The

CERN Analysis Facility

running for

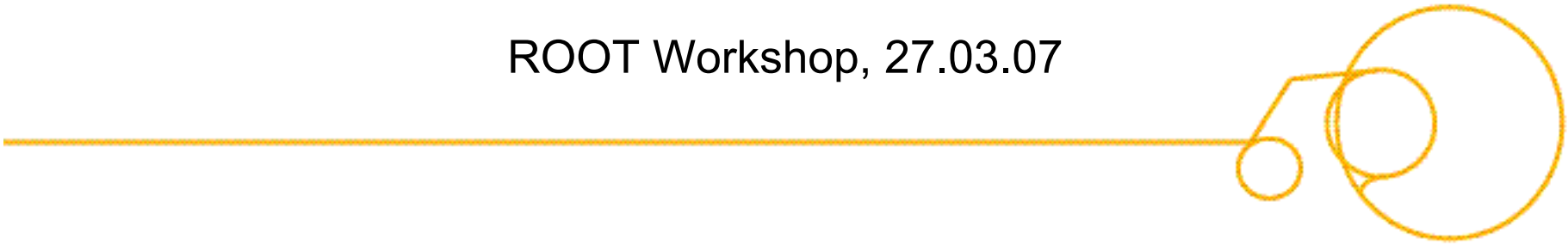
ALICE

enabled by

PROOF technology

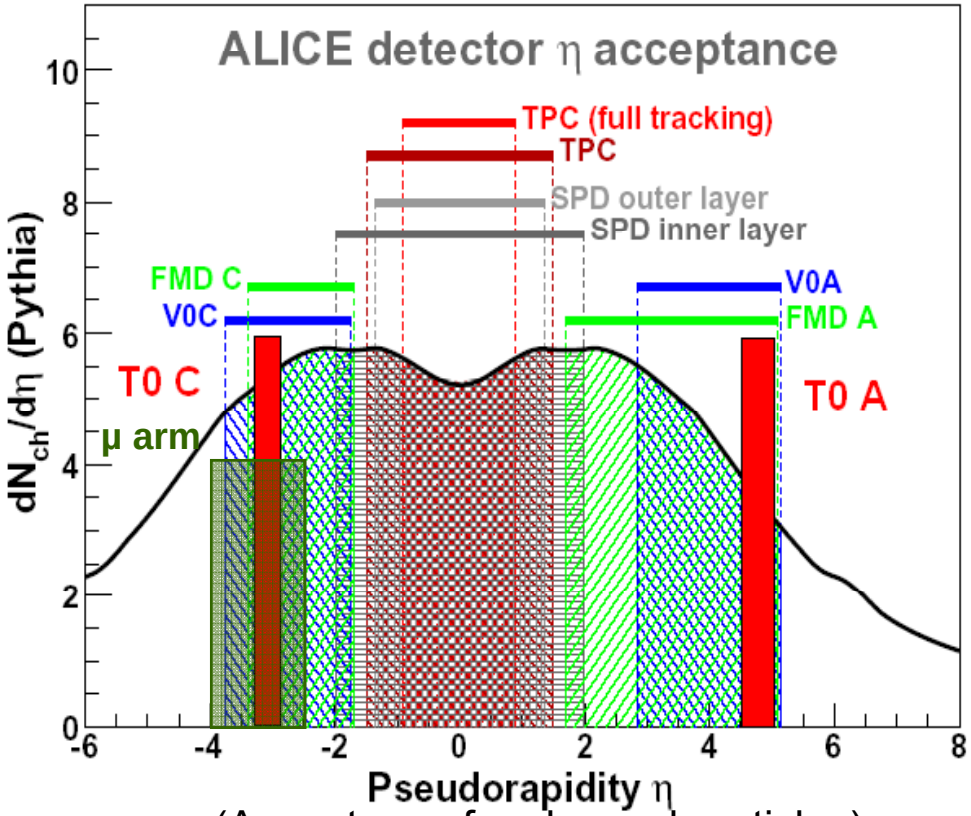
Jan Fiete Grosse-Oetringhaus, CERN PH/ALICE

ROOT Workshop, 27.03.07

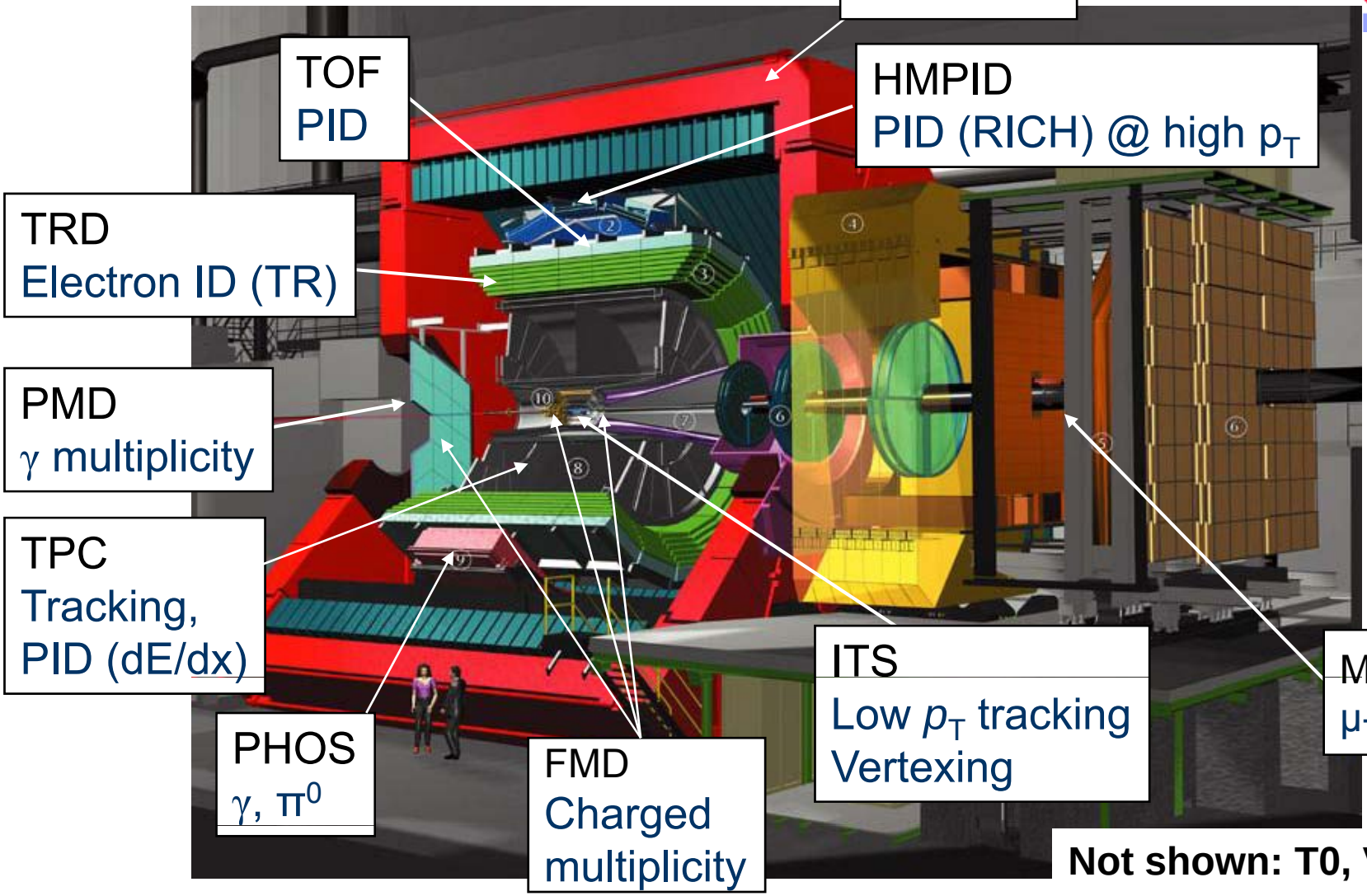


ALICE

- Heavy-ion experiment
- Designed for $dN_{ch}/d\eta \mid \eta = 0$ up to 8000 (not expected anymore)
- Reconstructs and identifies particles over broad momentum range (~ 100 MeV/c – ~ 100 GeV/c)
- Reconstruct short-lived particles (hyperons, B and D mesons) → decay vertices

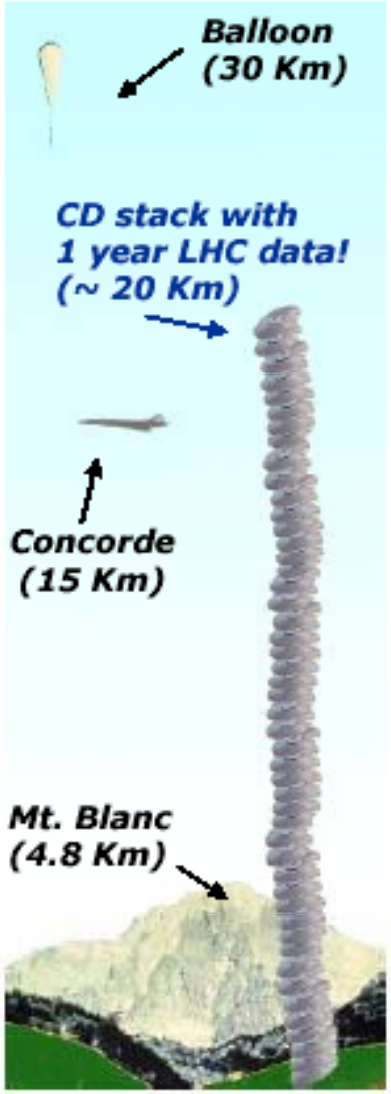


ALICE at Point 2



ALICE's Data production

- Pb+Pb collision rate: 4.000 Hz
- Storing („DAQ“) rate: 100 Hz (“Event”)
- Event size: 12,5 MB
- Bandwidth: 1,25 GB/s
- Raw data, reconstructed events, associated Monte Carlo production: 6 PB/year



CERN Analysis Facility

- The **C**ERN **A**nalysis **F**acility (CAF) will run PROOF for ALICE for tasks that run on a short time scale
  - Prompt analysis of pp data
  - Pilot analysis of PbPb data
  - Calibration & Alignment
- Additionally to using the Grid
  - Massive execution of jobs vs. fast response time
- Available to the whole collaboration but the number of users will be limited for efficiency reasons
- Design goals
  - 500 CPUs
  - 200 TB of selected data locally available

CAF Schema

Experiment

Disk Buffer

Tier-1 data export

Tape storage

Sub set (moderated)

Staging on request of physics working groups or single users

CAF computing cluster

Proof node | local disk

Proof node | local disk

Proof node | local disk

...

Proof node | local disk

Proof node | local disk

Proof node | local disk

Automatic Staging

- Transparent by opening/preparing a file in a given path, e.g. /castor (CASTOR), /alice (AliEn)
- Prepare requests (olb.prep) and stage (open) requests (oss.stagecmd) instantiate same Perl script
  - Front-End: Registers stage request
  - Back-End
    - Checks access privileges
    - Triggers migration from tape (CASTOR, AliEn)
    - Copies files, notifies xrootd
  - Garbage collector: Cleans up following policy file with low/high watermarks for given path patterns

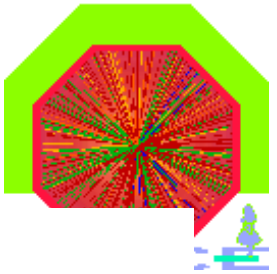
CAF Test Setup

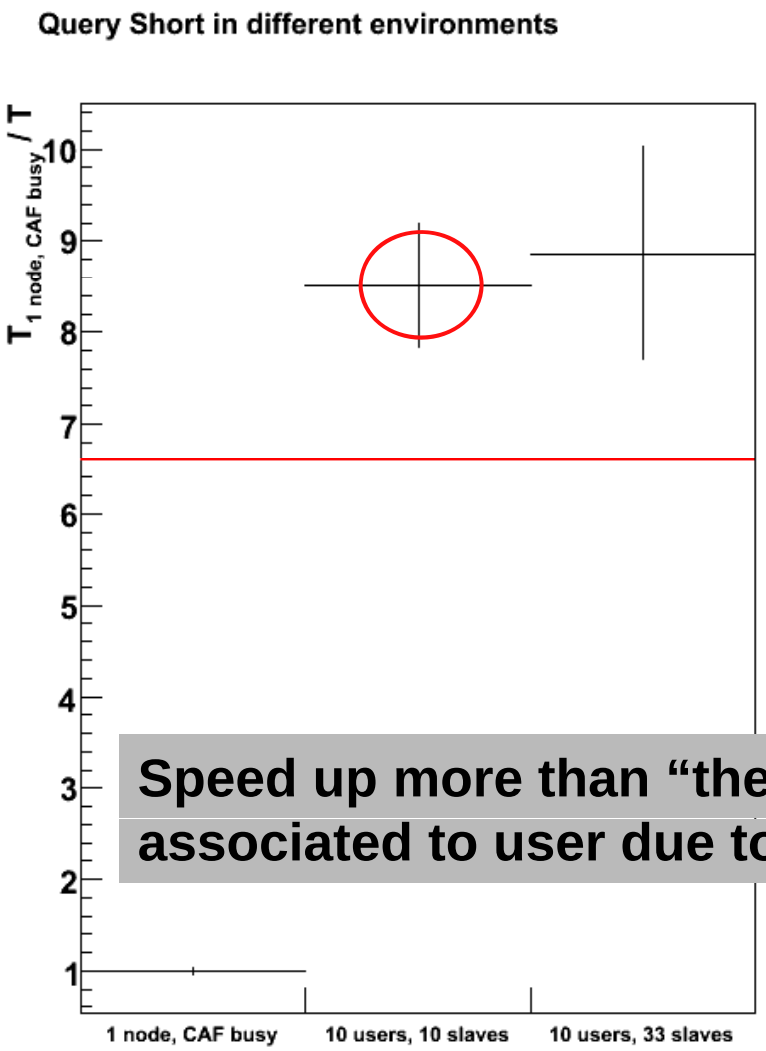
- Test setup since May 2006
  - 40 machines, 2 CPUs each, 200 GB disk
  - 5 as development partition, 35 as production partition
- Machines are a xrootd disk pool
  - Fraction of data of Physics Data Challenge ’06 distributed (~ 1 M events)
  - Automatic staging from AliEn/CASTOR in place, ~ 3M events staged on user request
- Tests performed
  - Usability tests (83 submitted bugs, 64 fixed, 19 open)
  - Speedup tests
  - Evaluation of the system when running a combination of query types
  - “Break down” test
- Integration with ALICE’s analysis framework (AliROOT)

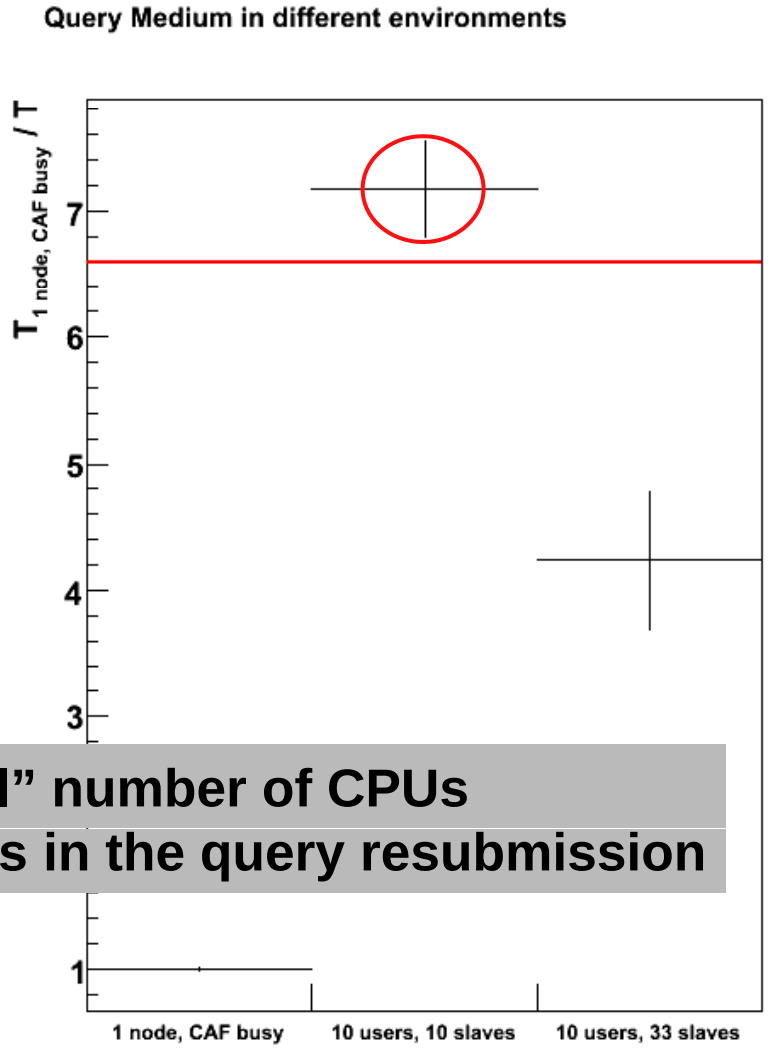
Query Type Cocktail

- A realistic stress test consists of different users that submit different types of queries
- 4 different query types
  - 20% very short queries
  - 40% short queries
  - 20% medium queries
  - 20% long queries
- User mix
  - 33 nodes available for the test
  - Maximum average speedup for 10 users = 6.6 (33 nodes = 66 CPUs)
  - Pauses between query resubmission

Relative Speedup
(preliminary)



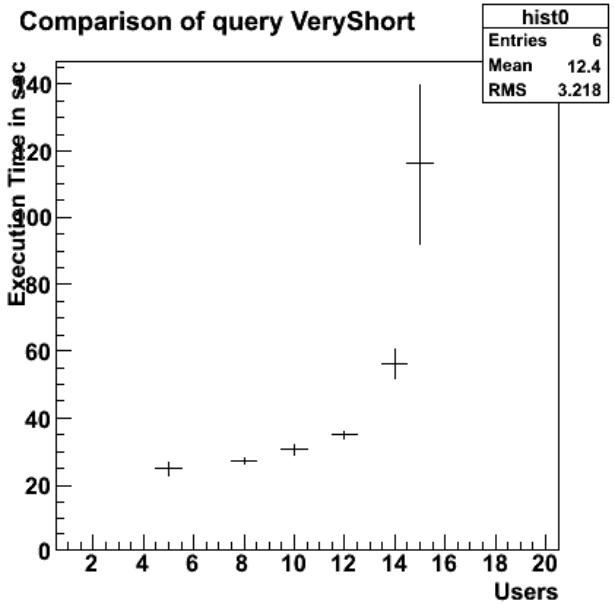


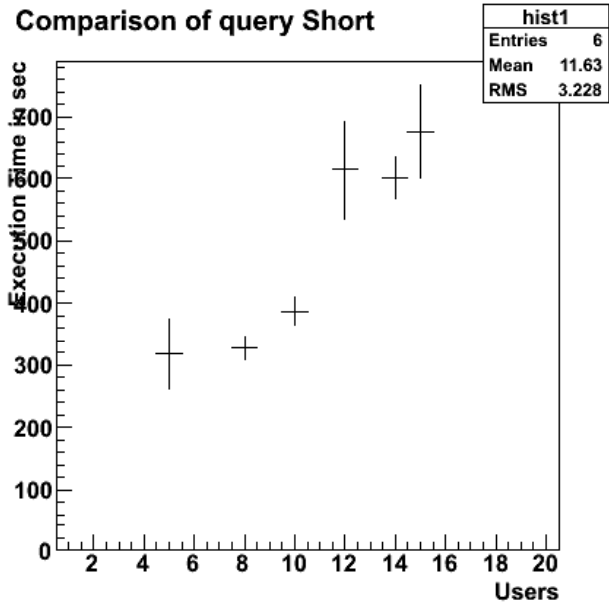


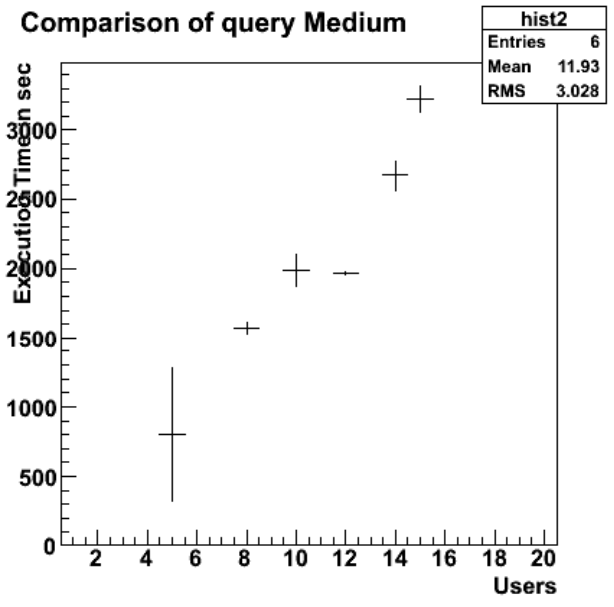
Speed up more than "theoretical" number of CPUs associated to user due to pauses in the query resubmission

“Break down” Test
(preliminary)

- Query cocktail run with different number of parallel users
  → Compare average execution time

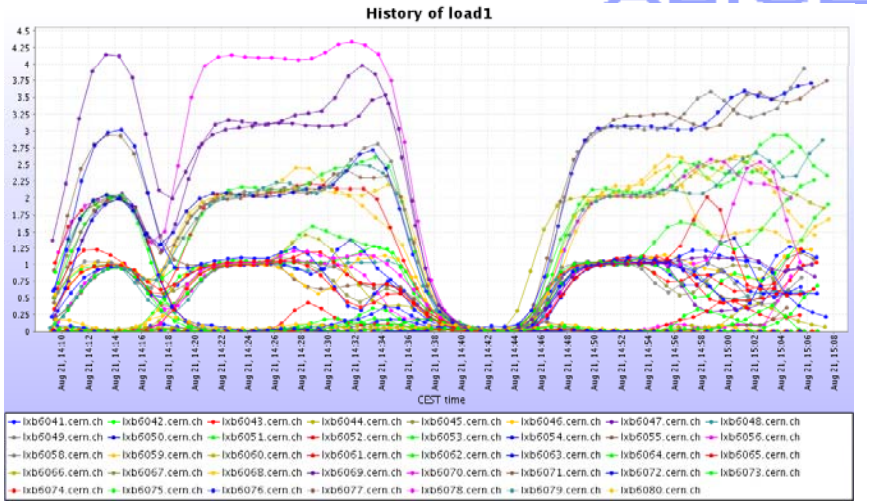






Evaluation of the System Behavior

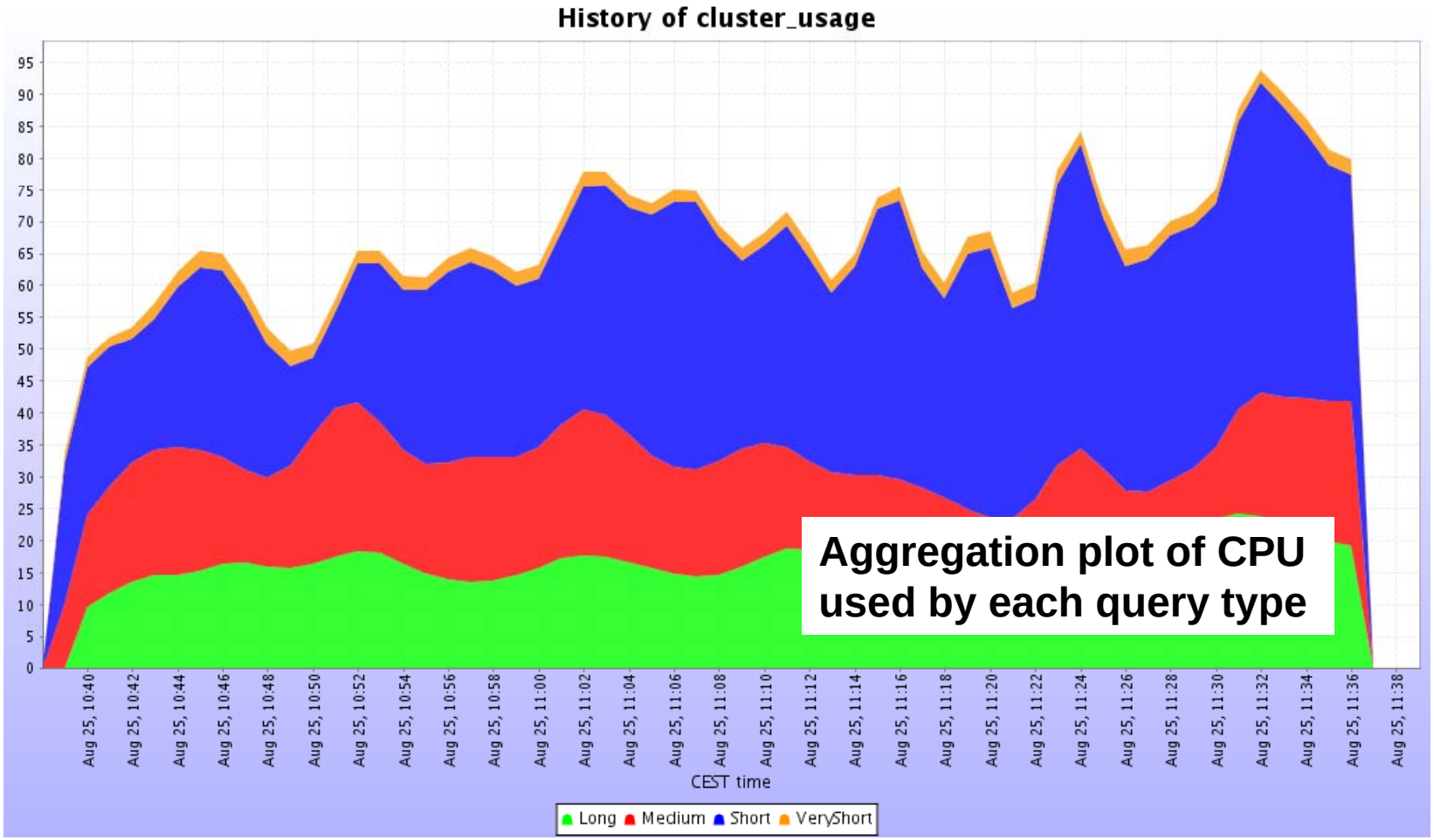
- MonALISA monitoring
  - Each host reports
  - Each slave reports
- Host
  - CPU, memory, swap, network
- Query (sum, per query type)
  - CPU, memory, event rate, file rate, IO vs. network rate
- pcalimonitor.cern.ch:8889
  - Click on CAF monitoring



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	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Overall Usage of the Cluster

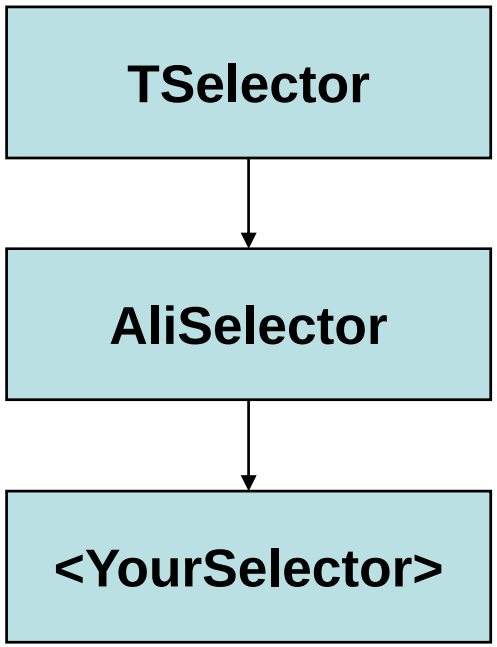


ALICE Software Framework

- AliROOT
  - Based on ROOT (ROOT + libraries = AliROOT)
- AliROOT + PROOF → How to load libraries?
- Reconstruction output
  - Event Summary Data (ESD)
  - Only requires one library (PROOF: one package), see next slide
- Physics working group libraries can be converted automatically to PROOF packages
  - Make target added to AliROOT Makefile

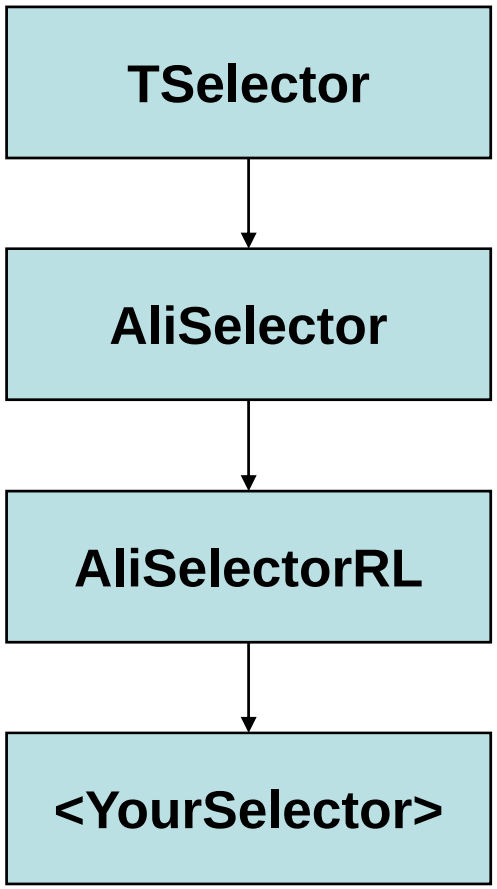
Accessing Event Summary Data (ESD)

- Reconstruction output: ESD
- libESD.so required
- ESD.par package uploaded by standard PROOF functionality
- Selector derives from AliSelector
- Access to data by member: fESD



Using Full AliRoot Framework

- Access to Kinematics, Clusters, etc. requires access to the RunLoader
- (Nearly) full AliRoot needs to be loaded
- AliRoot is manually deployed on the CAF system (all nodes)
- Enabled by 3 line macro
  - Sets environment variables
  - Loading libraries
  - Effectively converts ROOT instance running on proof slave into AliROOT
- Selector derives from AliSelectorRL
  - GetStack(), GetRunLoader(), GetHeader()



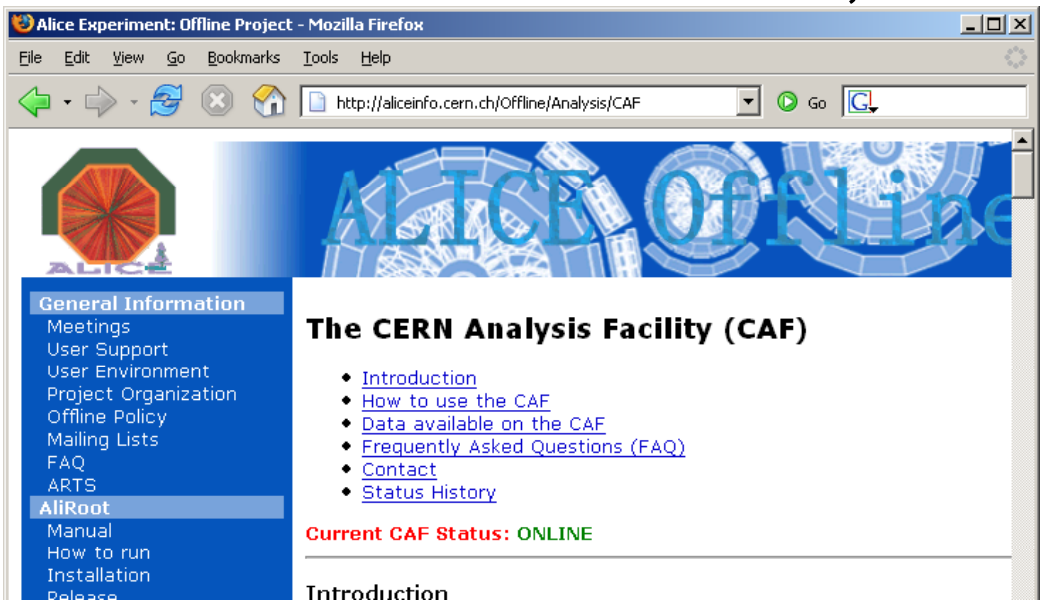
New Analysis Framework

- Framework to combine several analyses
  → “Analysis train”
- Benefit from having one event in memory
- Analyses appear as AliAnalysisTask (TTask)
  - Uses TTask hierarchy model
  - Extended with input, output slots
- Analysis Manager that steers is a TSelector
- AliAnalysisTasks need to be distributed to PROOF cluster
  - Frequent changes → .par file inconvenient
  - New PROOF functionality: TProof::Load(“class”)

Demo

More Information

- http://aliceinfo.cern.ch/Offline/Analysis/CAF
- Monthly tutorials for ALICE members at CERN, contact Yves Schutz
- Slides of the last tutorial (AliRoot, PROOF, AliEn)
  http://indico.cern.ch/conferenceDisplay.py?confId=13375



- Server installation at other institutes

  http://root.cern.ch/twiki/bin/view/ROOT/ProofInstallation

Backup

Outlook

- Priority management
- Evaluation of automatic staging
- Gradual increase of cluster size

Automatic AliEn staging

- ~ 1M events (PDC06 MB) already distributed
  - Get file list at http://aliceinfo.cern.ch/Offline/Analysis/CAF#data
- Triggering of any file that exists in AliEn can be triggered
  - By simply asking for the file TFile::Open("root://lxb6046.cern.ch//alice/sim/.../AliESDs.root")
  - The file will be picked up from AliEn, copied to the CAF and stays there (until it is not used anymore and the CAF runs out of disk space)
  - Retrieving of a file can take several hours (known from AliEn), due to tape-disk migration in CASTOR
  - All needed files can be requested at the same time
  - More information + detailed instructions http://aliceinfo.cern.ch/Offline/Analysis/CAF#data

Query Types

- A realistic stress test consists of different users that submit different types of queries

Name	# files	# evts	processed data	avg. time*	Submission Interval
VeryShort	20	2K	0.4 GB	9 ± 1 s	30 ± 15 s
Short	20	40K	8 GB	150 ± 10 s	120 ± 30 s
Medium	150	300K	60 GB	1,380 ± 60 s	300 ± 120 s
Long	500	1M	200 GB	4,500 ± 200 s	600 ± 120 s

*run in PROOF, 10 users, 10 PROOFservs each