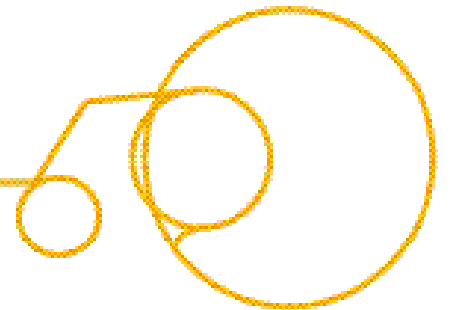


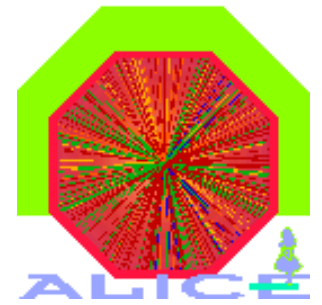
The CERN Analysis Facility - A PROOF Cluster for Prompt Physics Analysis

Jan Fiete Grosse-Oetringhaus, CERN PH/ALICE

DESY Computing Seminar

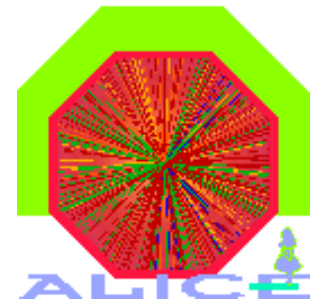


Content



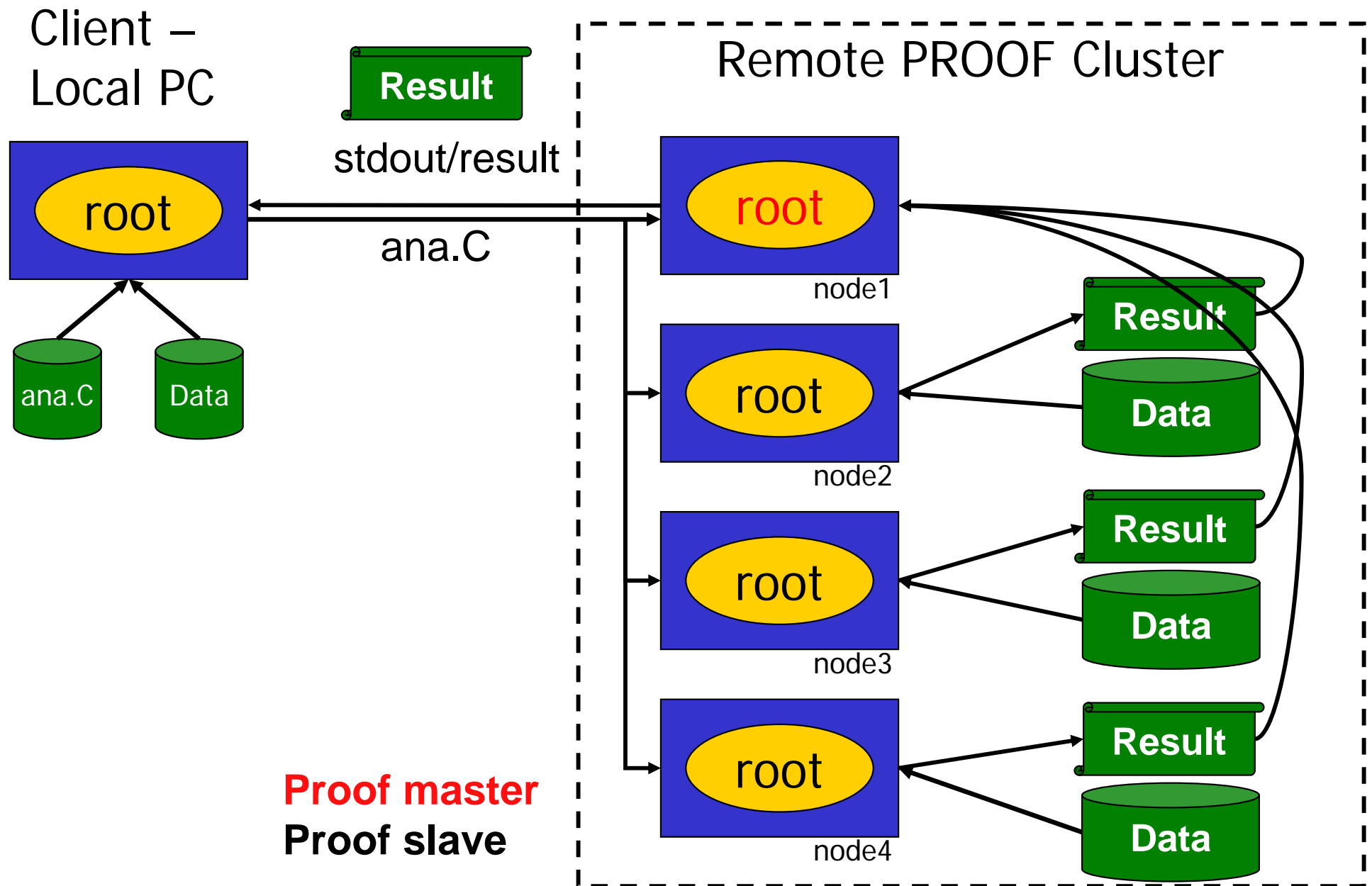
- PROOF system
 - xrootd
- ALICE
- PROOF setup for ALICE (CAF)
- ALICE's experience with PROOF
- Experience from other sites
 - PROOF workshop
- Demo

PROOF

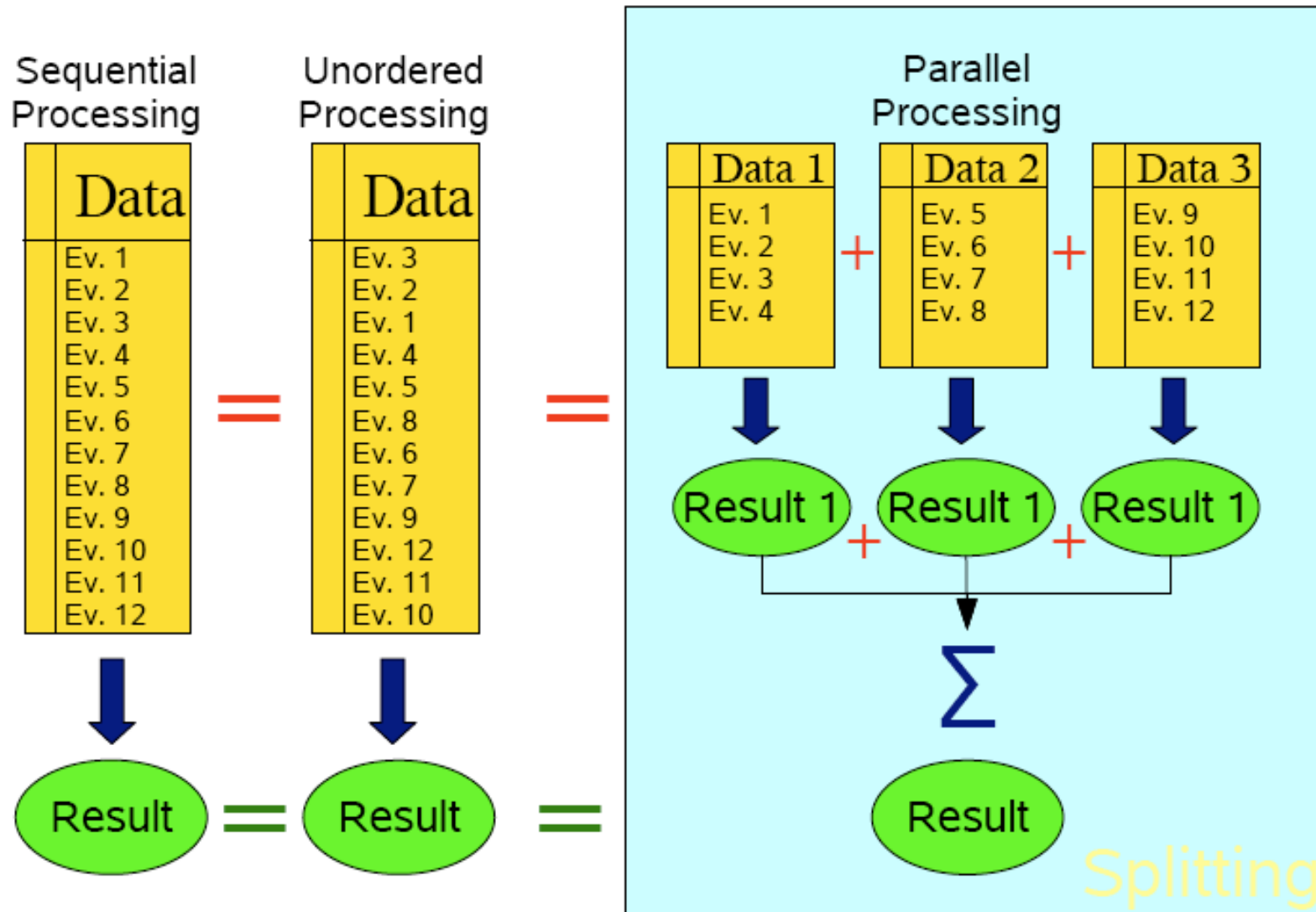
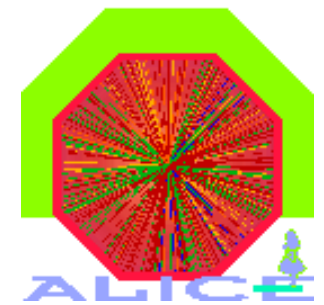


- Parallel ROOT Facility
- Interactive parallel analysis on a local cluster
 - Parallel processing of (local) data (trivial parallelism)
 - Output handling with direct visualization
 - **Not** a batch system
- PROOF itself is not related to Grid
 - Can access Grid files
- The usage of PROOF is transparent
 - The same code can be run locally and in a PROOF system (certain rules have to be followed)
- PROOF is part of ROOT

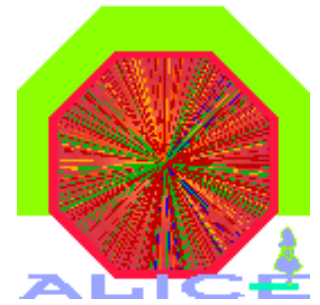
PROOF Schema



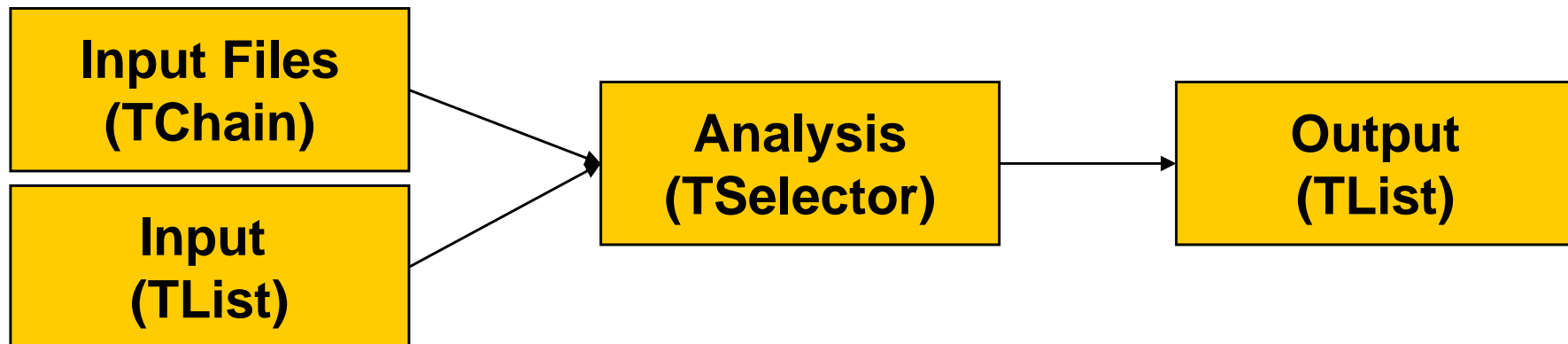
Event based (trivial) Parallelism

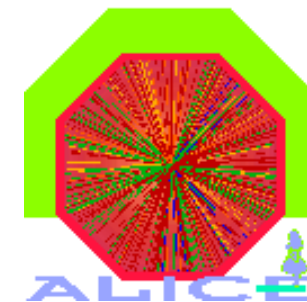


How to use PROOF



- Files to be analyzed are listed in a chain (→ TTree/TChain)
- Analysis written as a selector (→ TSelector)
- Input/Output is sent using dedicated lists
- If additional libraries are needed, these have to be distributed as packages ("par" = PROOF archive)





How to use PROOF (2)

- Classes derived from TSelector can run locally and in PROOF

– **Begin()** once on your client

– **SlaveBegin()** once on each slave

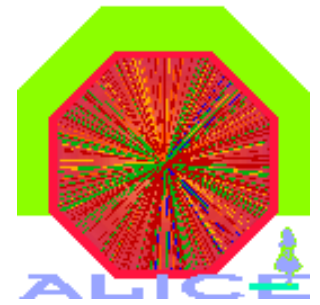
– **Init(TTree* tree)** for each tree

– **Process(Long64_t entry)** for each event

– **SlaveTerminate()**

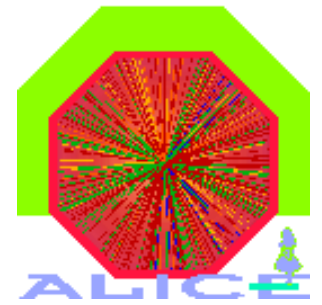
– **Terminate()**

PROOF History

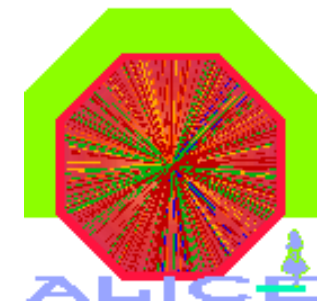


- First prototype in 1997
- Development temporarily at MIT by the Phobos group
- Development continued at CERN
- ALICE triggered many developments to make PROOF a production system
 - TProof::Reset that resets a user environment and kills remaining sessions
 - Before a intervention by an administrator was necessary who had to restart the whole cluster
 - New packetizer TAdaptivePacketizer (performs better on loaded clusters)
 - Possibility to selector ROOT version on startup
 - Very convenient for testing of new versions before migration
 - Receiving of log files of crashed sessions
- Manpower situation unclear! Contract of main developer is not being continued

xrootd

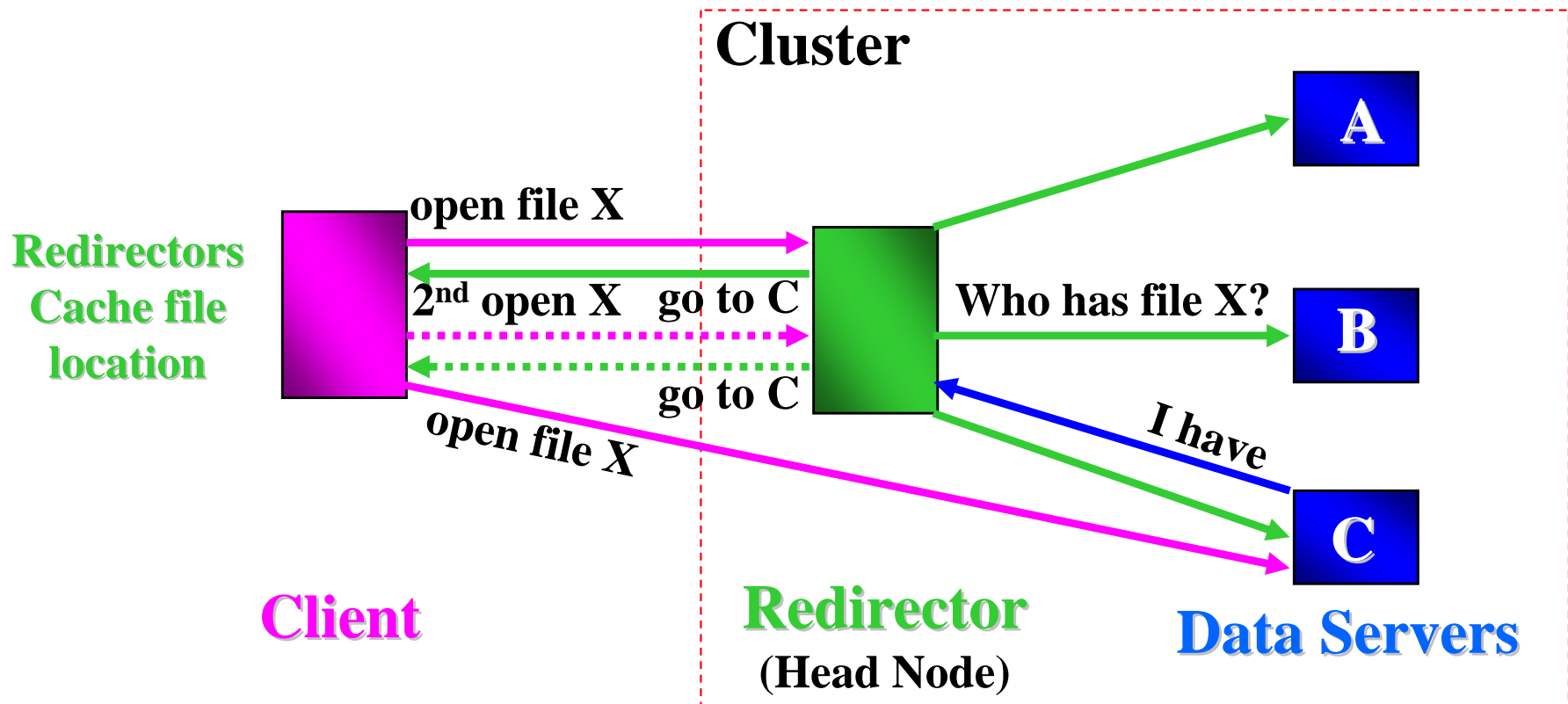


- Today's experiments produce PBs of data
 - Analysis needs access to this data
- xrootd is a file serving program
- Requirements
 - Reliable and fast
 - Scalable
 - Fault tolerant
- Developed at SLAC
 - Andy Hanushevsky (core development)
 - <http://xrootd.slac.stanford.edu>
- xrootd and PROOF
 - PROOF runs as a plugin to xrootd (xproofd)
 - xrootd can be used to serve files in a PROOF setup

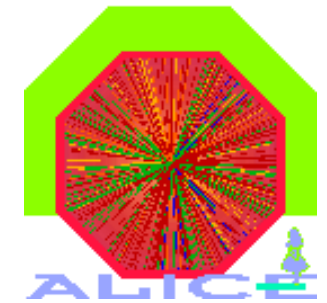


High-level architecture

- Data servers, that serve files
- Redirectors, that act as head node, redirect to the actual location of the file



The ALICE Experiment



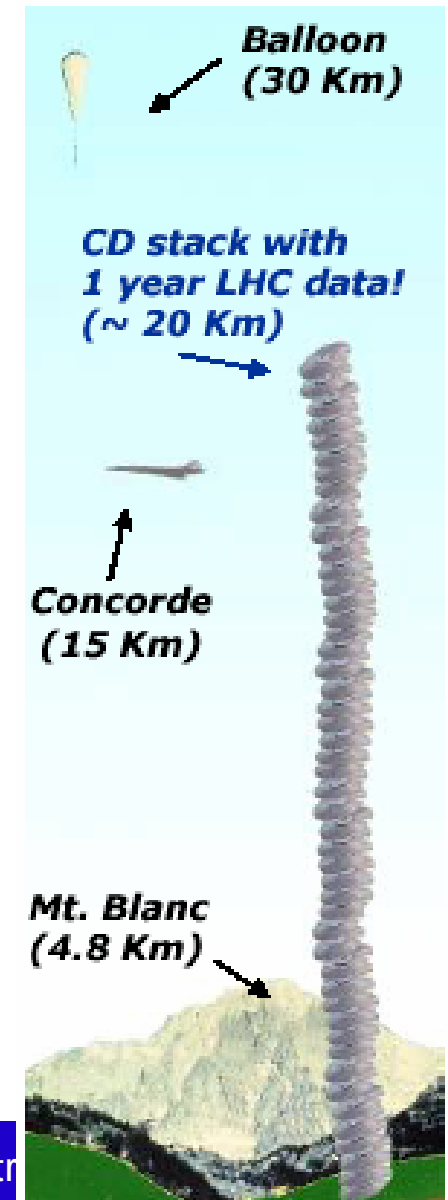
- ALICE (A Large Ion Collider Experiment) is one of the five LHC experiments currently being built at CERN
- LHC is a particle accelerator with a circumference of 27 km
- ALICE will study strongly interacting matter and the transition to the quark-gluon plasma



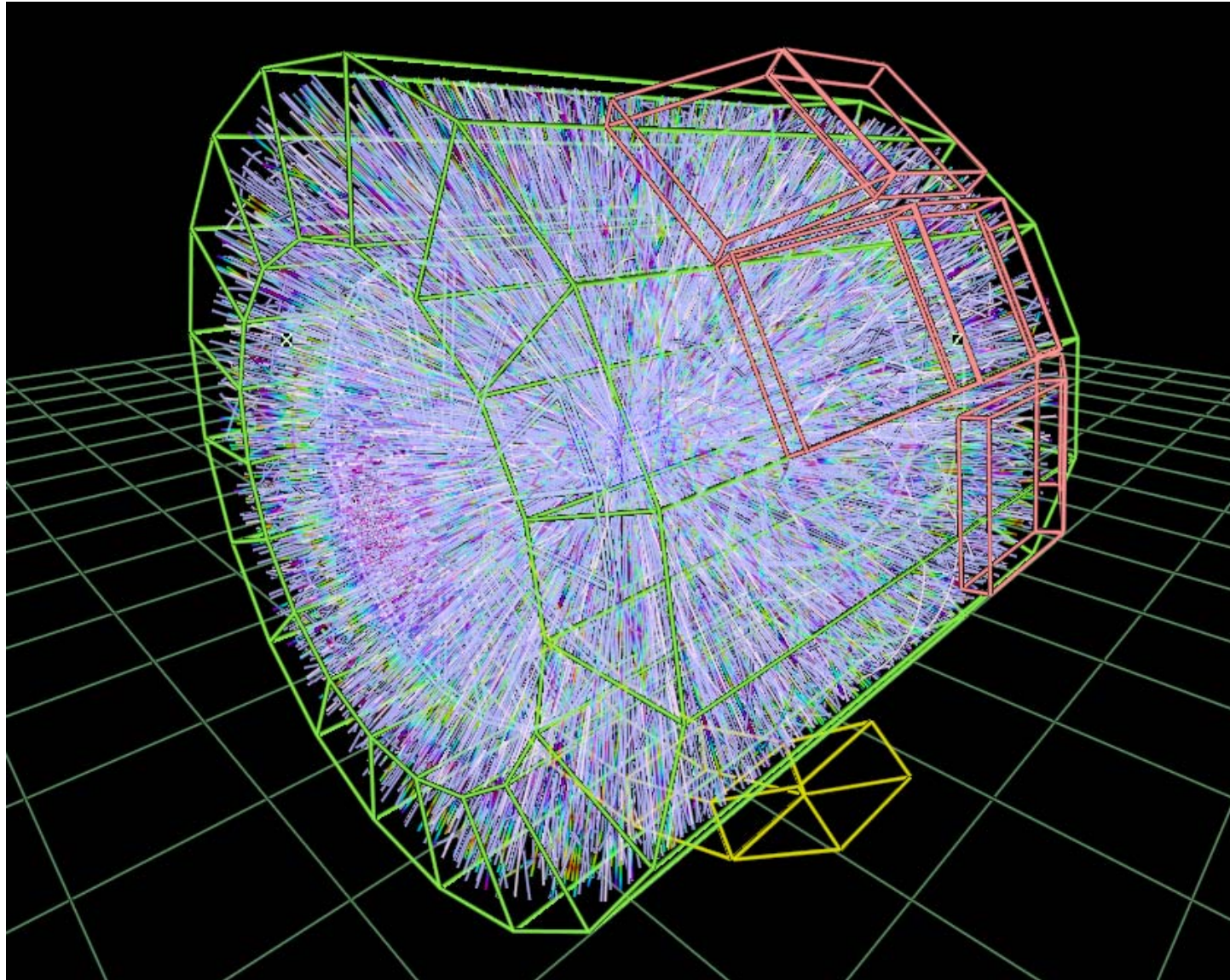
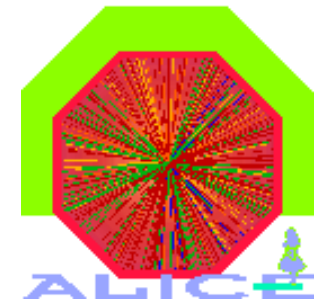
ALICE computing requirements



- Event rate to tape: 100 Hz
- Average event size: 12,5 MB
- Bandwidth: 1,25 GB/s
- Reconstruction of a PbPb event consumes 0,74 MSI2K · s or about 10 minutes
- Requirements (for 2010, ramp up until then)
 - Disk: 21 PB
 - MSS: 30 PB
 - CPU: 60 MSI2K (~ 40.000 cores 2 GHz)



Central PbPb Event



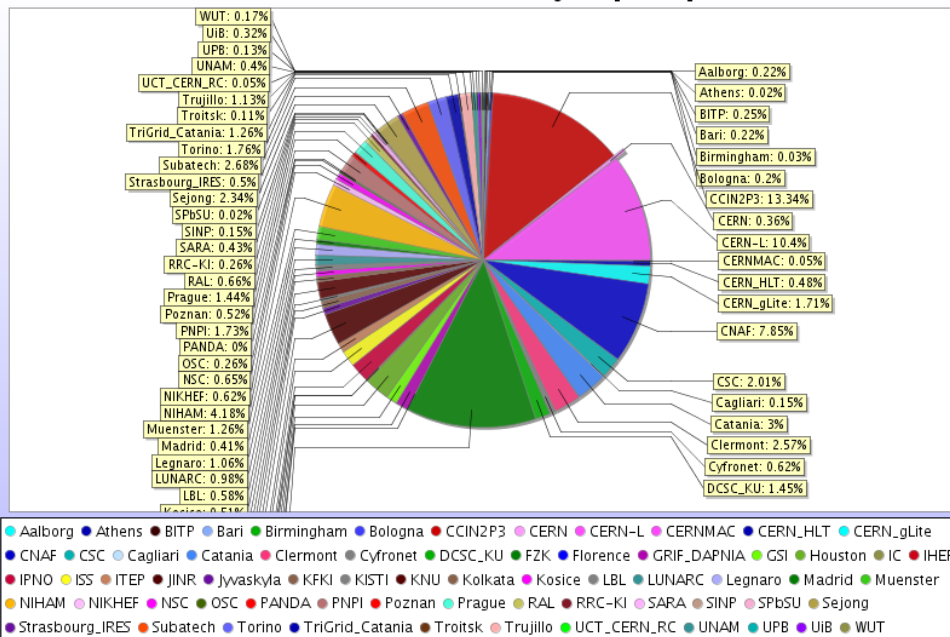
Matevz
Tadel

AliEn Grid Status

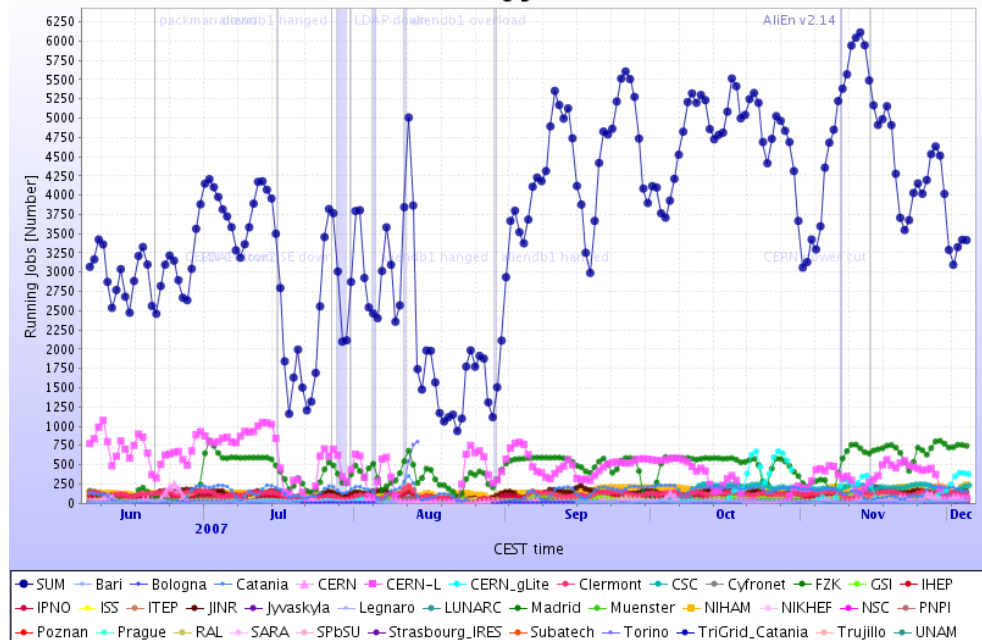
- Production running stable with jobs in the thousands for more than a year now
 - 62 sites on 4 continents
- Monitoring at <http://pcalimonitor.cern.ch>



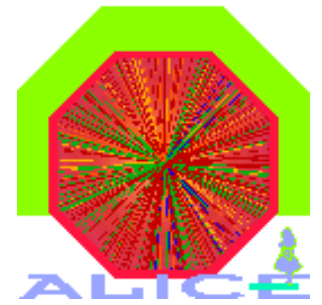
Total CPU time for ALICE jobs [hours]



Running Jobs

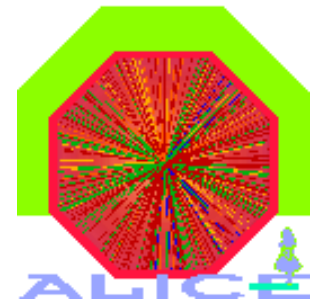


PROOF in Practice



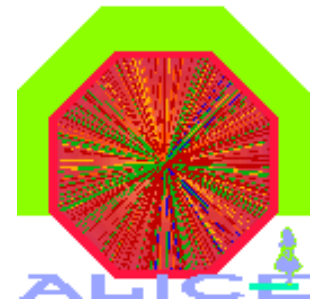
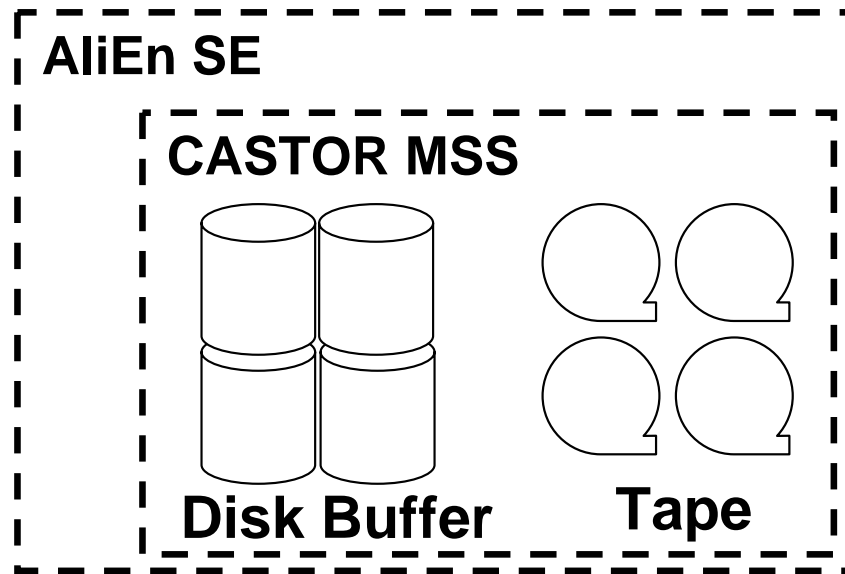
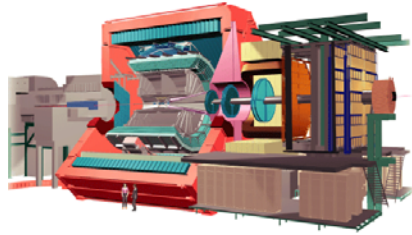
- Motivation
- Setup, Installation, Configuration, Monitoring
- Tests
- Grouping users, Fairshare, Staging, Quotas
- Integration with ALICE's software framework

PROOF for ALICE



- The **CERN Analysis Facility (CAF)** will run PROOF for ALICE for tasks that run on a short time scale and need many execution cycles
 - Prompt analysis of pp data
 - Pilot analysis of PbPb data
 - Calibration & Alignment
 - Fast simulation and reconstruction
- Additionally to using the Grid
 - Massive execution of jobs vs. fast response time
- Design goals
 - 500 CPU cores
 - 200 TB of selected data locally available

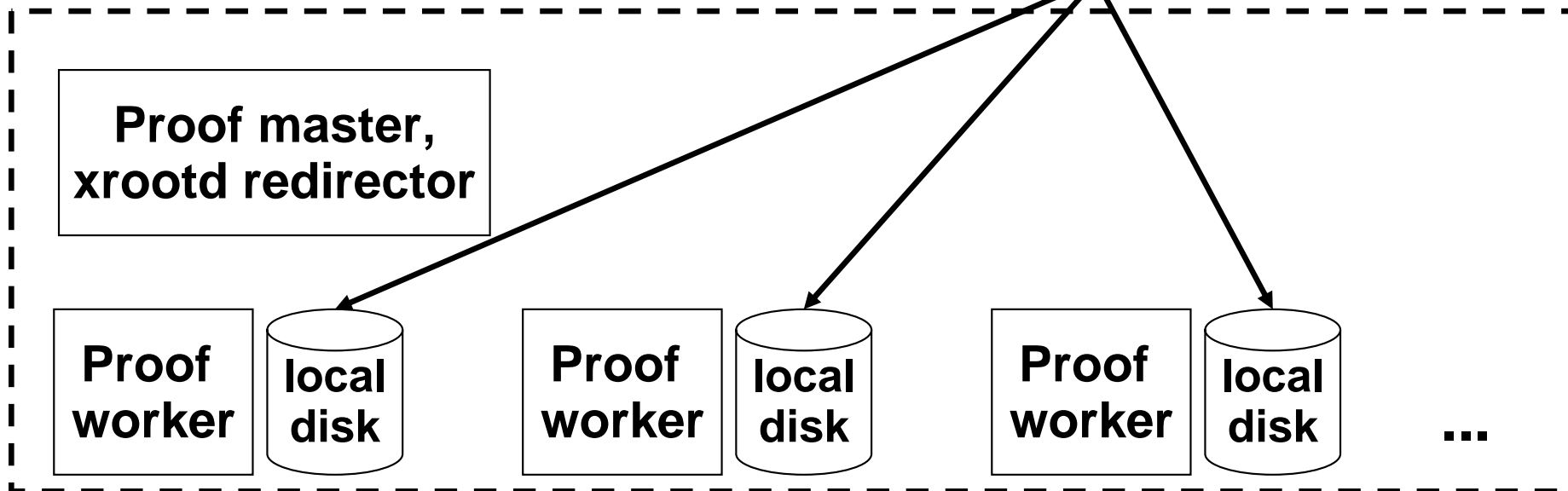
CAF Schema



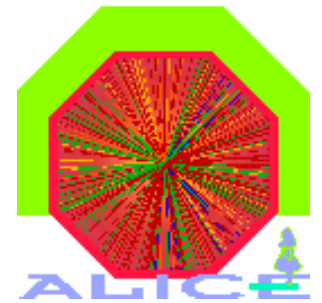
Tier-1
data export

Staging

CAF computing cluster

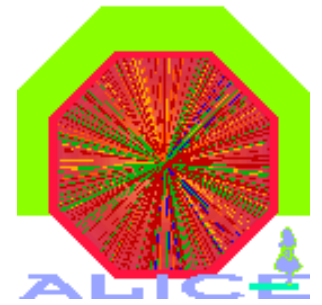


CAF Setup



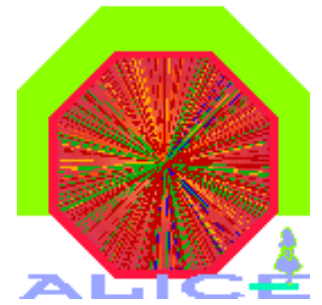
- Setup since May 2006
 - 40 machines, each: 2 CPUs, 200 GB disk → 8 TB
 - Standard CERN batch machines (lxbatch)
 - taken out of ALICE's LSF share
 - 5 as development partition, 35 as production partition
- Machines are a xrootd disk pool
 - xrootd redirector and PROOF master on head node
 - Other nodes are xrootd disk servers and PROOF slaves
 - Automatic staging from AliEn/CASTOR

Installation



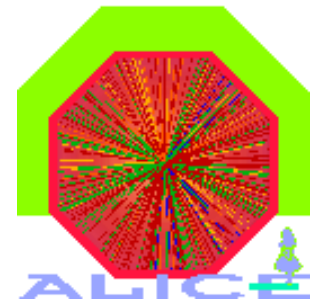
- ROOT (including xrootd) installed as RPM (deployed by IT, QUATTOR)
 - tar files for quick updates
 - ROOT version including sources also installed on AFS (public area)
 - As reference
 - For users running on lxplus
- AliRoot (ALICE's software framework) built against ROOT version on CAF, installed to AFS

Configuration



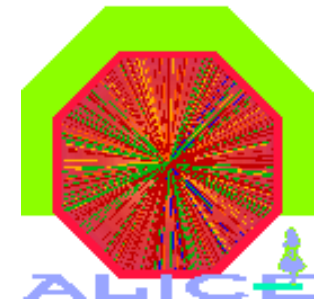
- For each partition (development, production)
 - xrootd/PROOF configuration (3 files) on AFS (public, no token needed!)
 - Staging script + alien api commands (runs on all slaves) on AFS
 - AliEn (Grid) user certificate deployed to each machine (to stage files from AliEn)
- Starting/ Stopping of services
 - Using wassh tool (execute commands on a list of hosts)

Notes on the Installation



- xrootd/olbd configuration sometimes requires a bit of trying, calm reflection and thinking outside of the box
 - in the end everything is logical 😊
- Certain turn-around time for RPMs, therefore "manual" deployment is advisable at the beginning
- Restarting kills all user sessions
 - Makes updates difficult in a production system
- Maintenance needs ~ 0.25 FTE (independent estimate from ALICE and ATLAS)

Monitoring

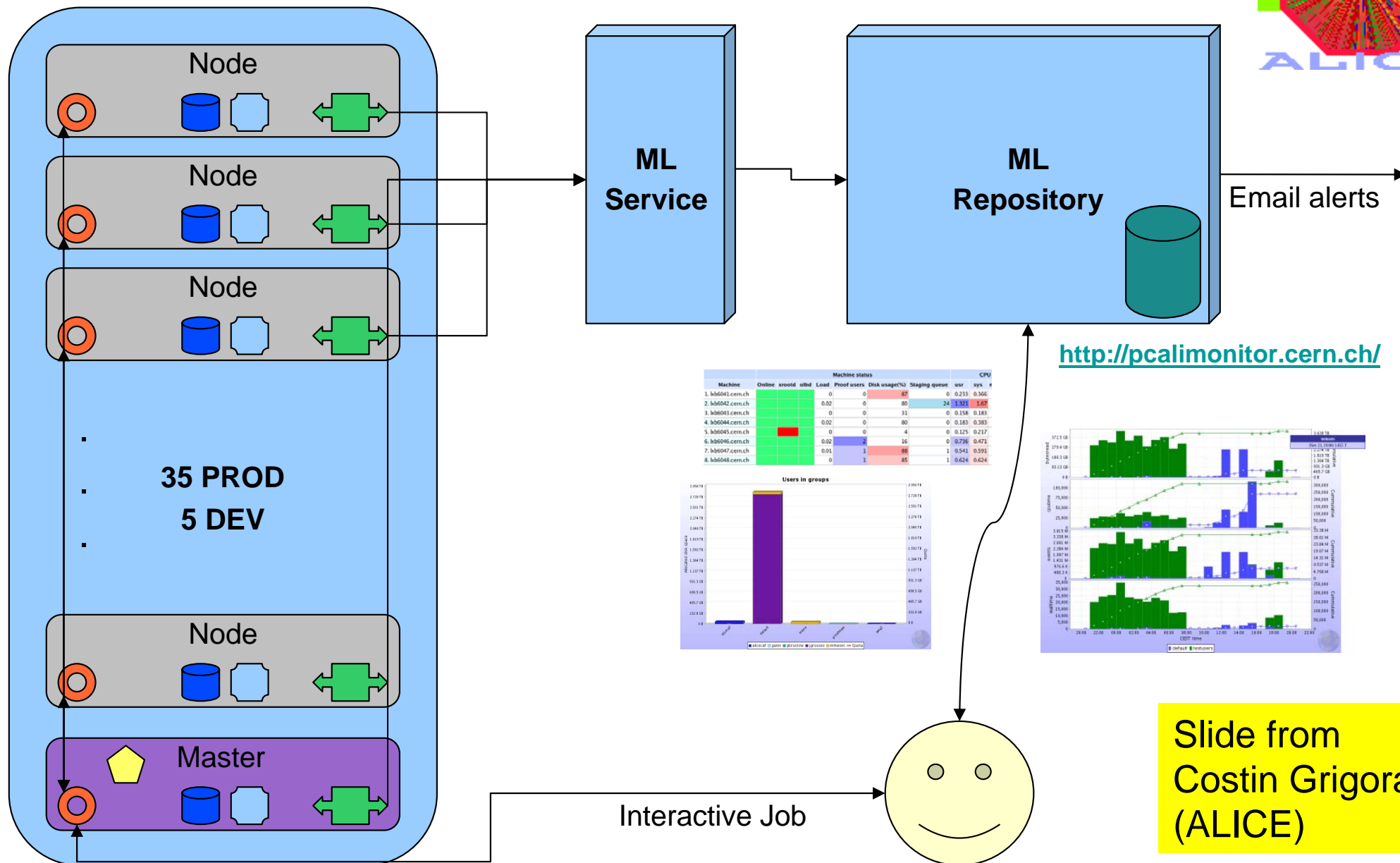
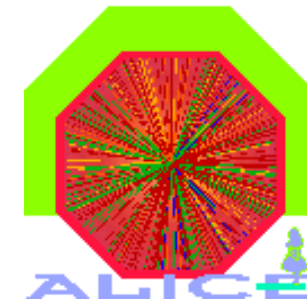


- Usual "lemon" monitoring by IT
- ALICE monitoring powered by MonALISA
- Machines
 - CPU, Memory, Network
 - Alerts
- Queries → CPU quotas
- Staging → Disk quotas
- Detailed query monitoring (to be enabled in query)
 - CPU, disk used by single queries
 - Network traffic correlation

Machine	Machine status					
	Online	xrootd	olbd	Load	Proof users	Disk
1. lxb6041.cern.ch	Green	Green	Green	0	0	Red
2. lxb6042.cern.ch	Green	Green	Green	0.08	0	White
3. lxb6043.cern.ch	Green	Green	Green	0.02	0	White
4. lxb6044.cern.ch	Green	Green	Green	0.33	0	White
5. lxb6045.cern.ch	Green	Red	Green	0	0	White
6. lxb6046.cern.ch	Green	Green	Green	0.11	2	Blue
7. lxb6047.cern.ch	Green	Green	Green	0.02	1	Red
8. lxb6048.cern.ch	Green	Green	Green	0.09	1	Red

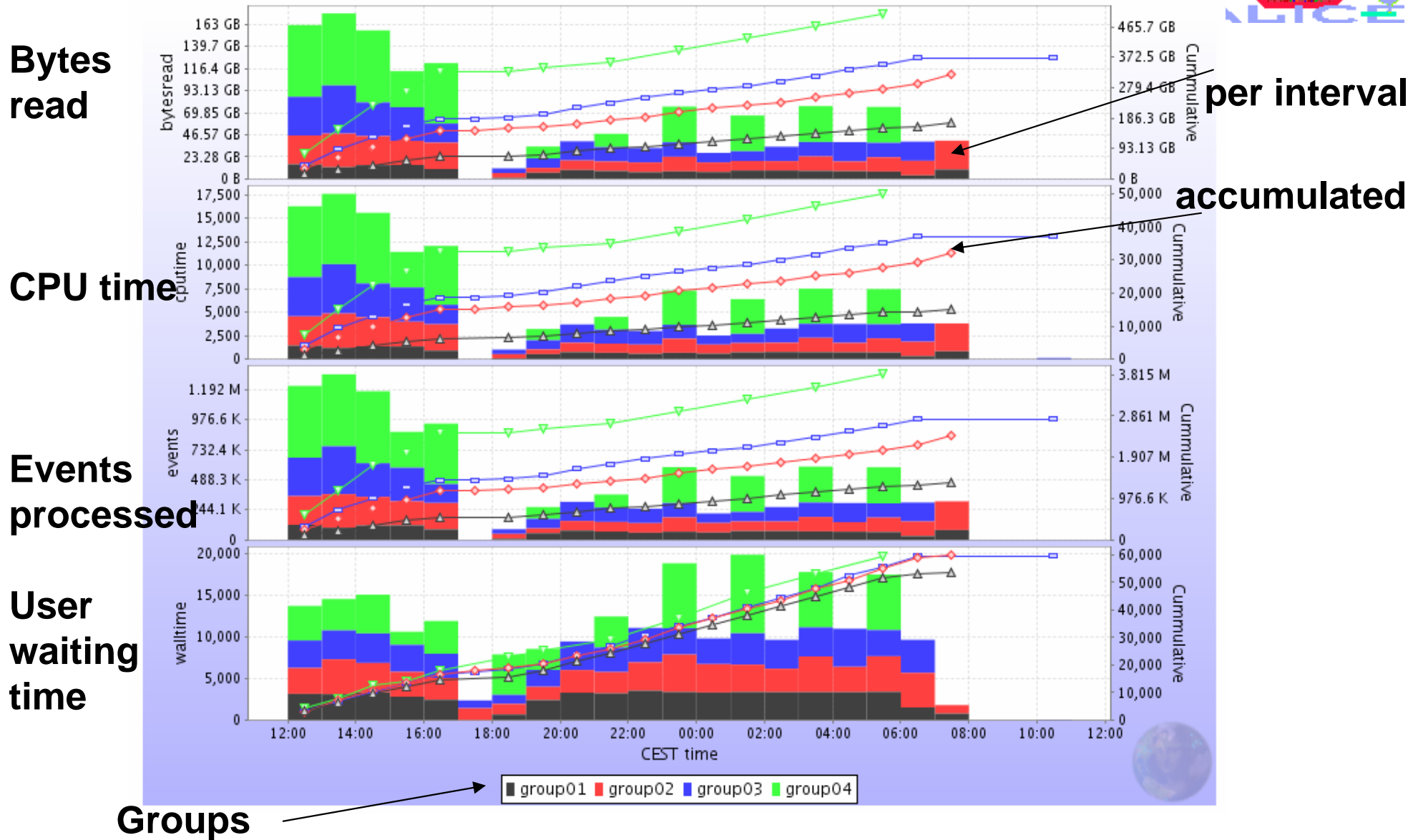
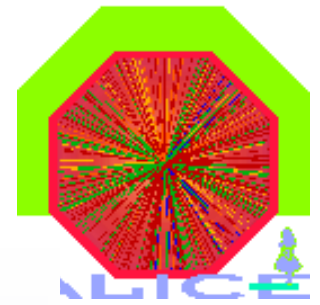
Machine	Traffic between the cluster machines (MB/sec) (last 0.5h average)																						
	6047	6048	6049	6050	6052	6053	6054	6055	6056	6057	6058	6059	6060	6061	6062	6063	6064	6065	6066	6067	6068	6069	607
1. 6047	0	-	-	-	-	-	2.927	2.018	-	-	1.094	-	-	-	1.908	4.112	-	-	0.974	0.614	0	0	-
2. 6048	-	9.406	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3. 6049	-	-	8.678	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4. 6050	-	-	-	6.692	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5. 6052	-	-	-	-	3.913	-	1.454	-	-	-	-	3.084	-	0.317	0	0	-	-	-	-	0.985	4.447	-
6. 6053	-	-	-	-	-	6.603	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7. 6054	0	-	-	1.363	-	-	6.195	-	-	-	0	-	-	-	0	-	-	-	-	-	0	-	1.58
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	-
13. 6060	-	-	-	-	-	-	-	2.068	-	-	-	-	11.59	-	-	1.06	-	-	-	-	-	-	2.02
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	-	-	-	0.58
18. 6065	0	-	-	-	3.52	3.165	-	0	-	-	0	-	-	-	-	-	3.034	0	1.579	-	0	-	-
19. 6066	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Monitoring with MonALISA

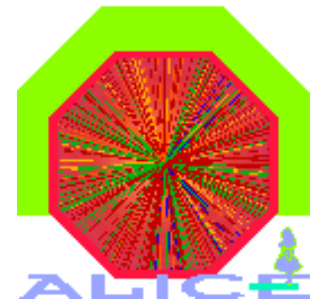


Slide from Costin Grigoras (ALICE)

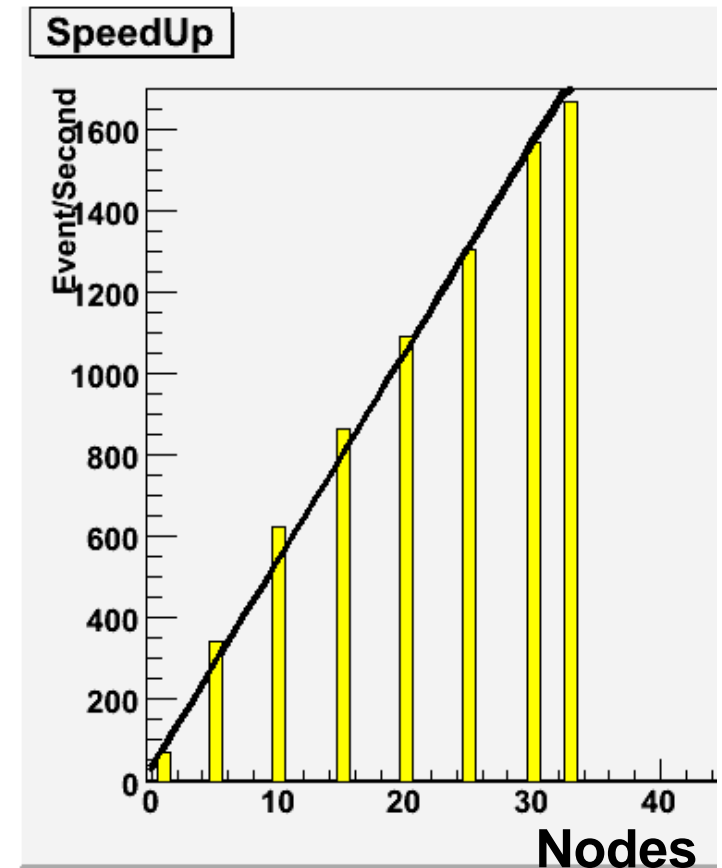
Monitoring of cluster usage



Tests



- Usability tests (~ 100 submitted bugs, ~ 20 open)
- Speedup test
- Evaluation of the system when running a combination of query types
- How does ALICE's analysis framework (AliROOT) work with PROOF

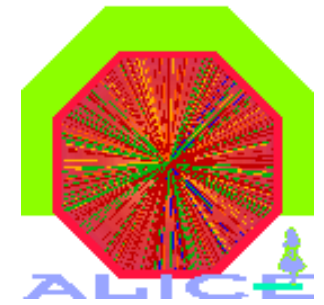


Usage Groups

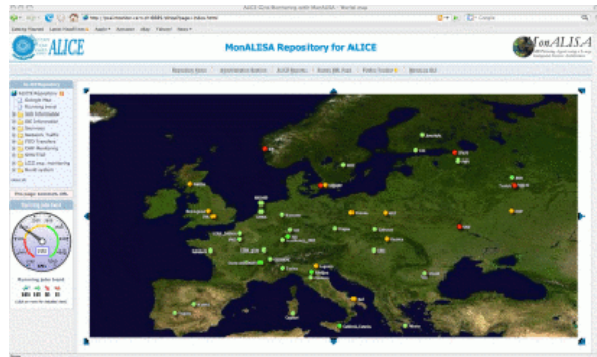


- ALICE wants to assure a fair use of the CAF resources
- Users are grouped
 - E.g. sub detectors or physics working groups
 - Users can be in several groups
- Each group has a CPU fairshare target
 - This governs the priority of concurrent queries
 - PROOF supports static (per query) priorities
 - A small piece of code provided by ALICE adjusts these priorities regularly (see next slides)
- Each group has a quota on the disk
 - This space can be used for datasets staged from AliEn
 - Involves a recent contribution from ALICE to the PROOF development (see next slides)

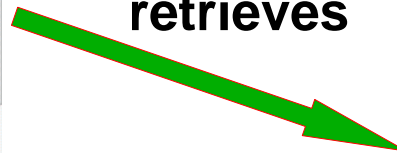
CPU Fairshare



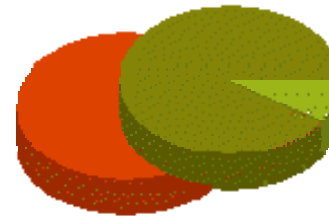
MonALISA



retrieves



Priority module

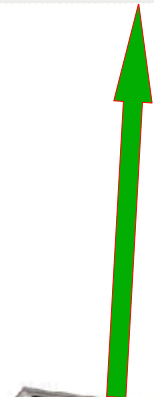


Retrieves **usage** values averaged over 1 month

$$f(x) = \alpha q + (\beta - \alpha) q * e^{\left(\frac{x}{q} * \ln \frac{1}{4}\right)}$$

Uses a correction formula to compute new **priorities**. It takes into account usage (x), quota and interval.
(i.e. $\alpha=0.5$, $\beta=2$, $q=20\%$, interval [10% ..40%])

reports



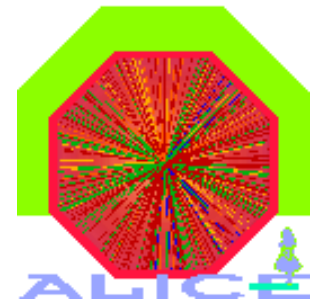
CAF cluster

stores priorities



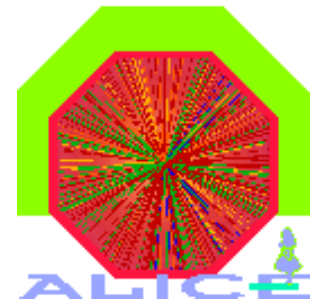
Slide from Marco Meoni (ALICE)

Staging



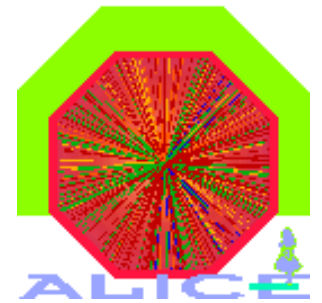
- Files are stored in AliEn SEs (for CERN: CASTOR)
- Step 1 (first months): Manual
 - Files copied by a shell script to redirector that balances between disk servers
 - To allow user staging the nodes were open for writing
 - Complicated for users, no control over quotas, difficult garbage collection
- Step 2 (until mid 2007): Semi-automatic
 - Staging script plugged into xrootd
 - Prepare request with stage flag or open request to a file triggered staging
 - User gets list of files from the AliEn File Catalog and triggers staging for all files
 - Convenient for users, no quotas, difficult garbage collection

Staging (2)



- Step 3 (now): Automatic
 - Staging script plugged into olbd
 - Implementation of PROOF datasets (by ALICE)
 - Staging daemon that runs on the cluster
 - Transparent migration from AliEn collection to PROOF datasets
 - Convenient for users, quota-enabled, garbage collection
- 3 TB in 100.000 files staged to the system

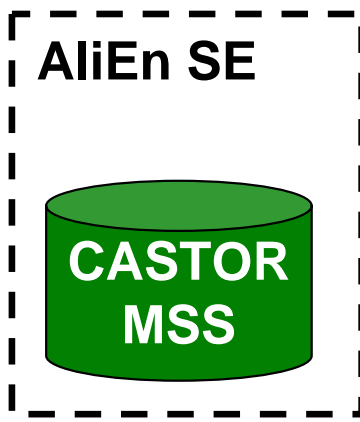
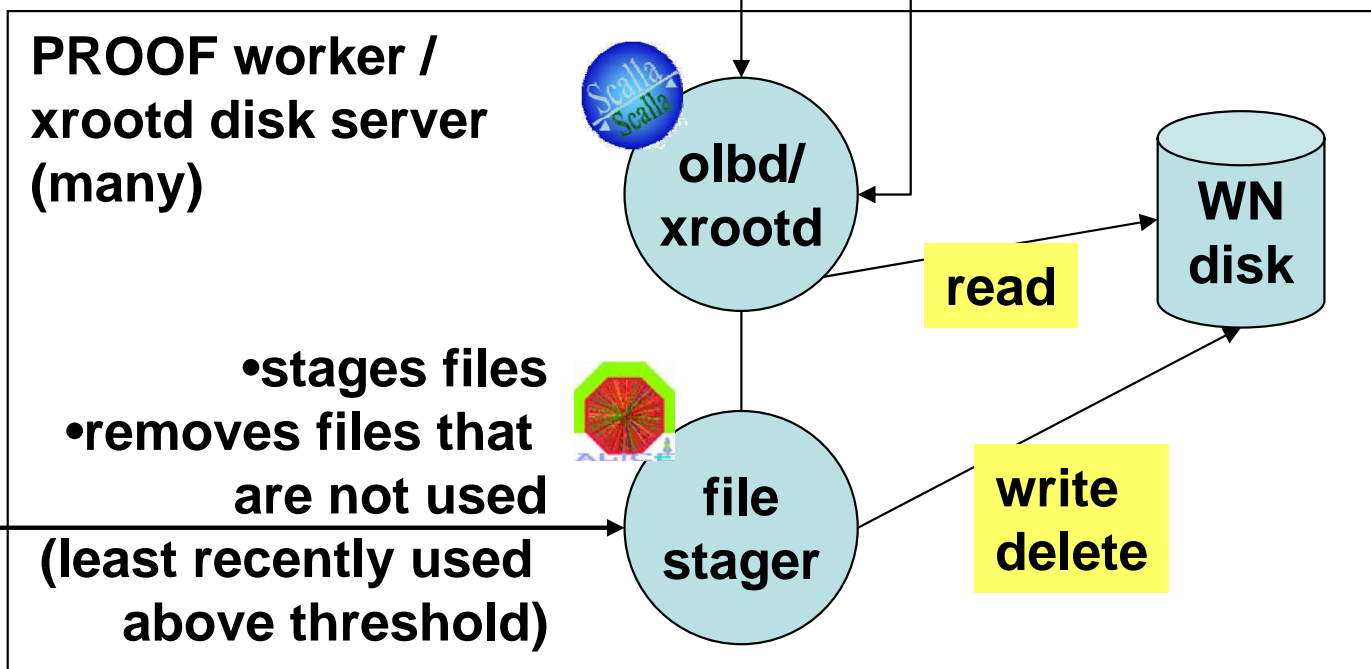
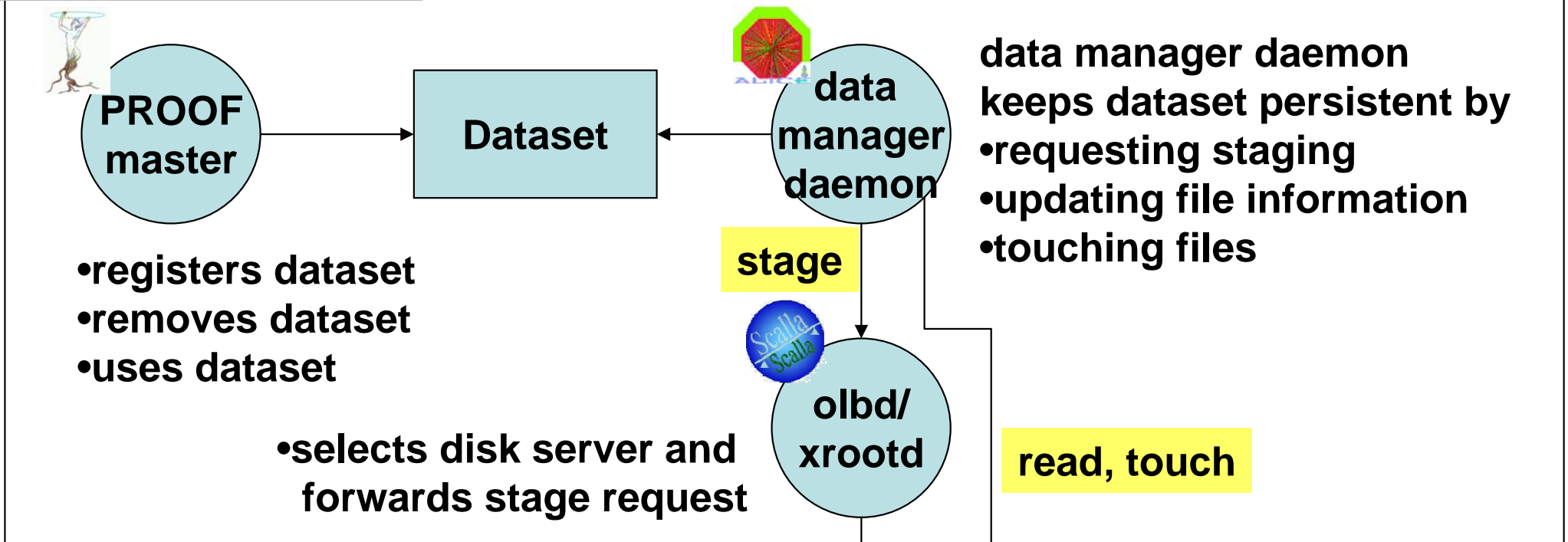
PROOF Datasets



- A dataset represents a list of files (e.g. physics run X)
 - Correspondence between AliEn collection and PROOF dataset
- Users register datasets
 - The files contained in a dataset are automatically staged from AliEn (and kept available)
 - Datasets are used for processing with PROOF
 - Contain all relevant information to start processing (location of files, abstract description of content of files)
- File-level storing by underlying xrootd infrastructure
- Datasets are public for reading (you can use datasets from anybody!)
- There are common datasets (for data of common interest)
- Datasets can also be used for files on a storage pool

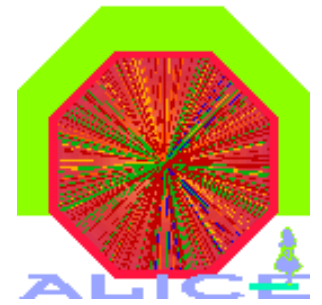
Dataset concept

PROOF master / xrootd redirector



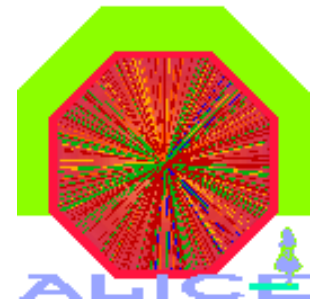
...

PROOF Master



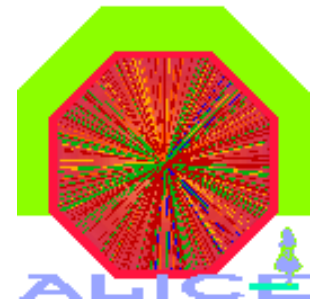
- Registering, removal of datasets
 - Checks quota upon registration (group level quotas)
- Display datasets, quotas
- Use datasets
 - Meta data contained in dataset allows to skip lookup and validation step

Data Manager Daemon



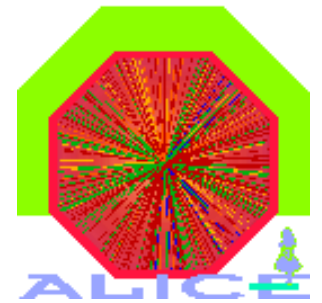
- Keeps content of datasets persistent on disk
- Regularly loops over all datasets
- Sends staging requests for new files
 - Converts filenames
e.g. alien://<path> to root://<redirector>/alien/<path>
- Extracts meta data from recently staged files
- Verifies that all files are still available on the cluster (by touch, prevents garbage collection)
 - Speed: 100 files / s

File Stager



- Two directories configured in xrootd/olbd for staging
 - /alien
 - /castor
- Staging script (given with olb.prep directive)
 - Perl script that consists of 3 threads
 - 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 (least recently used above threshold)

Datasets in Practice



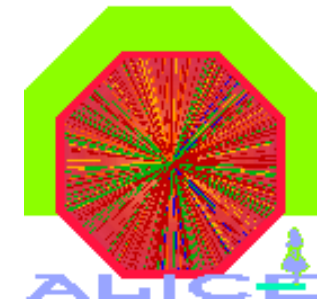
- Create DS from AliEn collection
 - collection = TGrid::OpenCollection(lfn)
 - ds = collection->GetFileCollection()
- Upload to PROOF cluster
 - gProof->RegisterDataSet("myDS", ds)
- Check status: gProof->ShowDataSet("myDS")

```
root [9] gProof->ShowDataSet("ESD5000")
Info in <TXProofServ::GetDataSet> on master-0: uri=ESD5000

TFileCollection ESD5000 - title contains: 21899 files with a size
of 1177346326985 bytes, 100.0 % staged
The files contain the following trees:
Tree /esdTree: 2189800 events
Tree /HLTesdTree: 2189800 events
```

- Use it: gProof->Process("myDS", "mySelector.cxx+")

Dataset in Practice (2)

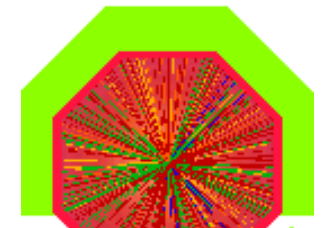


- List available datasets: `gProof->ShowDataSets()`

```
root [3] gProof->ShowDataSets()
Dataset URI                                     |# Files|Default tree|# Events| Disk | Staged
/default/jgrosseo/ESD100                       |  6764|/esdTree   | 676400| 343 GB| 100 %
/default/jgrosseo/run82XX_part1                | 10000|/esdTree   | 998900| 288 GB| 99 %
/default/jgrosseo/run82XX_part2                | 10000|/esdTree   | 944700| 272 GB| 94 %
/default/jgrosseo/run82XX_part3                | 10000|/esdTree   | 987900| 285 GB| 98 %
/default/jgrosseo/ESD600                       |  1844|/esdTree   | 184400|  51 GB| 100 %
/default/jgrosseo/ESD_FullMisalignment         |   944|/esdTree   |  92100|  47 GB| 97 %
/default/jgrosseo/run12000                     |    62|/esdTree   |    49 |   4 GB| 79 %
/default/jgrosseo/ESD5000                      | 21899|/esdTree   |2189800|1096 GB| 99 %
/default/jgrosseo/ProofSessionFiles            |25438|/esdTree   |4460900|1640 GB|100 %
```

- You always see common datasets and datasets of your group
- This method was used to stage 3 M events (30.000 files)

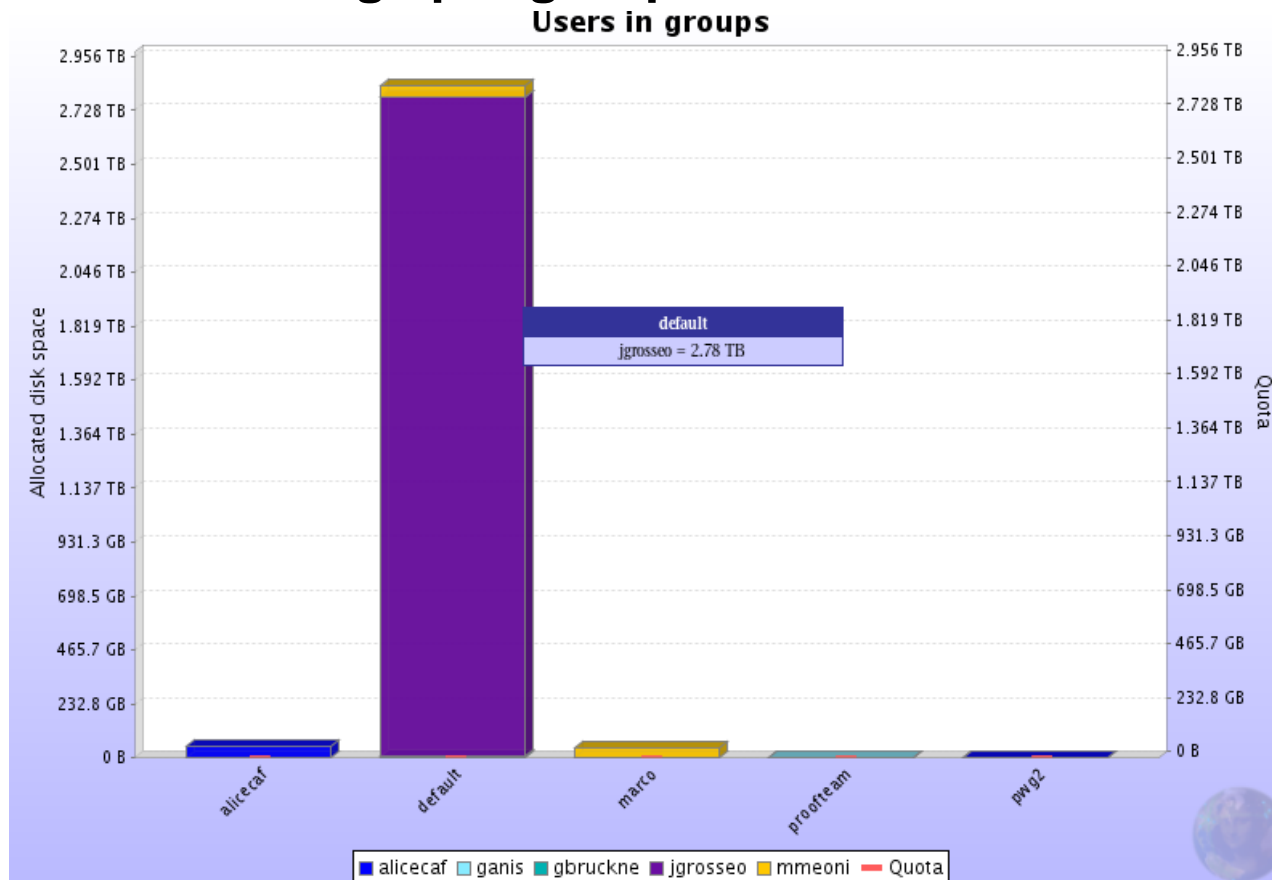
Monitoring of datasets



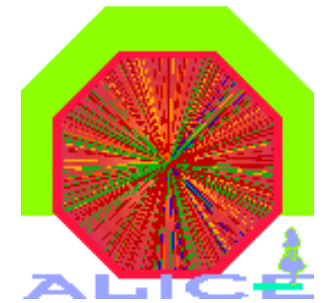
Number of files per host

15. lxb6055.cern.ch	2765	133.8 GB
16. lxb6056.cern.ch	2795	133.2 GB
17. lxb6057.cern.ch	2785	131.9 GB
18. lxb6058.cern.ch	2772	129.1 GB
19. lxb6059.cern.ch	1704	64.2 GB
20. lxb6060.cern.ch	1917	82.55 GB
21. lxb6061.cern.ch	2766	130.3 GB
22. lxb6062.cern.ch	1228	40.67 GB
23. lxb6063.cern.ch	2789	131.9 GB
24. lxb6064.cern.ch	2777	131.3 GB
25. lxb6065.cern.ch	2755	128.2 GB
26. lxb6066.cern.ch	2364	118.2 GB
27. lxb6067.cern.ch	2745	127.9 GB
28. lxb6068.cern.ch	2740	128.2 GB
29. lxb6069.cern.ch	2778	129.6 GB
30. lxb6070.cern.ch	2749	128.9 GB
31. lxb6071.cern.ch	2756	130.4 GB
32. lxb6072.cern.ch	2699	124.8 GB
33. lxb6073.cern.ch	2741	128 GB
34. lxb6074.cern.ch	2737	127.2 GB
35. lxb6075.cern.ch	2755	128.9 GB
36. lxb6076.cern.ch	2741	130.6 GB
37. lxb6077.cern.ch	2731	125.8 GB
38. lxb6078.cern.ch	2741	128.5 GB
39. lxb6079.cern.ch	2739	126.8 GB
40. lxb6080.cern.ch	601	16.8 GB
Total	96964	4.431 TB
Average	2620	122.6 GB

Data set usage per group



ALICE Software Framework



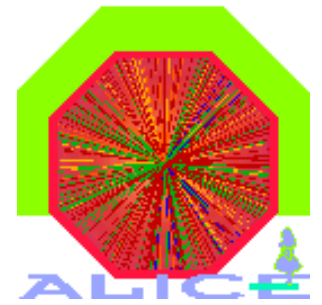
- AliROOT
 - Based on ROOT (ROOT + libraries = AliROOT)
 - AliROOT + PROOF → How to load libraries?
- Data that is to be analyzed
 - Reconstruction output: Event Summary Data (ESD), Analysis Object Data (AOD)
 - Only requires a few libraries (PROOF: few packages)
 - Physics working group libraries can be converted automatically to PROOF packages
 - Make target added to AliROOT Makefile to create .par files

Analyzing Reconstruction Output



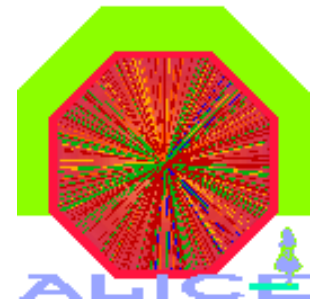
- Step 1 (until mid 2007): (Ali-)Selector
 - ESD.par package uploaded by standard PROOF functionality
- Step 2 (now): Analysis Framework
 - Framework to combine several analyses → “Analysis train”
 - Benefit from having one event in memory
 - Analyses appear as AliAnalysisTask (TTask)
 - Framework in several .par files (desirable: global package space)
 - AliAnalysisTasks need to be distributed to PROOF cluster
 - Frequent changes → .par file inconvenient
 - New PROOF functionality: TProof::Load(“class”)

Using Full AliRoot



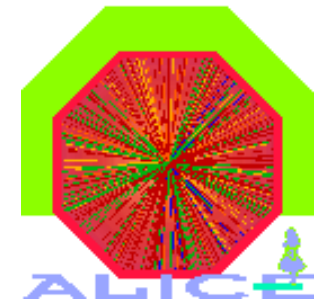
- Access to clusters, raw data (detector debugging)
- (Nearly) full AliRoot needs to be loaded
- AliRoot is manually deployed on AFS built against the ROOT version(s) running on CAF
- Enabled by 3 line macro
 - Sets environment variables, loads libraries
 - Effectively converts ROOT instance running on proof slave into AliROOT
- Feasible method for all ROOT based software

User Support

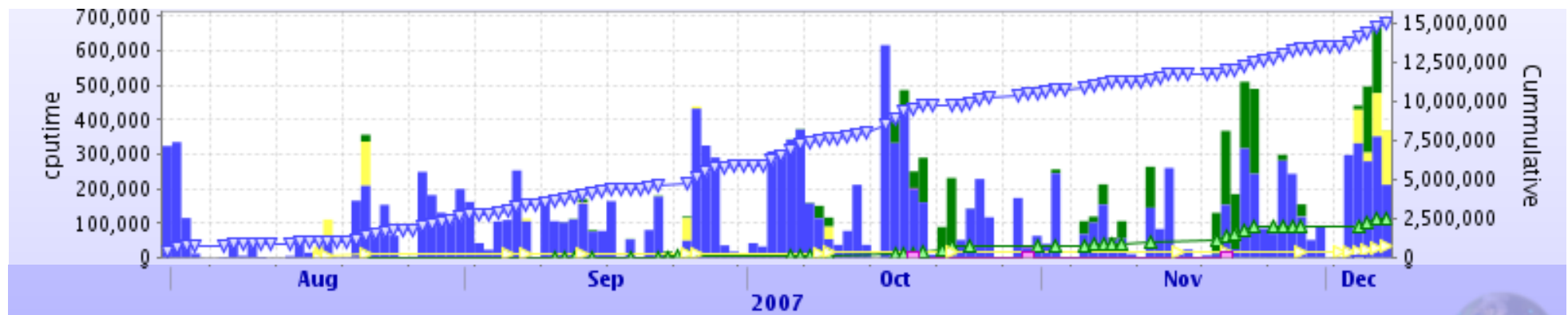


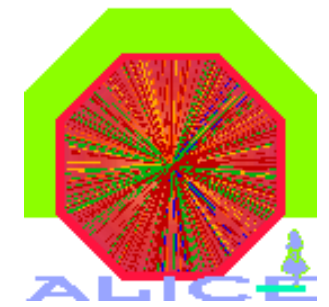
- A PROOF training session (3 hours) is part of the standard ALICE tutorials since over a year now
 - Connecting, package management
 - Using AliRoot in PROOF
 - Running the analysis framework with PROOF
 - Lots of hands-on examples
 - 200 users followed the tutorial so far
- Dedicated analysis mailing list

Usage Statistics



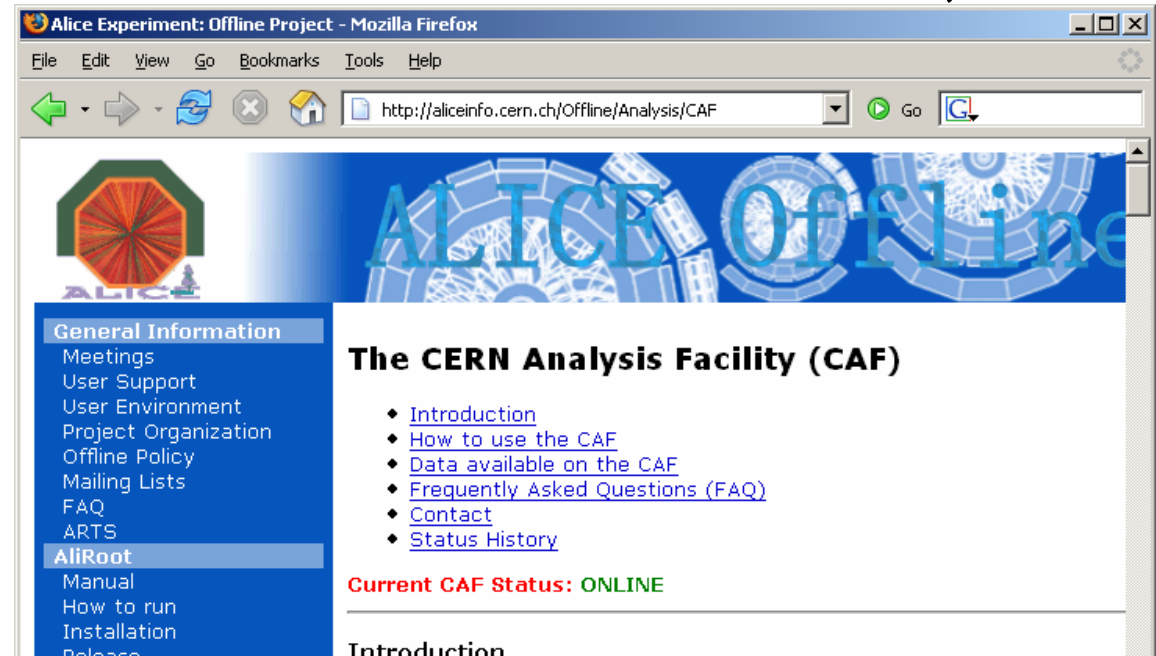
- Detailed monitoring started 4 months ago
 - 13 M CPU seconds by users = 150 days, but only 4% of total available CPU
 - Due to not enough active users
 - ~ 60 TB read from disk, 500 M events
 - 45 unique users (+ training)
 - > 30 users use the system regularly



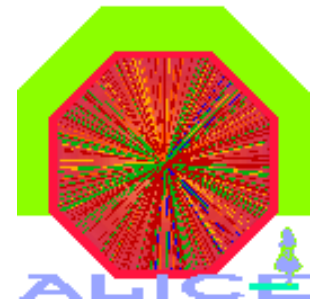


More Information about the ALICE CAF

- <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://aliceinfo.cern.ch/Offline/Analysis/Tutorial/>
- Information about server installation
<http://root.cern.ch/twiki/bin/view/ROOT/ProofInstallation>

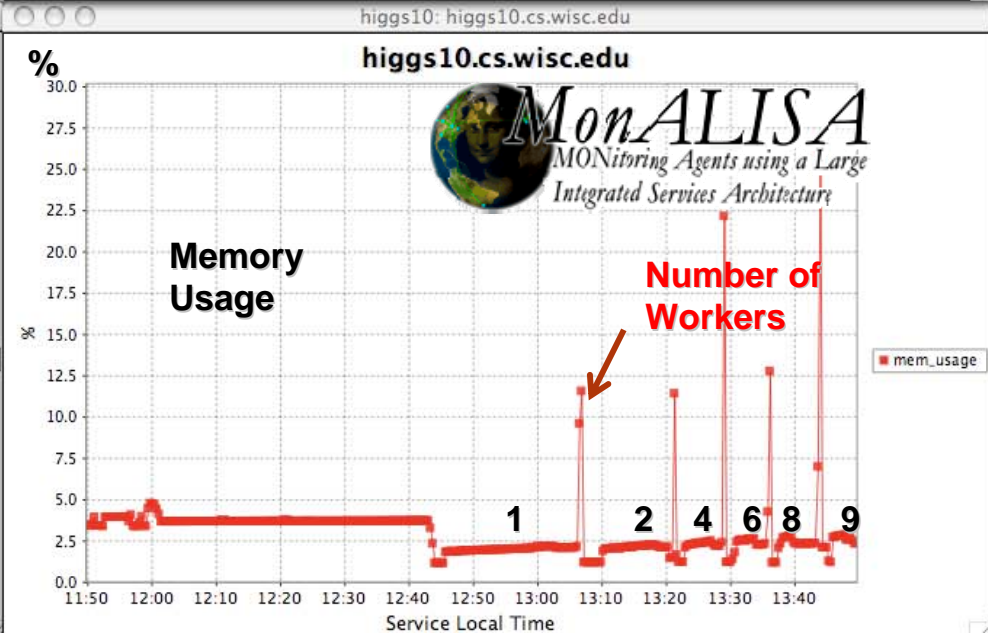
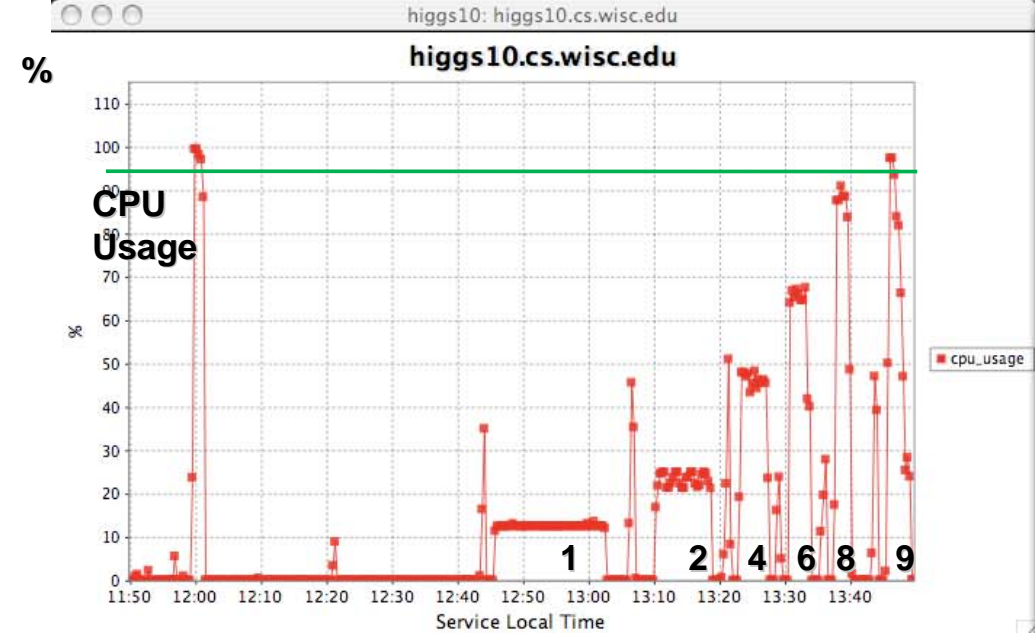
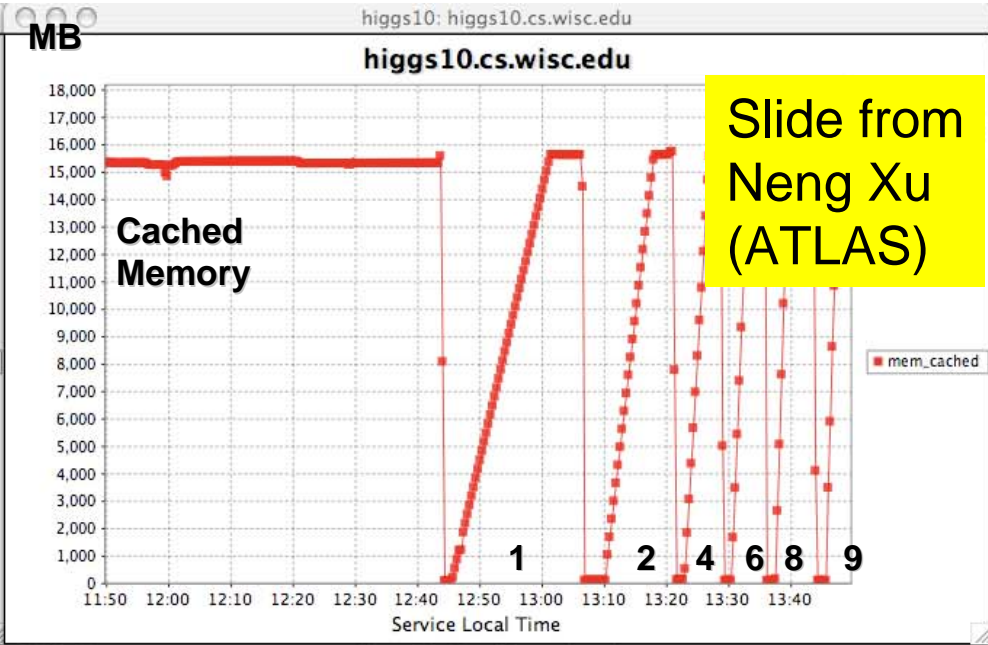
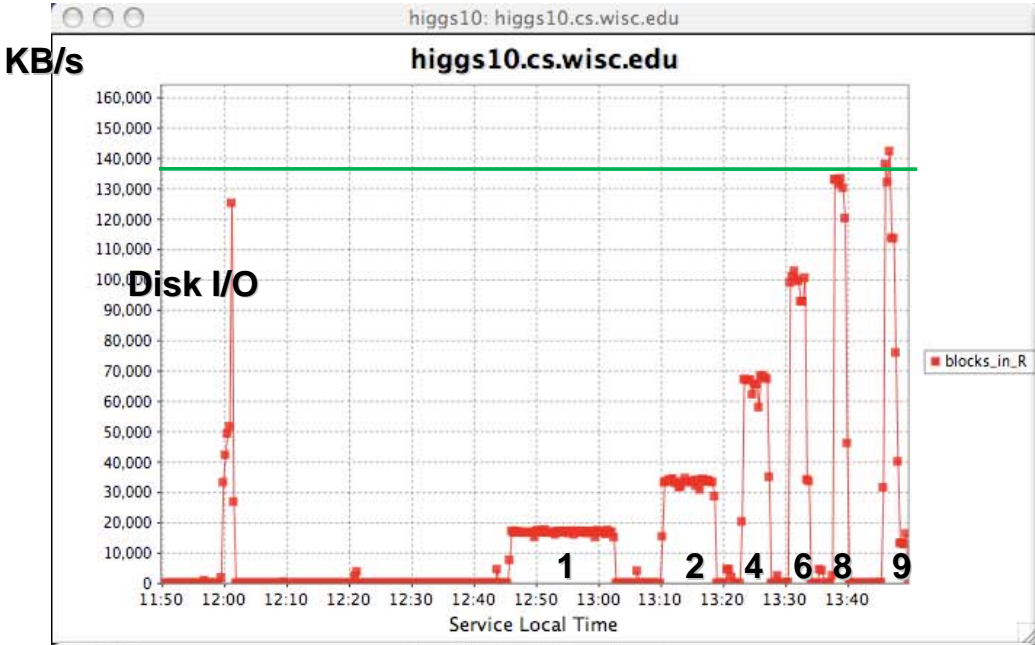


CAF/PROOF Workshop



- CAF/PROOF workshop at CERN on the 29th / 30th Nov
- 18 contributions, 70 registrations
- Topics
 - The PROOF system, the xrootd system
 - Benchmarks
 - CAF setups: Experience from ALICE (CERN, GSI) and ATLAS (BNL, Wisconsin)
 - System administration, Installation, Configuration
 - Staging and file distribution
 - Combination with CONDOR, gLite
 - User Feedback
- <http://root.cern.ch/root/PROOF2007>

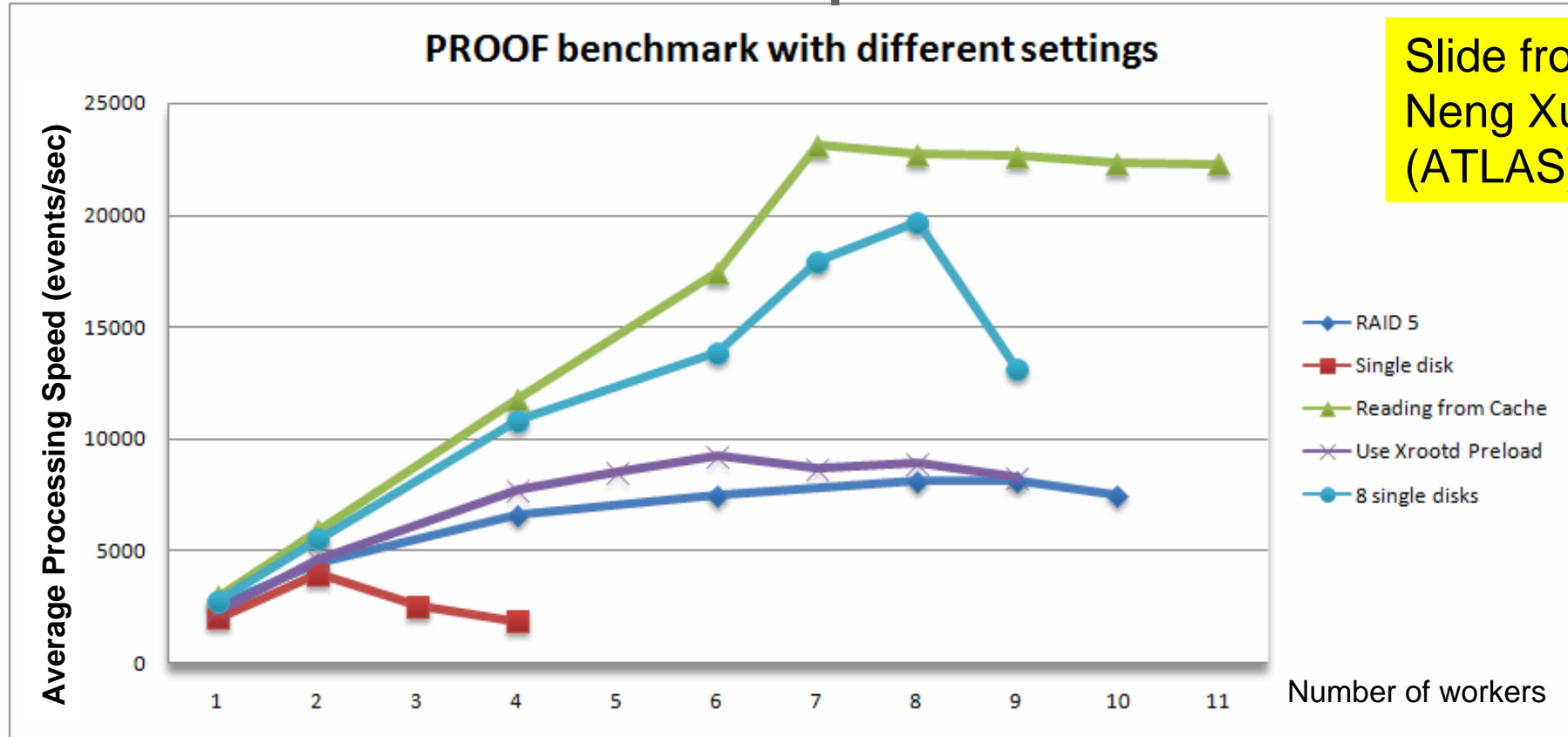
Benchmark files, big size, read all the data



The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory,

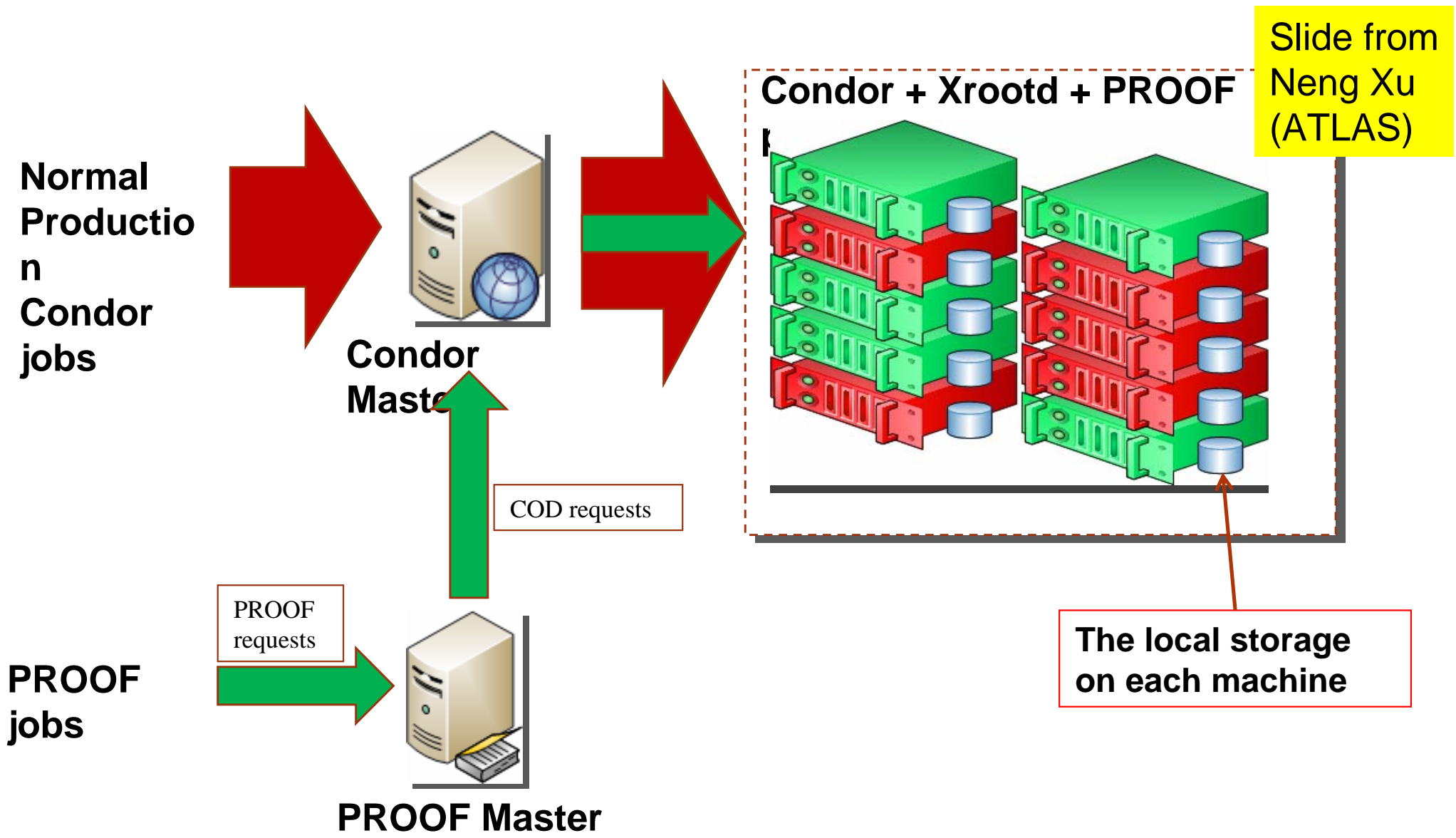
8 single disks on RAID Controller

An overview of the performance rate



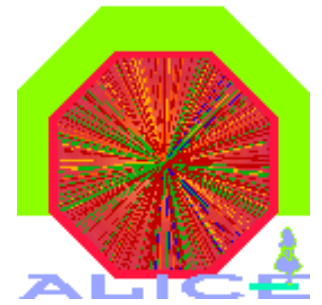
- For I/O bound job, disk throughput will be the bottleneck for most of the hardware configurations; reading data from memory will provide the best scalability and performance.
- Trade-off between the data protection and the working performance has to be made.
- Using Xrootd Preload function will help to increase the analysis speed.
- 2 cores vs. 1 disk seems to be a reasonable hardware ratio without using Raid technology.

The Developing PROOF+COD Model



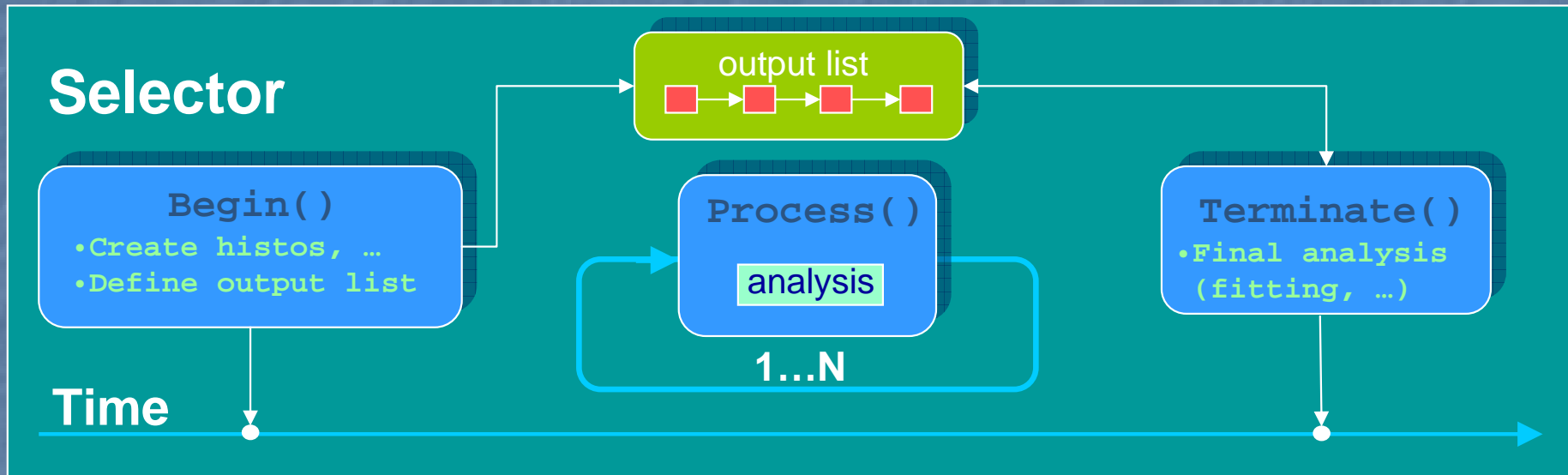
User Feedback

(from the CAF workshop)



- CAF preferred given the complexity of Grid solutions
- Analysis speed appreciated 😊
 - Minutes vs. Hours
 - Gain in speed can be higher than the available CPU for I/O bound analysis and non-local data
- Fail-over problems (e.g. crash on one worker)
- Memory monitoring needed
- "Simulation" of a PROOF session on the own PC needed (for debugging)

Implement algorithm in a TSelector



New TProof::Process(const char *selector, Long64_t times)

```
// Open the PROOF session
root[0] TProof *p = TProof::Open("master")
// Run 1000 times the analysis defined in the
// MonteCarlo.C TSelector
root[1] p->Process("MonteCarlo.C+", 1000)
```

Slide from Gerri Ganis (CERN)



Output file merging

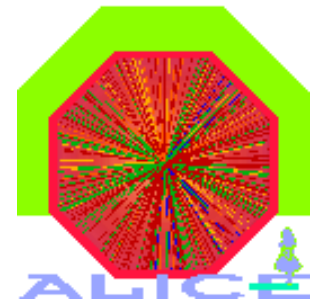


L. Tran-Thanh

- **Large output objects** (e.g. trees) create memory problems
- Solution:
 - save them in files on the workers
 - merge the files on the master using TFileMerger
- New class **TProofFile** defines the file and provide tools to handle the merging
 - Unique file names are created internally to avoid crashes
- Merging will happen on the Master at the end of the query
- Final file is left in sandbox on the master or saved where the client wishes
- Included in ROOT 5.17/04

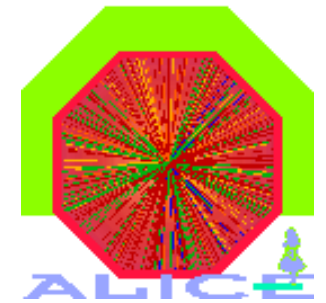
Slide from
Gerri Ganis
(CERN)

Demo



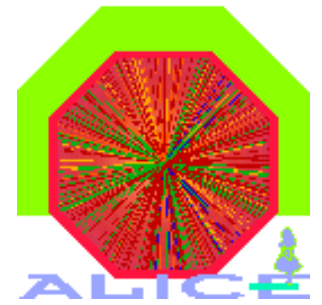
- Run a simple analysis that creates an uncorrected p_T spectrum from the reconstruction output (ESD)
- Needed steps
 - Connect to the PROOF cluster
 - Load libraries/par files
 - Initialize Framework
 - Create input list (chain)
 - Run analysis
- Analysis 2000 files, 100 events each, 2.5 MB each
 - Only track branch switched on
- Compare: Local PC at CERN vs. CAF at CERN

Summary



- ALICE uses PROOF on a cluster called CERN Analysis Facility. It will allow prompt and pilot analysis, calibration/alignment, fast simulation and reconstruction
→ Fast response time
 - ALICE computing model only foresees a CAF at CERN, maybe also good solution for other centers
 - **Discontinuity of contract for main developer seen very critical**
- A setup is in place since May 2006
 - Users are trained in tutorials and use the system for "analysis"
- Active collaboration with ROOT/PROOF team
 - Contribution from ALICE to PROOF development
 - Dataset concept and CPU quotas implemented and under test

Credits



- Gerardo Ganis (CERN, PROOF team)
- Andreas Joachim Peters (CERN, IT)
- Costin Grigoras (ALICE)
- Marco Meoni (ALICE)