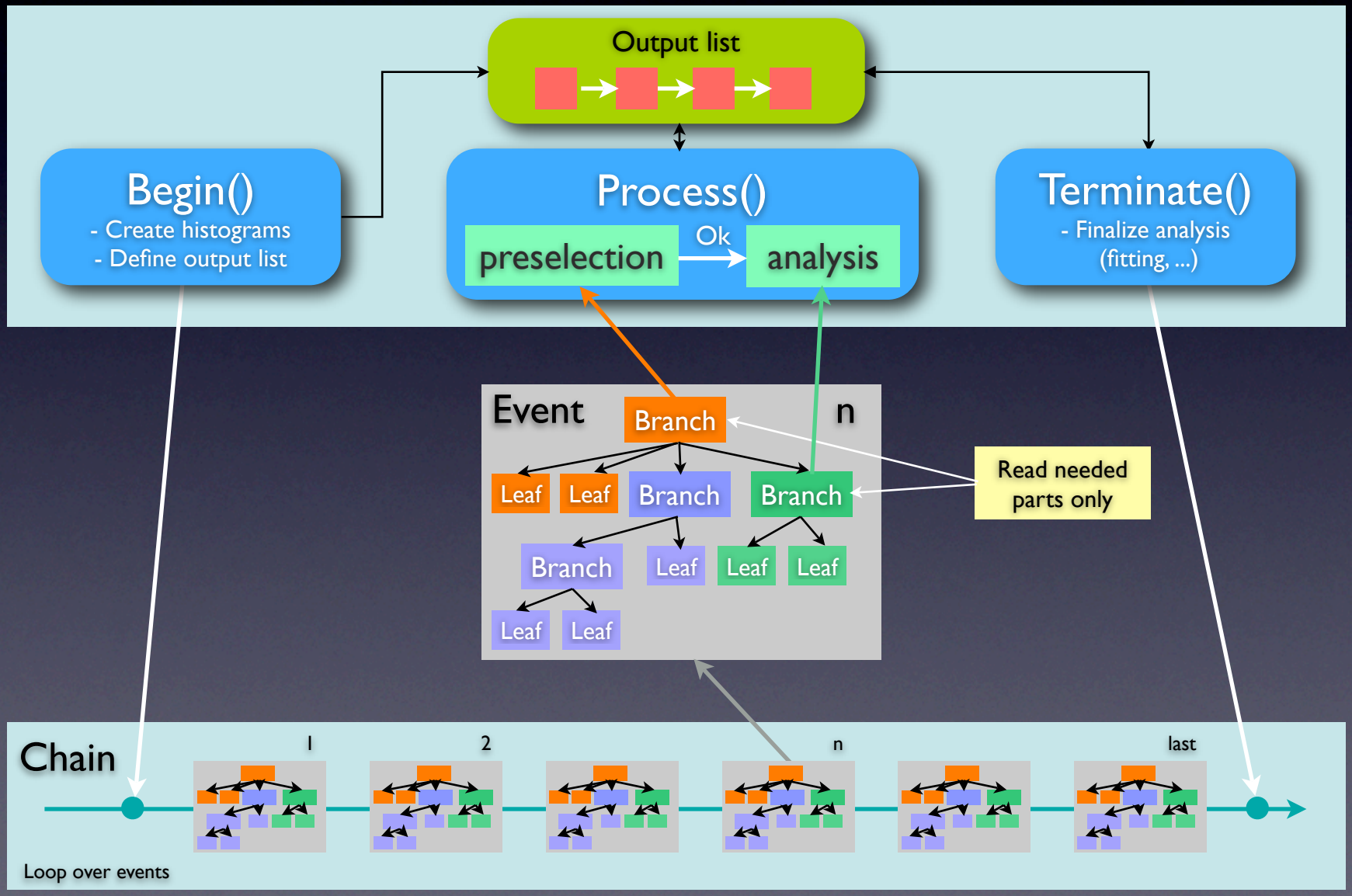
- System for running ROOT queries in parallel on a large number of distributed computers
- PROOF is designed to be a transparent, scalable and adaptable extension of the local interactive ROOT analysis session
- Extends the interactive model to long running "interactive batch" queries

Where to Use PROOF

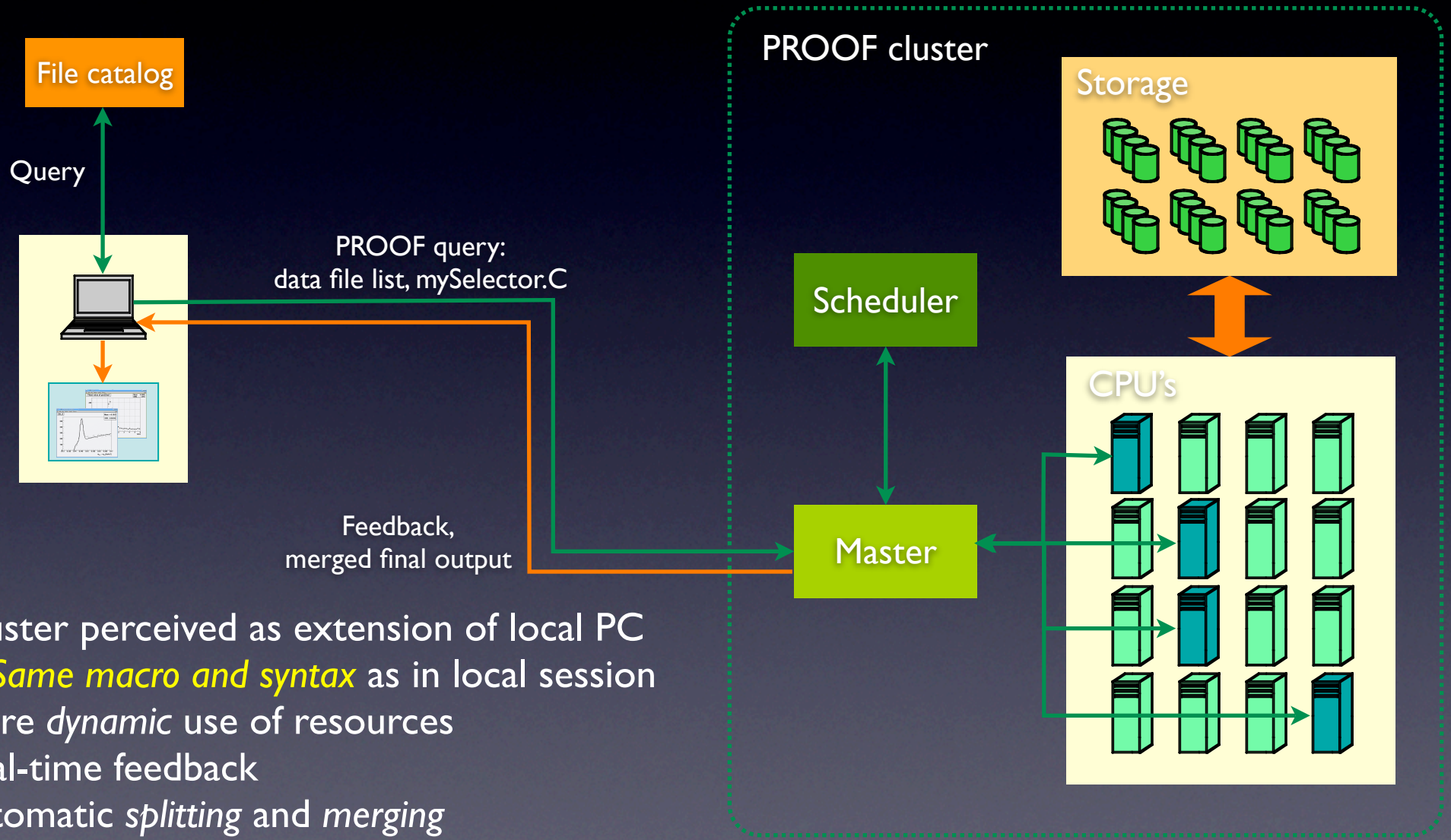
- Central Analysis Facility (CAF)
- Departmental workgroups (Tier-2's)
- Multi-core, multi-disk desktops (Tier-3/4's)

The ROOT Data Model

Trees & Selectors

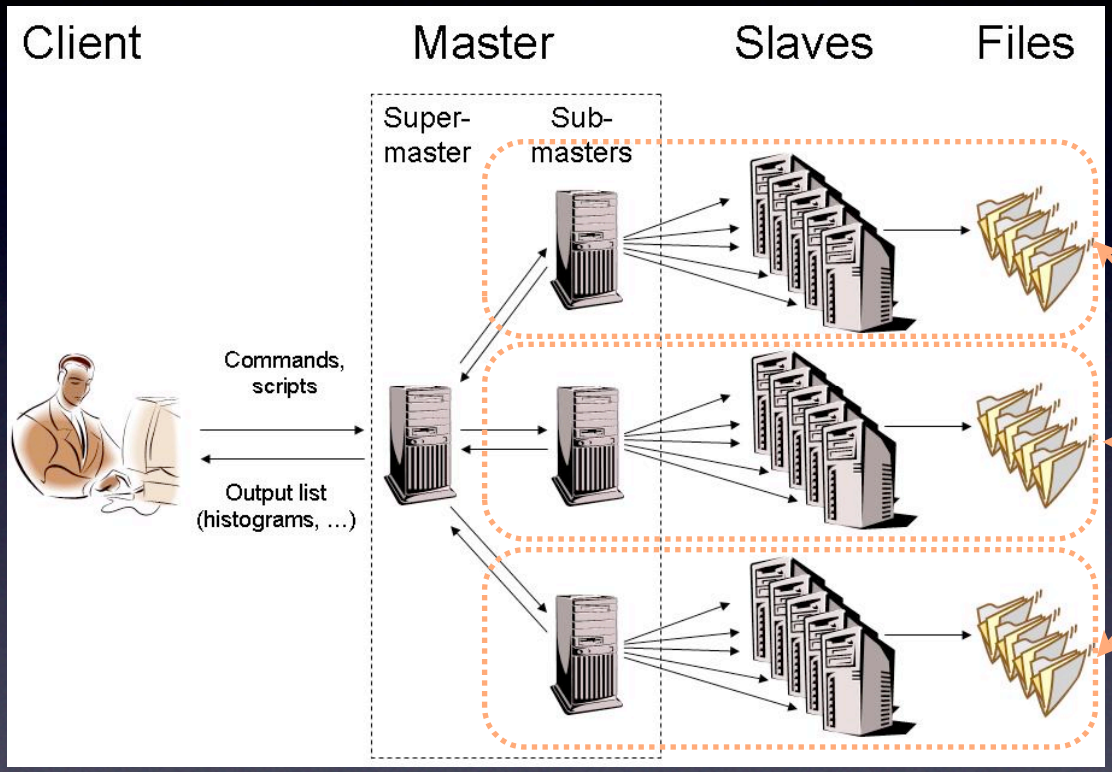


The PROOF Approach



- Cluster perceived as extension of local PC
  - *Same macro and syntax* as in local session
- More *dynamic* use of resources
- Real-time feedback
- Automatic *splitting* and *merging*

Multi-Tier Architecture



Adapts to wide area *virtual* clusters

Geographically separated domains, heterogeneous machines

Network performance

Less important → VERY important

Optimize for data locality or high bandwidth data server access

TSelector - User Code

```
// Abbreviated version
class TSelector : public TObject {
protected:
    TList *fInput;
    TList *fOutput;
public
    void   Init(TTree*);
    void   Begin(TTree*);
    void   SlaveBegin(TTree *);
    Bool_t Process(int entry);
    void   SlaveTerminate();
    void   Terminate();
};
```

TSelector::Process()

```
...
...
// select event
b_nlhk->GetEntry(entry);          if (nlhk[ik] <= 0.1)     return kFALSE;
b_nlhpi->GetEntry(entry);         if (nlhpi[ipi] <= 0.1)   return kFALSE;
b_ipis->GetEntry(entry); ipis--; if (nlhpi[ipis] <= 0.1) return kFALSE;
b_njets->GetEntry(entry);         if (njets < 1)           return kFALSE;

// selection made, now analyze event
b_dm_d->GetEntry(entry);          //read branch holding dm_d
b_rpd0_t->GetEntry(entry);        //read branch holding rpd0_t
b_ptd0_d->GetEntry(entry);        //read branch holding ptd0_d

//fill some histograms
hdmd->Fill(dm_d);
h2->Fill(dm_d,rpd0_t/0.029979*1.8646/ptd0_d);
...
...
```

The Packetizer

- The packetizer is the heart of the system
- It runs on the master and hands out work to the workers
- 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
- Current packetizer uses fixed number of event per packet

PROOF Scalability

- CAF, 4 dual Xeon machines
- CMS selector, 120 MB data (290 files), distributed on the 4 machines
- Strictly concurrent user sessions (100% CPU used)

- No inefficiencies introduced by PROOF internals

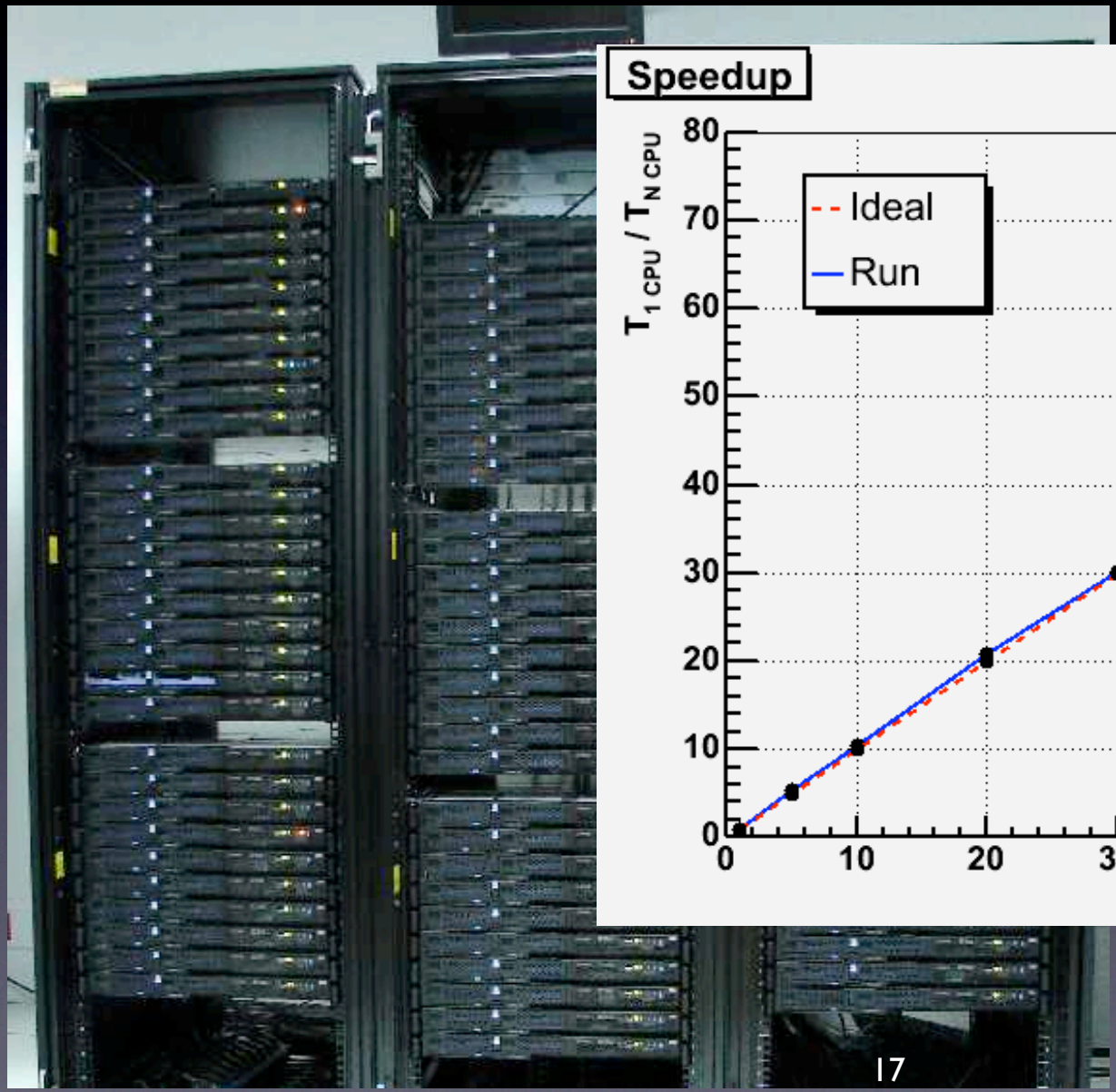
Some More Test Results

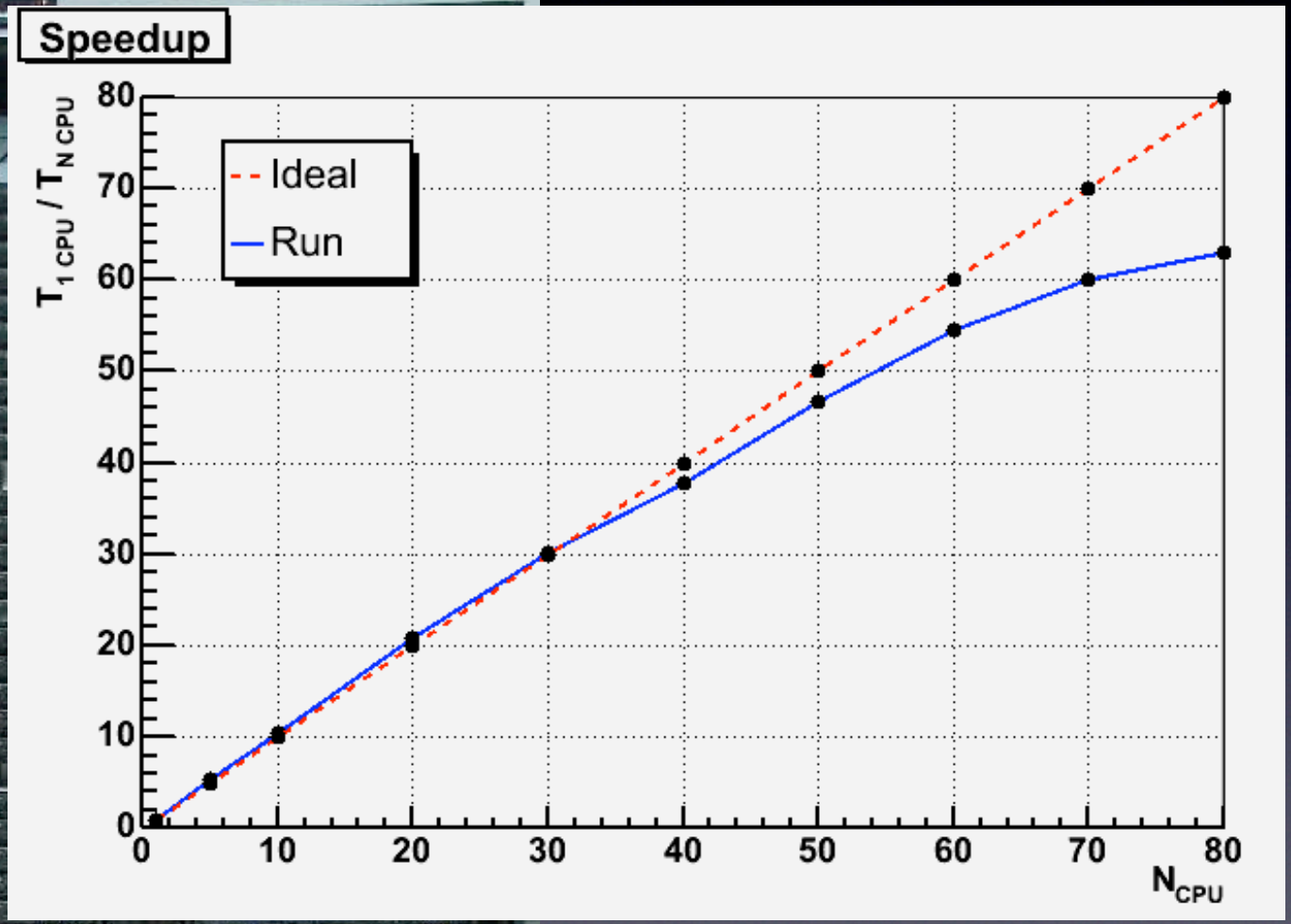
- 90 nodes
- 2 CPU Xeon 3.2 GHz
- 480 GB SATA disk
- Non-blocking GB Eth
- 1 master, 80 slaves
- 10K events per node, 1.4 GB

- On 1 CPU about 4 hours
- On 80 CPUs about 4 min

From I. Gonzalez, Univ. de Cantabria

Some More Test Results





Production Usage in Phobos

PROOF in PHOBOS

Maarten Ballintijn / MIT

maartenb@mit.edu

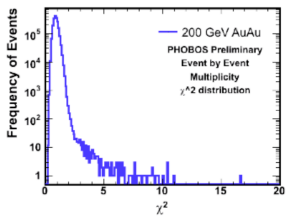
May 24, 2006 – Application Area Meeting

Components of the Facility

- 25 Interactive Nodes
- 425 Compute nodes w/ distributed disk
  - 100 TB disk space
  - Mix of 100Mb and 1Gbit Ethernet
- HPSS tape robot / Mass Storage System
- Centralized disk space
  - NFS (0.9 TB) – home directories, software
  - Panasas (3.8 TB) – data, proof work directories

May 24th, 2006 PROOF in PHOBOS 6

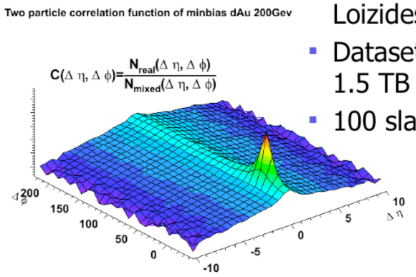
Rare high multiplicity event search



- Burak Alver
- Dataset: 11k files, 4.5 TB
- 150 slaves, ~1 hour

May 24th, 2006 PROOF in PHOBOS 13

Two Particle Correlations @ 200GeV



- Wei Li, Constantin Loizides
- Dataset: 4.5k files, 1.5 TB
- 100 slaves, 75 min

May 24th, 2006 PROOF in PHOBOS 15

Interactive Batch

- Allow submission of long running queries
- Allow client/master disconnect, reconnect
- Allow interaction and feedback at any time during the processing

Analysis Scenario

AQ1: 1s query produces a local histogram
AQ2: a 10m query submitted to PROOF1
AQ3 - AQ7: short queries
AQ8: a 10h query submitted to PROOF2

Monday at 10:15
ROOT session on my laptop

BQ1: browse results of AQ2
BQ2: browse intermediate results of AQ8
AQ3 - AQ6: submit 4 10m queries to PROOF1

Monday at 16:25
ROOT session on my desktop

CQ1: browse results of AQ8, BQ3 - BQ6

Wednesday at 8:40
Browse from any web browser

New xrootd Based Connection Layer

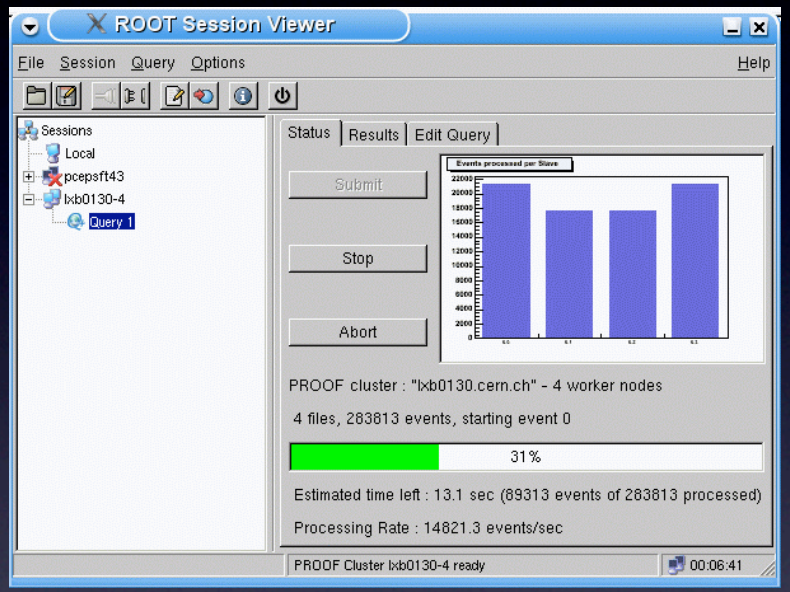
- Interactive batch requires a coordinator on the server side
- Use xrootd
  - Light weight, industrial strength, networking and protocol handler
  - New PROOF protocol, xpd, implemented as a plug-in
  - Plug-in launches and controls PROOF sessions
- Disconnect / reconnect handled naturally
- Can use the same daemon for data and PROOF serving

Management Tools

- Data sets
  - Distribution of data files on the PROOF cluster
    - By direct upload
    - By staging out from mass storage (e.g. CASTOR)
- Query results
  - Retrieve and archive
- Packages
  - Optimized upload of additional libraries needed by the analysis

Session Viewer GUI

- Open/close sessions
- Define a chain
- Submit a query, execute a command
- Query editor

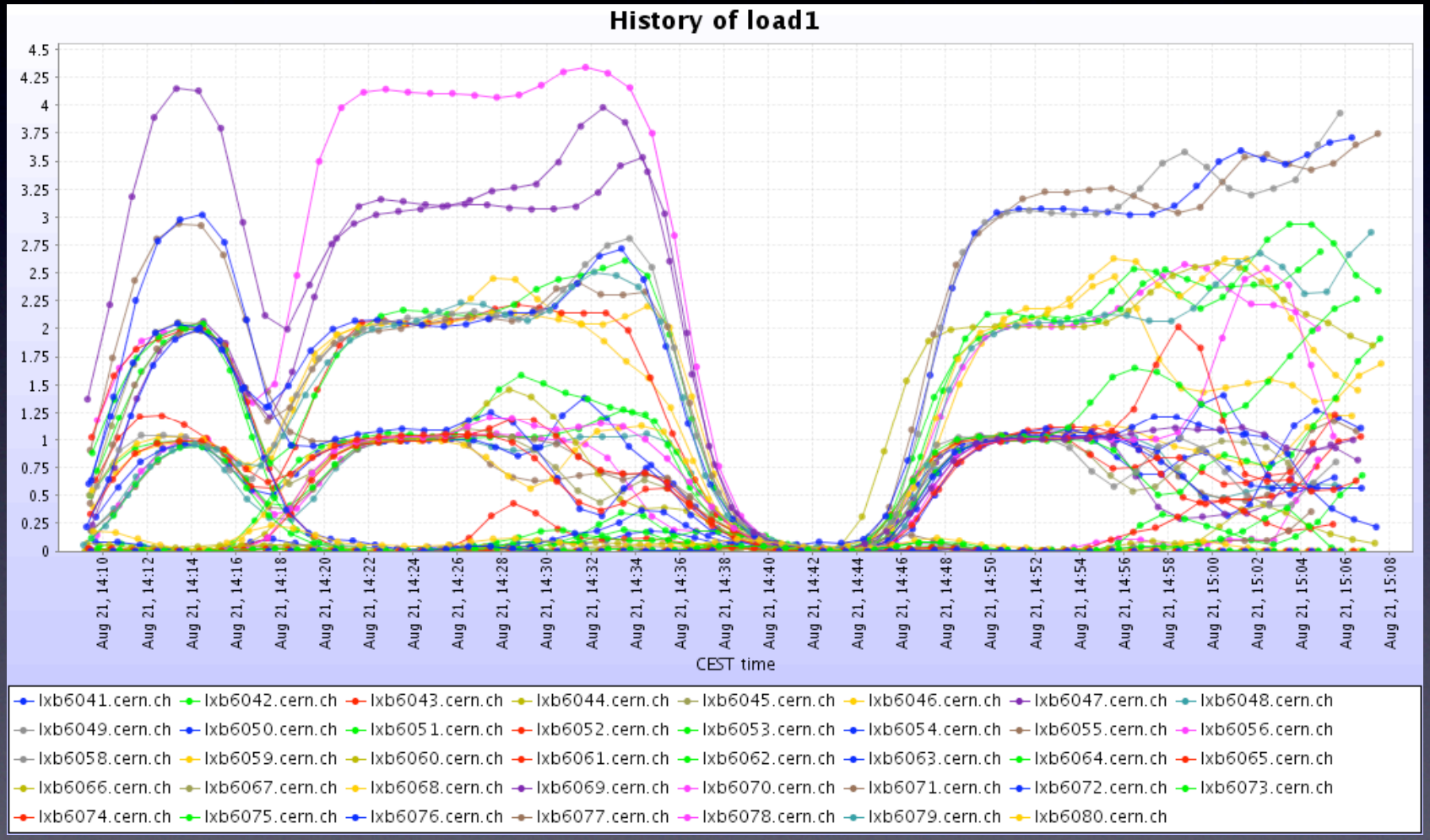


- Online monitoring of feedback histograms
- Browse folders with query results
- Retrieve, archive and delete query results

Monitoring

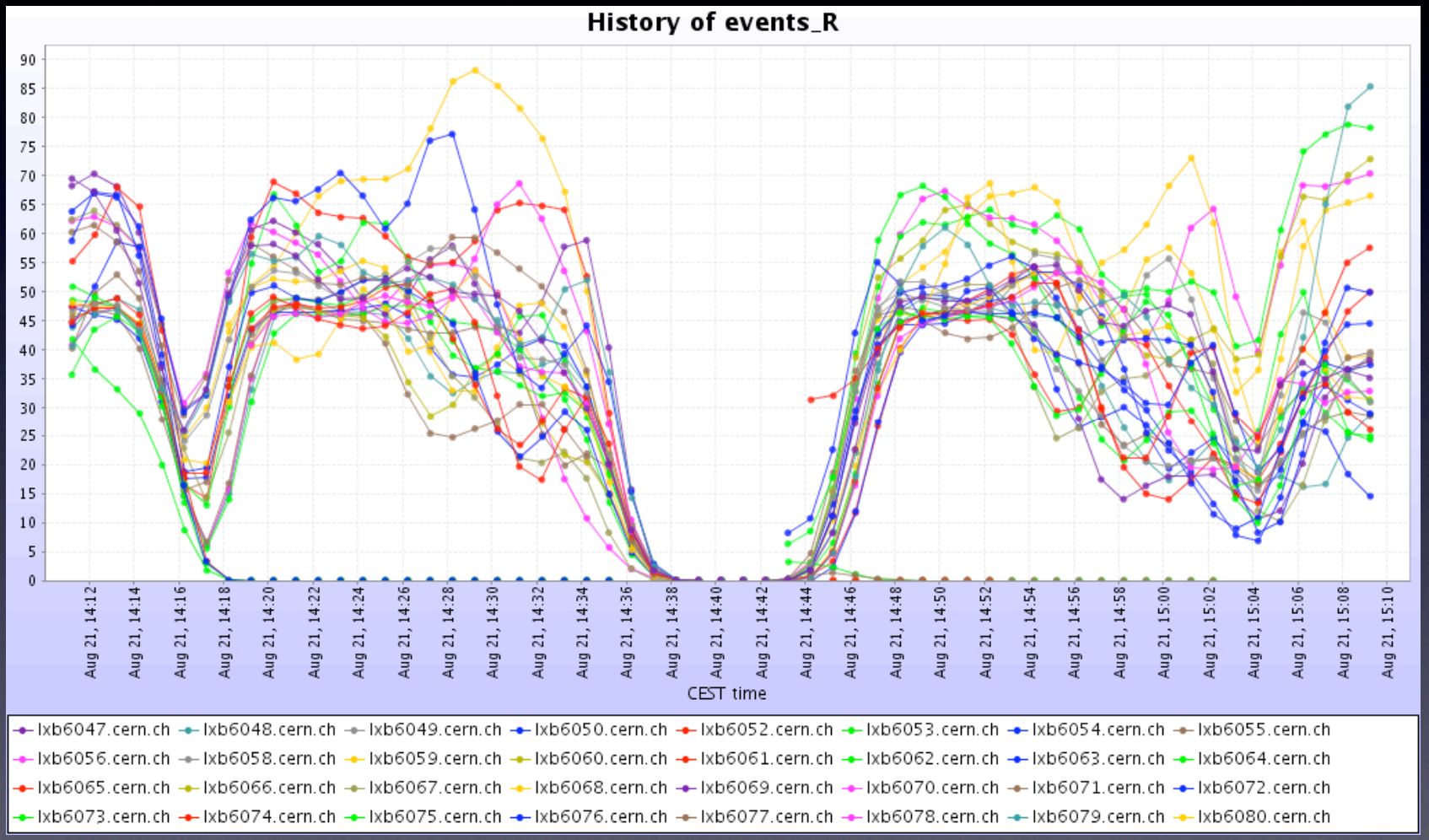
- MonALISA based monitoring
  - Each host reports to MonALISA
  - Each proofserv reports to MonALISA
- Internal monitoring
  - File access rate, packet generation time and latency, processing time, etc.
  - Produces a tree for further analysis

Host Monitoring



The same for CPU, memory, swap, network, …

Query Monitoring



The same for: CPU usage, cluster usage, memory, event rate, local/remote MB/s and files/s

Network Traffic

Traffic between the cluster machines (MB/sec) (last 0.5h average)																							
Machine	**6047**	**6048**	**6049**	**6050**	**6052**	**6053**	**6054**	**6055**	**6056**	**6057**	**6058**	**6059**	**6060**	**6061**	**6062**	**6063**	**6064**	**6065**	**6066**	**6067**	**6068**	**6069**	**607**
1. 6047	0	-	-	-	-	-	2.927	2.018	-	-	1.094	-	-	-	1.908	4.112	-	-	0.974	0.614	0	0	
2. 6048	-	9.406	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
3. 6049	-	-	8.678	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
4. 6050	-	-	-	6.692	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
5. 6052	-	-	-	-	3.913	-	1.454	-	-	-	-	3.084	-	0.317	0	0	-	0	-	-	0.985	4.447	
6. 6053	-	-	-	-	-	6.603	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
7. 6054	0	-	-	1.363	-	-	6.195	-	-	-	0	-	-	-	0	-	-	-	-	0	-	-	1.5
8. 6055	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
9. 6056	-	-	-	-	-	-	-	-	4.962	-	2.442	0.525	-	-	-	-	-	-	-	-	-	-	
10. 6057	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
11. 6058	1.164	-	-	-	0	-	-	-	2.531	-	0	0	-	-	-	-	1.103	-	0	-	-	-	
12. 6059	3.755	-	0.622	-	-	-	-	-	-	-	-	11.76	1.955	0	0.677	1.848	0	-	-	-	2.812	-	0.7
13. 6060	-	-	-	-	-	-	-	-	2.068	-	-	-	11.59	-	-	1.06	-	-	-	-	-	-	2.0
14. 6061	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
15. 6062	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
16. 6063	-	-	-	-	1.655	0.27	2.416	-	-	-	-	-	-	0	-	6.38	-	0	-	0	-	-	
17. 6064	-	-	-	-	-	1.123	-	2.822	-	-	-	-	1.621	-	0	-	3.117	-	0	0	-	-	0.5
18. 6065	0	-	-	-	3.52	3.165	-	0	-	-	0	-	-	-	-	-	3.034	0	1.579	-	0	-	
19. 6066	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Diagonal elements: local traffic (TFile), other elements: network traffic(TXNetFile)

Deployment and Testing

- Working intensively with ALICE to test performance and functionality of PROOF on the CERN CAF
- CMS has shown interest and wishes to test PROOF by the end of the year

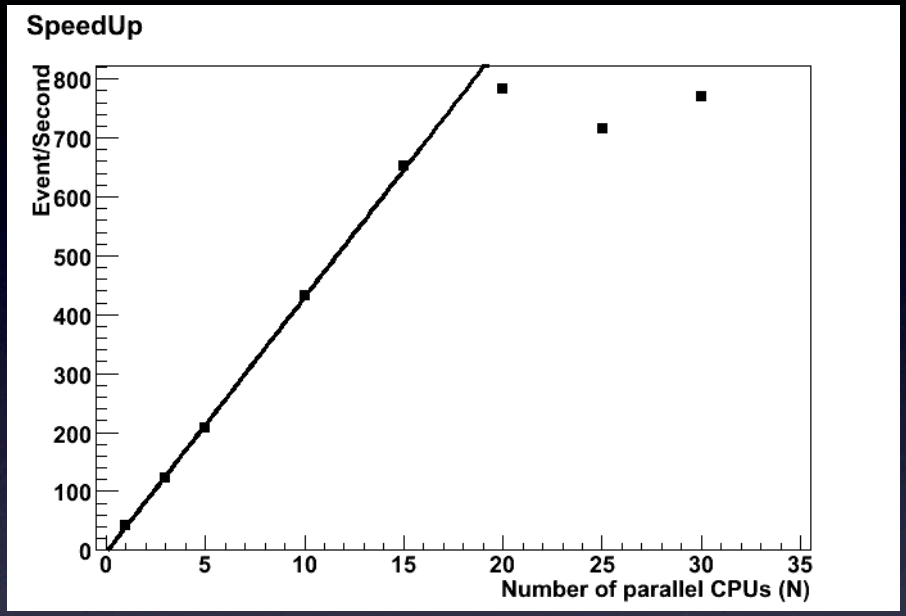
ALICE CAF Test Setup

- Since May evaluation of CAF test setup
  - 33 machines, 2 CPUs each, 200 GB disk
- Tests performed
  - Usability tests
  - Simple speedup plot
  - Evaluation of different query types
  - Evaluation of the system when running a combination of query types
- Work done for ALICE by Jan Fiete Grosse-Oetringhaus

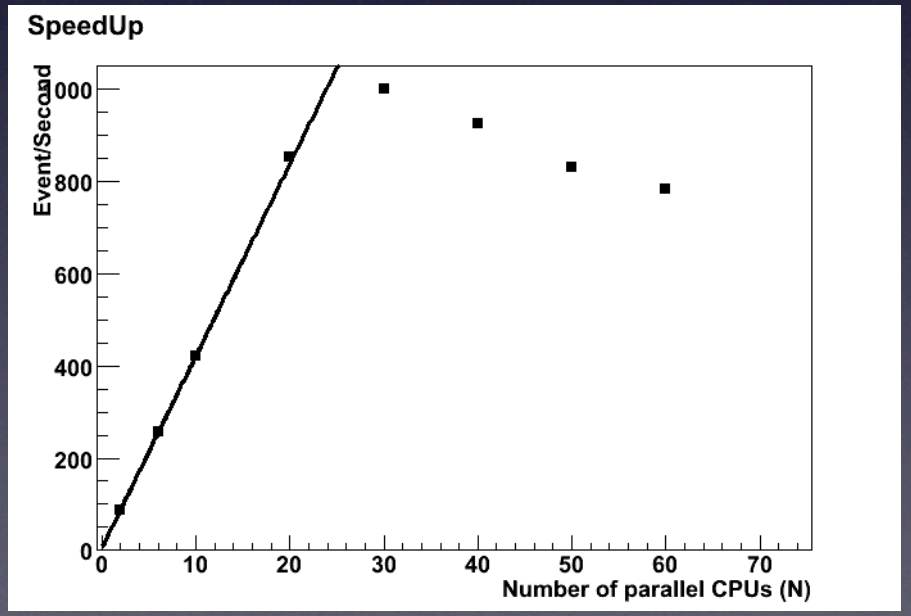
File Distribution

- The files have been distributed using xrootd
  - All were copied to the redirector machine that redistributed them over the cluster
  - xrootd tries to distribute the files evenly, but some nodes host more files than others (difference up to 50%)
  - We did not correct because this is a realistic scenario for analysis
  - For each query we selected files at random between those available
- PROOF favors local files over remote files

Simple Speedup

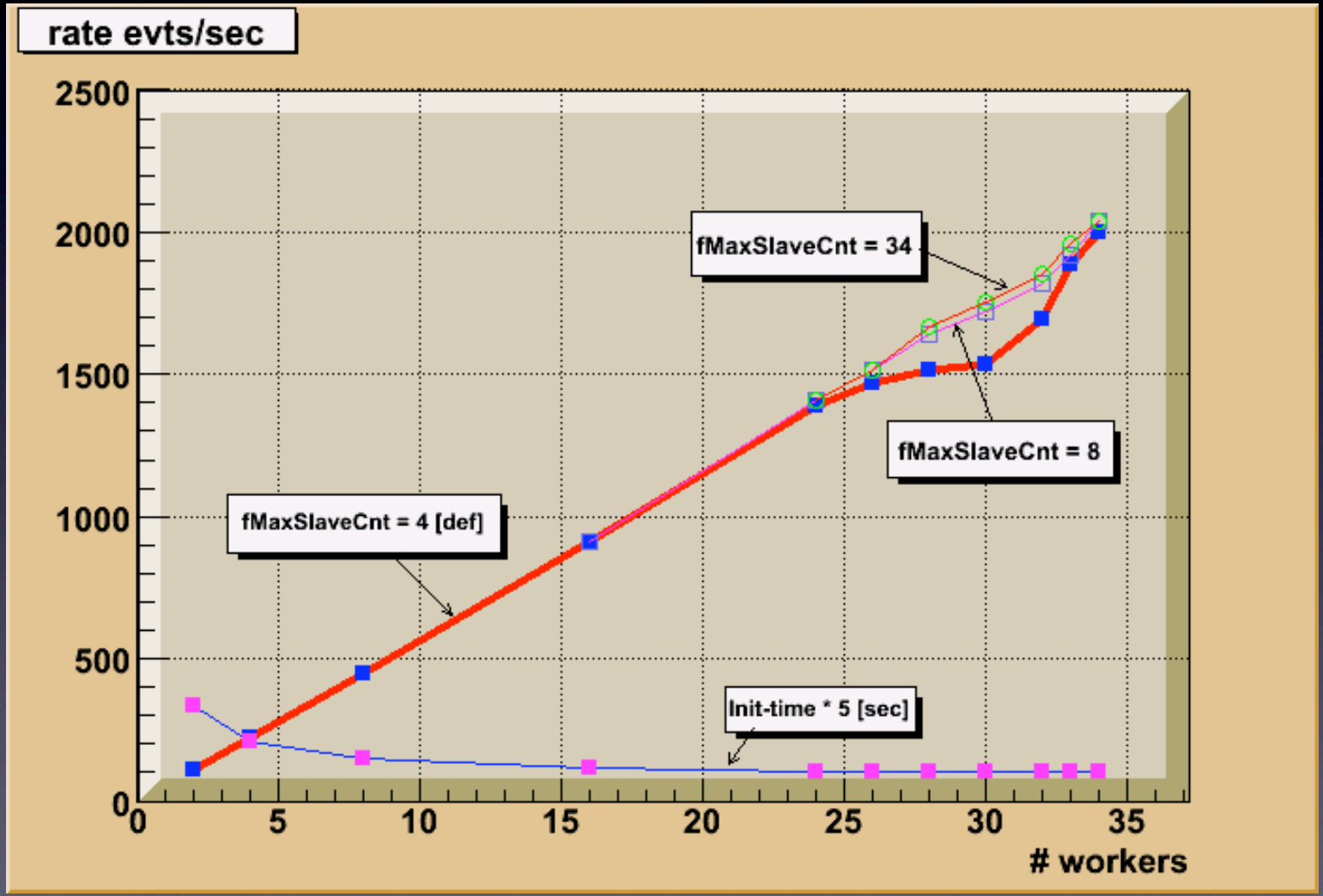


- One query to an empty CAF
- Each query processes at least 10min to minimize overhead
- Different data files used per query to avoid caching



- Breakdown in parallelism was initially puzzling

Understanding Speedup



Query Types

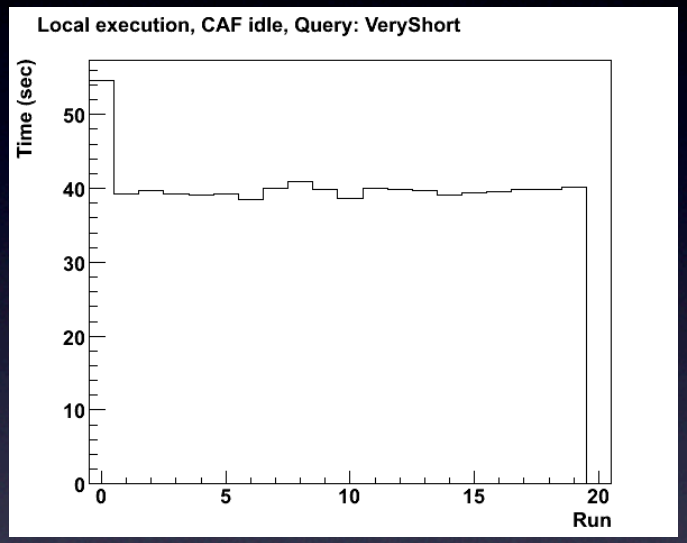
Type	# files	# evt	processed data (GB)	avg. time * (s)	I/O rate (MB/s)	submission interval (s)
VeryShort	20	2K	0.4	9 ± 1	44.4	30 ± 15
Short	20	40K	8	150 ± 10	53.3	120 ± 30
Medium	150	300K	60	1380 ± 60	43.5	300 ± 120
Long	500	1M	200	4500 ± 200	44.4	600 ± 120

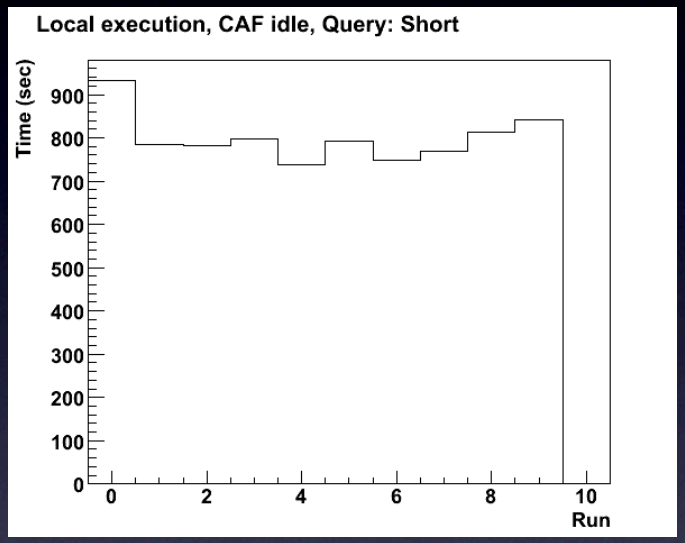
* Using PROOF, 10 users, 10 parallel workers each

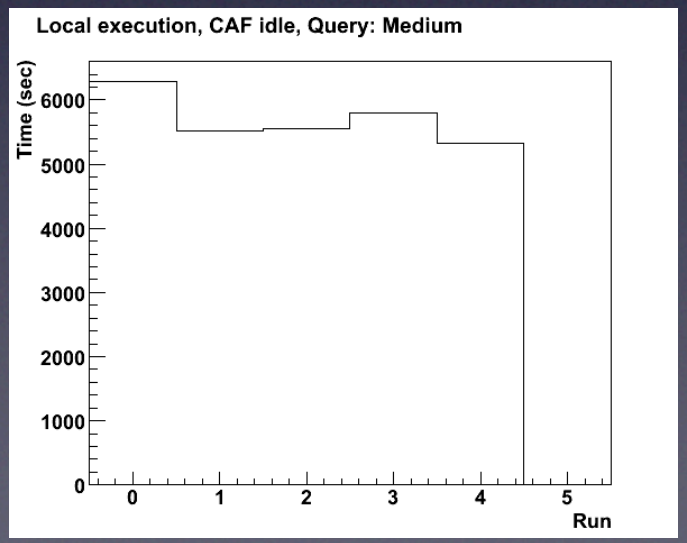
Query Type Cocktail

- 4 different query types
  - 20% very short queries
  - 40% short queries
  - 20% medium queries
  - 20% long queries
- User mix
  - 33 nodes
  - 10 users, 10 or 30 workers/user, max ave. speedup = 6.6
  - 5 users, 20 workers/user
  - 15 users, 7 workers/user

Time Evolution Single Worker

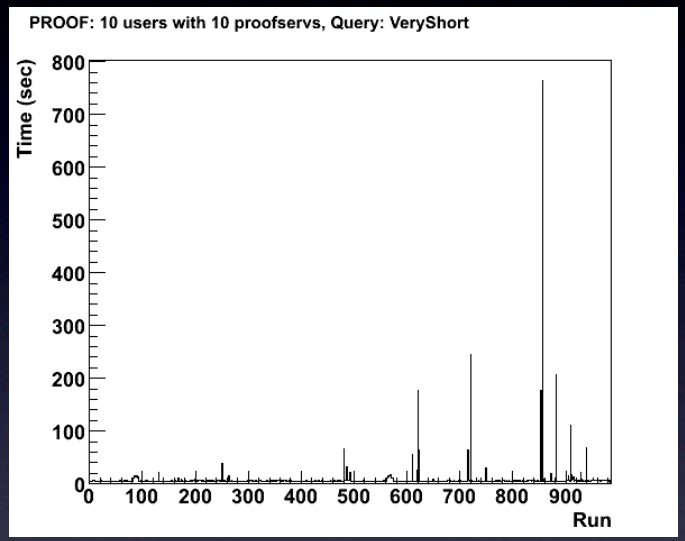


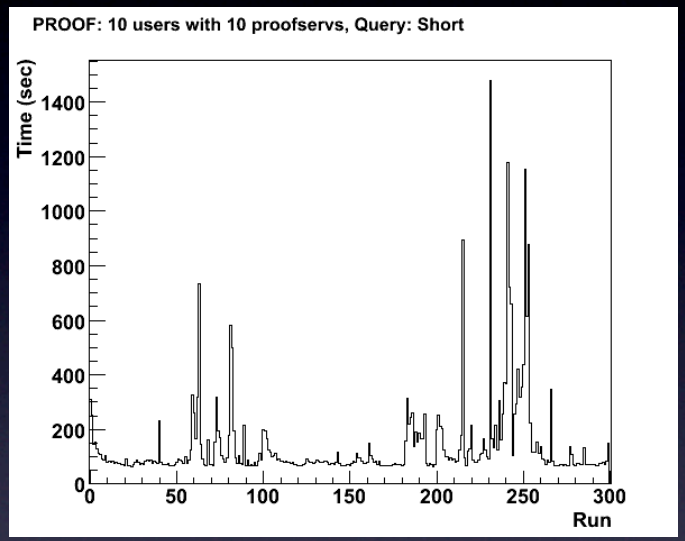


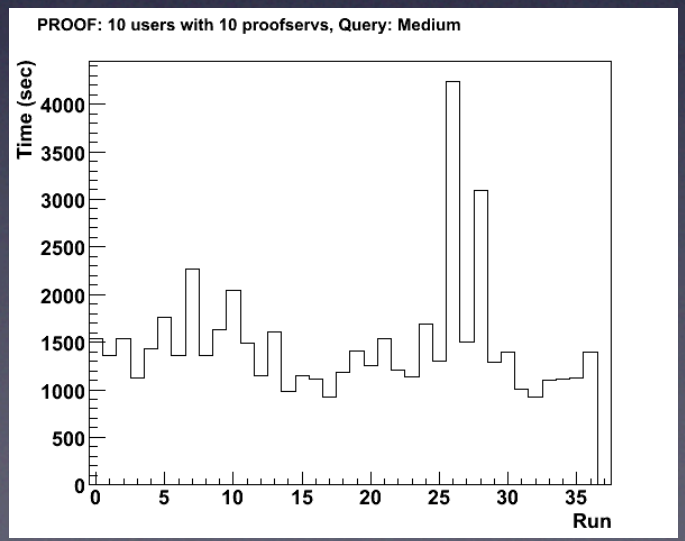


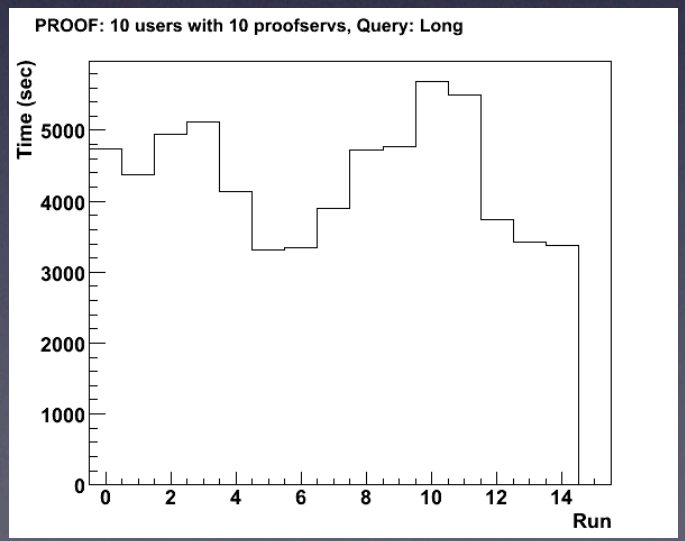
- Even distribution of files, i.e. 1/33 are local
- Second query faster because the files are cached in the machines serving the files

Time Evolution Multiple Workers

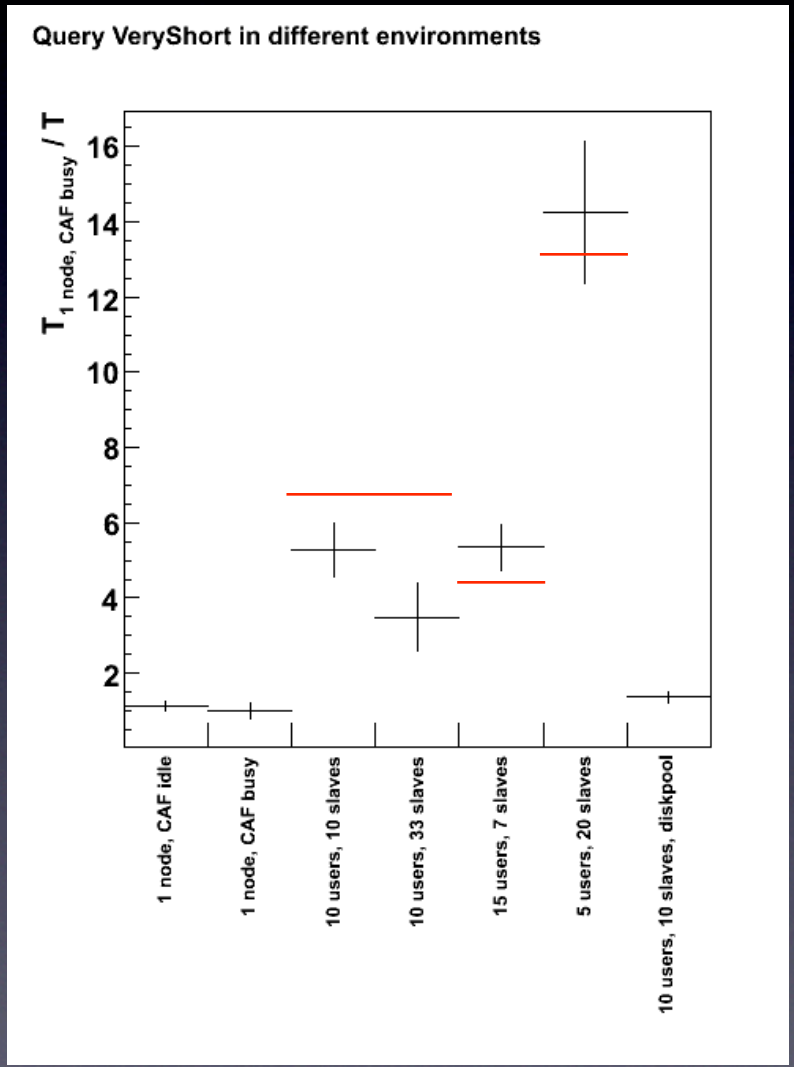


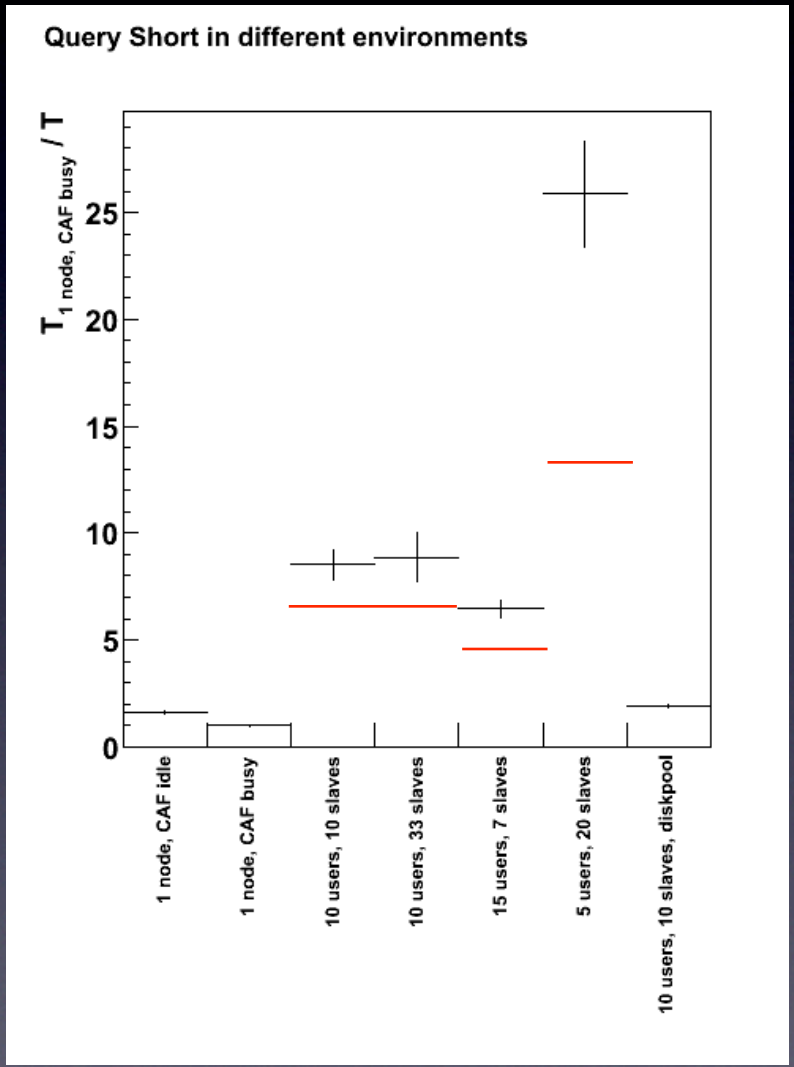






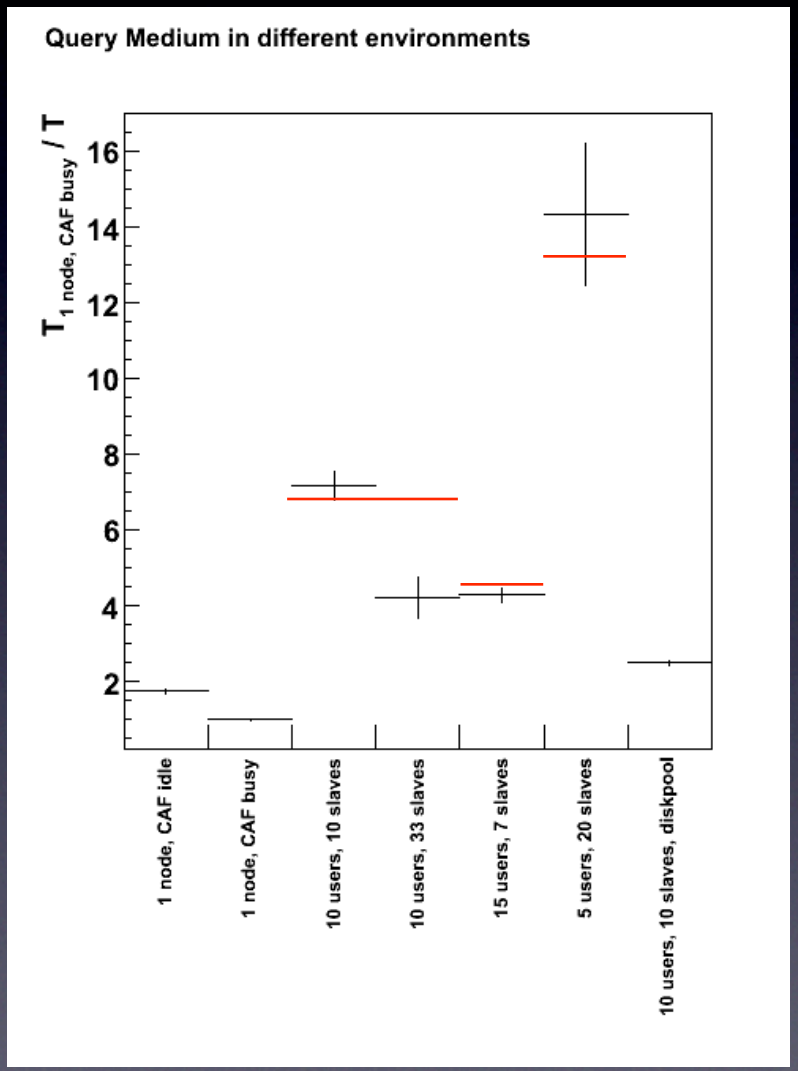
Relative Speedup

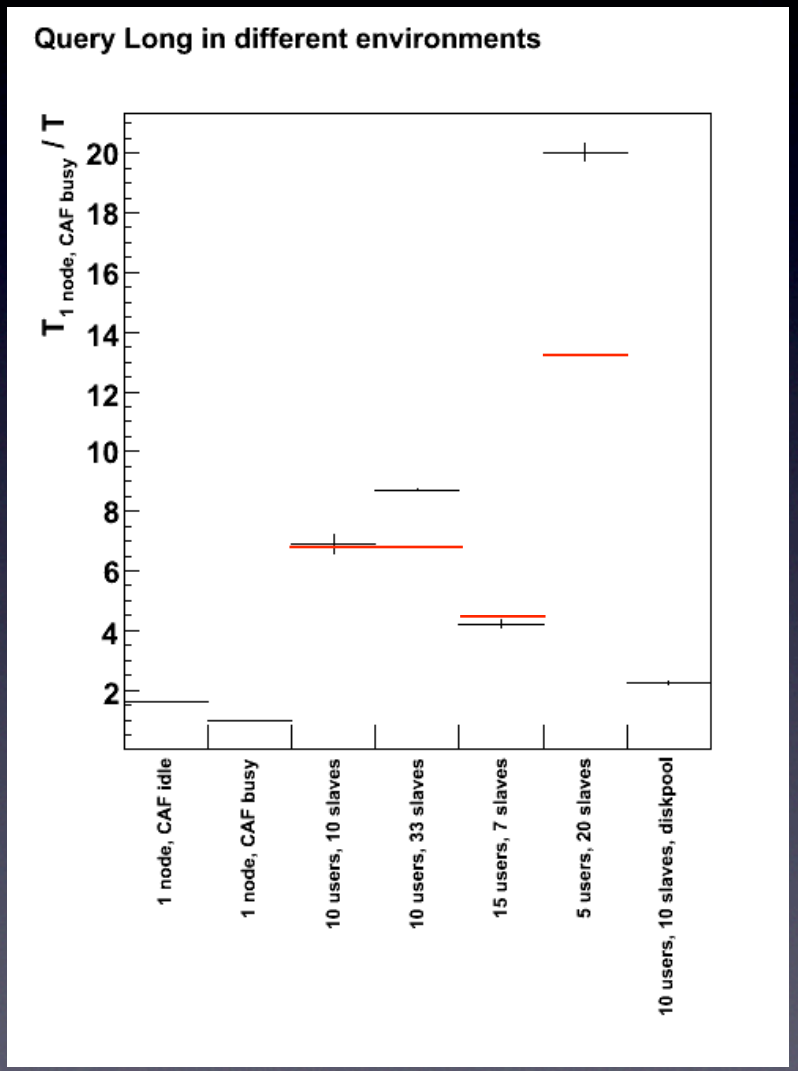




Average expected speedup

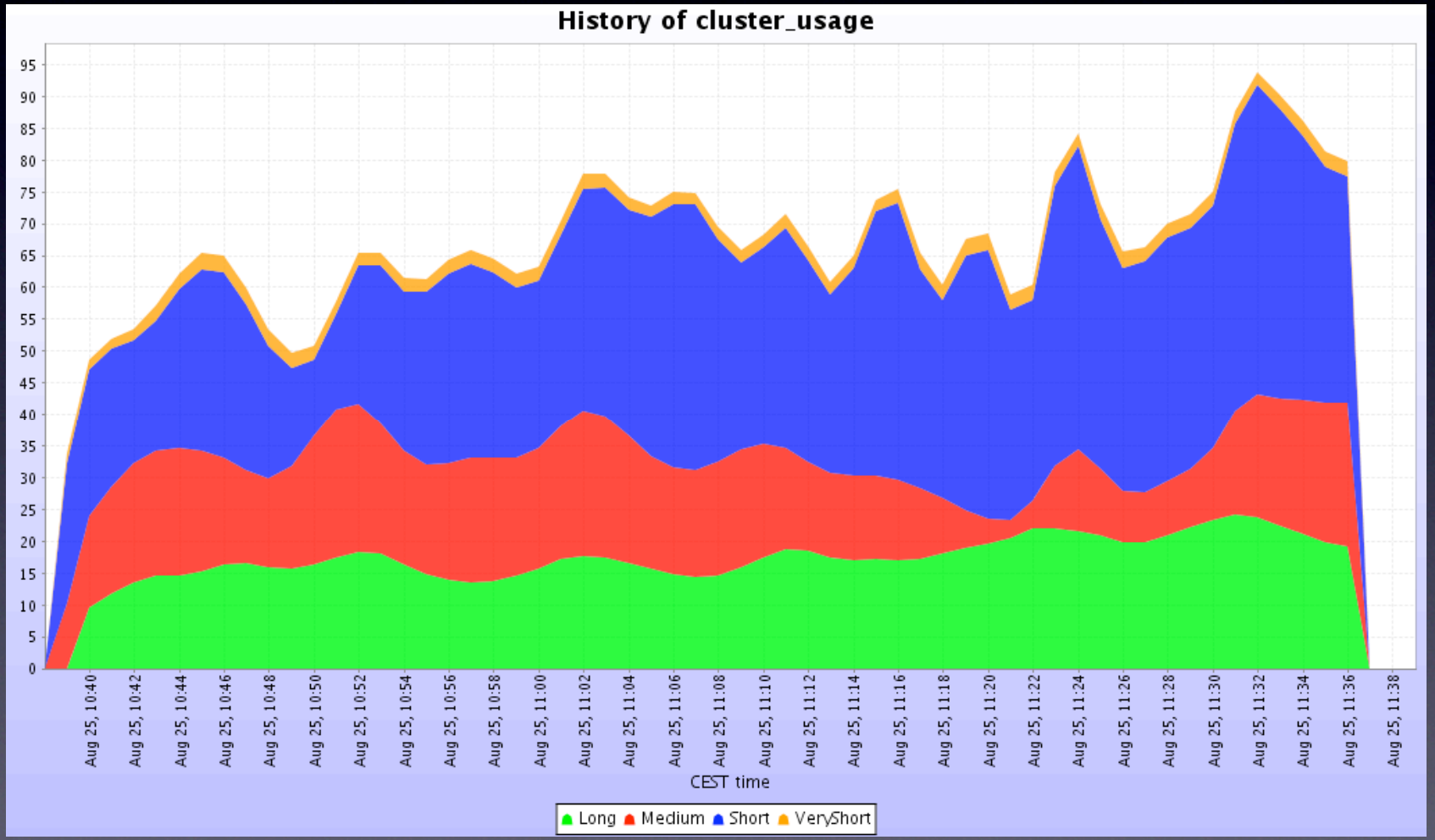
Relative Speedup





Average expected speedup

Cluster Efficiency



Development Plans

- Improve packetizer
  - Constant time per packet, smaller packets at end of query
- Dynamic cluster configuration
  - Come and go of worker nodes
- Improve handling of error conditions
- Support multiple server versions
- Data access optimization
- Multi-user scheduling
- GUI improvements
- Generic processing
- Testing and consolidation

Data Access Optimizations

- Low latency data access is essential
- Reduce file opening overhead by using asynchronous open
- Reduce data access latency by using:
  - Tree branch read-ahead and caching
  - Asynchronous reading
  - Asynchronous data decompression

Multi-User Scheduling

- Scheduler is needed to control the use of available resources in multi-user environments
- Decisions taken per query based on the following metric:
  - Overall cluster load
  - Resources needed by the query
  - User history and priorities
- Requires dynamic cluster reconfiguration
- Generic interface to external schedulers planned (Condor, LSF, ...)

Conclusions

- PROOF promises to become a powerful tool for the efficient analysis of large data sets in the era of large clusters and multi-core CPUs
- Exciting development plans to increase the efficiency of the system and improve the user experience
- First results in the ALICE environment look good, first users will be exposed to the system soon, ALICE will follow the developments aggressively