The CERN Analysis Facility – A PROOF Cluster for
Day-One Physics Analysis

Jan Fiete Grosse-Oetringhaus
CERN, 1211 Geneva 23, Switzerland
E-mail: Jan.Fiete.Grosse-Oetringhaus@cern.ch

Abstract. ALICE (A Large Ion Collider Experiment) at the LHC plans to use a PROOF
cluster at CERN (CAF – CERN Analysis Facility) for analysis. The system is especially aimed
at the prototyping phase of analyses that need a high number of development iterations and thus
require a short response time. Typical examples are the tuning of cuts during the development of
an analysis as well as calibration and alignment. Furthermore, the use of an interactive system
with very fast response will allow ALICE to extract physics observables out of first data quickly.
An additional use case is fast event simulation and reconstruction. A test setup consisting of
40 machines is used for evaluation since May 2006. The PROOF system enables the parallel
processing and xrootd the access to files distributed on the test cluster. An automatic staging
system for files either catalogued in the ALICE file catalog or stored in the CASTOR mass
storage system has been developed. The current setup and ongoing development towards disk
quotas and CPU fairshare are described. Furthermore, the integration of PROOF into ALICE’s
software framework (AliRoot) is discussed.

1. Introduction
Today’s high energy physics experiments, like ALICE, produce tremendous amounts of data
(in the order of several PB per year). Because such an amount of data cannot be processed
on single batch farms Grid solutions have been developed, extensively discussed also during
this conference. ALICE’s software runs in batch mode on the Grid. It performs large-scale
production or reconstruction as well as analysis over large amounts of data.

This model introduces a delay in obtaining the results. Users receive them after a given
time which is the execution time of the program itself plus an overhead that tends to be
bigger than in case of batch systems. Some tasks have a short execution time and need many
iterations. Examples are the development of algorithms and analysis code as well as calibration
and alignment procedures. The use of a local computer (which has clearly the lowest overhead)
is feasible only if the data volume to be processed and CPU time needed are reasonably small.

In addition to the Grid solution, ALICE decided to offer to its users a system that allows to
process a big amount of data (compared with a single machine) without the need of submitting
a Grid job. The system – called CERN Analysis Facility (CAF) – runs PROOF on a cluster at
CERN. PROOF allows interactive parallel data processing, thus the overhead is very small and
many development cycles are possible in short time. However, only a fraction of the total data
recorded by the experiment will be available on the CAF. Therefore, after successful prototyping
the analysis code will be sent as a Grid job to process all available data.
These proceedings shortly introduce the PROOF system. The structure of the CERN Analysis Facility is described with the solutions for monitoring, data distribution and management as well as CPU fairshare. The last section describes the integration with ALICE’s software framework AliRoot.

2. The PROOF system
The Parallel ROOT Facility (PROOF) [1] enables interactive parallel data processing on a cluster. The system is particularly suited to process events produced by high-energy physics experiments: in most analysis use cases, events can be processed in an arbitrary order and partial results can be summed up after processing (trivial parallelism).

Figure 1. Schema of the PROOF system

Figure 1 shows the schema of the system. A user running a ROOT session on her client connects to a so-called PROOF master which in turn opens a ROOT session on each so-called PROOF worker. The user sends a query, that consists of the analysis code and a list of files that are to be processed (step 1 in Figure 1). The master assigns data fragments to each worker which are then processed (step 2). The data is assigned such that data local to the worker is processed first, then non-local data, if remaining. After processing, the results are merged on the master (step 3) and returned to the user (step 4). Naturally this requires mergeable results, which is the case for typical ROOT objects like histograms and trees. User objects have to implement the merging functionality themselves. This is done by exposing a merge function.

The system is used interactively from the ROOT prompt, thus results can be visualized directly. At the expense of some overhead, result objects, e.g. histograms, can be monitored while they are being produced (so-called feedback histograms). User libraries for processing can be distributed with so-called par archives (also called PROOF packages).

PROOF runs as a plugin to xrootd [2] and uses its communication layer. Furthermore, xrootd is used as file-serving system. Generally, a PROOF setup consists of a few PROOF
masters and so-called xrootd redirectors. These can be, but must not be on the same machines. The remaining machines act as PROOF workers and so-called xrootd disk servers. To benefit from the advantage of processing data that is local to the PROOF worker, they ideally act as PROOF worker and xrootd disk server at the same time.

It is reminded that each xrootd redirector as well as each xrootd disk server consists of two services: the xrootd part that serves files and the olbd part that handles load-balancing and file discovery. More details can be found in [2].

3. The CERN Analysis Facility (CAF)
The CERN Analysis Facility (CAF) is a cluster at CERN running PROOF. It is planned to comprise 500 CPUs and 100 TB of local disk space attached to the workers. The data is staged from AliEn [3] upon request of physics working groups and detector experts. The CAF shall allow prompt analysis of p+p data, pilot analysis of Pb+Pb data, fast event simulation and reconstruction and especially calibration and alignment. The focus of the system is on fast response time. Figure 2 shows the schema of the setup together with the data flow that will be explained in this section.

![Figure 2. Schema of the CERN Analysis Facility](image)

3.1. Test Setup
A test setup is in place since May 2006. It consists of 40 "standard" CERN batch machines. Each of them has two cores with 2.8 GHz, 4 GB of RAM and 250 GB of disk. The cluster has been divided into a development partition of 5 machines to allow testing of new versions and a production partition of 35 machines that is open to users. In each partition one machine is configured as PROOF master and xrootd redirector, the remaining machines as PROOF workers and xrootd disk servers.
Especially at the beginning, the testing led to a considerable amount of fixes for discovered problems and feature requests. Examples for the latter are the possibility to retrieve the output of a crashed session, the introduction of global packages (par files that are available to all users), the functionality to select the ROOT version upon connect and a method to easily distribute a single class (e.g. containing quickly evolving analysis code) to the cluster, that is then compiled and loaded, without the need to create a par file.

3.2. Monitoring
The CAF is monitored on different levels with the help of MonALISA [4]. Apart from host-monitoring, query statistics is recorded after each query. This includes the amount of data processed, the number of events, the CPU time consumed and the time the user waited for the query to finish (walltime). This information is used to visualize the utilization and performance of the cluster as well as for a CPU fairshare mechanism that is under development.

![Figure 3. Cluster utilization monitoring with MonALISA](image)

The monitoring output can be seen at http://pcalimonitor.cern.ch, section "CAF monitoring". As an example, Figure 3 shows the usage history of different users and groups (color coding). Shown is the consumed CPU time and processed data where the bars indicate the usage per interval (left axis) and the line the accumulated usage (right axis). The small overlay shows the relative CPU usage of several groups in a given period as pie chart.

The monitoring allows to verify that the cluster resources are fully exploited. The utilization during test runs that saturate the cluster shows that between 70% – 90% of overall CPU is consumed by user processes.

3.3. Data Distribution
The disk space on CAF is not a permanent storage. Instead, it is used as a cache space for data imported from other external data storage systems. It is not meant to store permanent data, because the data that is requested for analysis might change quickly. Furthermore, the requirements on data integrity have been kept low, thus there is no backup of the disks (standard batch machines are used). Instead, the CAF is able to stage data stored in AliEn SEs. The automatic staging is described in the next section.

About 30,000 files have been staged, which corresponds to 3 million p+p events with a total size of 1.5 TB. The availability of these files has been communicated to the users using text files
that contain the list of available files. This simple way will be replaced by datasets that are introduced in Section 3.3.2.

### 3.3.1. Datastager

The staging of data from AliEn is done by a datastager script that is plugged into the olbd on each disk server. It has two tasks: dynamic staging and garbage collection.

The workflow is as follows: upon a stage request, the xrootd redirector selects a xrootd disk server and forwards the request. The disk server sends the request to its datastager which performs the staging. The staging itself is performed by interaction with the AliEn services. The datastager is also able to stage files directly from CASTOR [5], however, this is not used in ALICE.

The garbage collection is triggered when the disk usage is above a high-water mark (90%). It deletes files with the oldest access time first and requires the files not to have been accessed for a fixed amount of time (1 day). This condition preserves files from deletion that are virtually currently in use and is important for the consistency of the datasets. The garbage collection stops when a low-water mark (80%) is reached. The numbers quoted in parentheses are in use on the test setup and are open for optimization.

### 3.3.2. Datasets & Automatic Staging

Using single files as entities for analysis does not seem appropriate. Usually statistics from more files is needed and users prefer to run over the same set of files several times to have reproducibility. Files are therefore grouped and this set of data (or dataset) is used to describe the input for analysis. Datasets also allow transparent automatic staging of the contained files.

In collaboration with the PROOF team, ALICE developed a dataset implementation for PROOF with the following features:

- A dataset represents a list of files (e.g. a physics run).
- A correspondence between AliEn datasets and PROOF datasets exists, i.e. a PROOF dataset can be created from an AliEn dataset.
- Analysis can be performed on a dataset, thus certain information is hold with the dataset (e.g. exact location of the files, abstract description of the content).
- Datasets are owned by users, but are public for reading by everybody. Furthermore, global datasets of public interest can be defined. These are not accounted against someone’s quota.

It is optional if the files contained in a dataset are staged automatically to the cluster (by default they are). In this case the following applies:

- After registration of a dataset, the contained files are staged automatically and kept available on the cluster.
- The files themselves are stored by the underlying xrootd infrastructure.
- Files that are in several datasets are staged only once.

Figure 4 shows the structure of the solution that has been developed. The PROOF master registers and removes datasets upon request from the user. The dataset itself (the description, not the files) is stored in a file on the master machine, or in case of several masters in a shared file system or via xrootd.

A data manager daemon (that is ALICE-specific) loops regularly over the datasets and checks that all files that are listed in the datasets are available. Staging is triggered for missing files by sending a stage request to the redirector’s olbd service. Once a file is staged, file meta information needed for processing is extracted and stored in the dataset. This file meta information includes the file’s exact location, its size and the number of contained events. By regularly touching the file it is prevented that it is cleaned by the garbage collection on the disk server. Upon deletion
of a dataset the files are not directly deleted, instead they are not touched anymore and are cleaned shortly after by the garbage collection (1 day in the current configuration). This way simplifies the system and has advantages over the case of direct deletion. Direct deletion would imply to verify that a file is not in use by any other dataset before deleting it. This would require to loop over all datasets which would feature poor scaling properties. Alternatively a file catalog of the staged files could be kept, which is a quite complicated approach. Furthermore, the delayed deletion has the advantage that the files stay on the cluster until the space is needed (it was mentioned before that the garbage collection runs only when the disk usage is above a given high-water mark). Thus if another user creates a dataset with files that still reside on the cluster, they do not need to be staged again.

On each xrootd disk server the mentioned datastager takes care of the staging of the files. The system has been implemented and is currently under test on the development partition of the CAF.

3.4. Groups & Quota & Fairshare

Users are organized in groups. Groups are e.g. physics working groups or subdetectors of ALICE. Users can be member of several groups.

The utilized resources per group are monitored. These are the disk space used by datasets and the consumed CPU time. Quotas are applied for the disk space. Furthermore, a fairshare algorithm is applied that adjusts the priorities of concurrently running PROOF sessions to allow the consumed CPU time to reach a certain target per group. A recently developed priority mechanism in the PROOF system is used. ALICE provides the input which are the relative priorities of the groups that use the system. This input is calculated based on consolidated

Figure 4. Schema of the dataset and automatic staging solution
usage values retrieved from MonALISA.

3.5. Integration with ALICE’s software framework AliRoot

ALICE’s software framework AliRoot [6] is based on ROOT. It can be described as a ROOT session in which a set of libraries has been loaded. Naturally, the set of libraries needed depends on the type of analysis to be done:

- For end-user analysis (including comparison to Monte Carlo) only a few small libraries are needed that enable to understand ALICE’s data format, i.e. ESD (Event Summary Data) and AOD (Analysis Object Data).
- Other tasks need access to the raw detector output and the intermediate reconstruction output, e.g. the clusters. Examples are detector debugging as well as calibration and alignment. For this kind of analysis the full framework is needed.

ALICE stores several AliRoot versions in a shared space (AFS) that can be loaded into the PROOF session. This is done by a few commands that set the needed include and library paths and then trigger the loading of the libraries. Effectively this converts the session running on the PROOF worker from a ROOT session into an AliRoot session (second case mentioned above).

Optionally, user libraries as well as packages from physics working groups can be distributed as par files. Generally, the ESD and AOD libraries could also be distributed as par archives. ALICE tries to discourage that to prevent the use of wrong versions of the libraries.

3.6. Outreach

Some education is needed for users to realize the advantages of an interactive parallel system and to get acquainted with its usage. The fact that analysis of LHC data will not be feasible on a user’s laptop anymore has not reached the users or is not accepted by the users yet. ALICE facilitates the usage of systems like PROOF by offering active support in the form of monthly hands-on tutorials and fast response on mailing lists. 180 users followed the tutorials up to now, out of which about 20 use the system regularly. Furthermore, the usage is encouraged by the fact that the available data challenge data can only be analyzed using distributed systems like PROOF or AliEn.

4. Summary

These proceedings present the approach of the ALICE experiment enabling fast analysis with many iterations on a reasonably large amount of data. Complementary to the Grid solution, the CERN Analysis Facility running PROOF will allow to perform calibration and analysis as well as to prototype analysis that is then used to process large amounts of data on the Grid.

A test setup is in place since May 2006, that has been evaluated and introduced to ALICE users and shows promising performance. The uptime of this system was over 99%. About 180 users have been trained during hands-on tutorials, out of which 20 regularly use the system. The number of active users can clearly be increased.

While exercising the system ALICE has identified desired feature requests and contributed to the PROOF development, examples are dataset handling and the CPU fairshare.

The next steps are to increase the cluster size and to extensively test the new features. ALICE is looking forward to upcoming PROOF developments like the possibility to write output to files on each PROOF worker that are automatically merged as well as to run analysis that is not tree-based. The latter will allow to run event simulation and reconstruction on the CAF.

1 It is planned to use the AliEn package manager to deploy AliRoot.
Acknowledgments
The author would like to thank Gerardo Ganis for the good teamwork during the PROOF developments, Andreas-Joachim Peters for the collaboration during the development of the datastager, Andrew Hanushevsky for his support with xrootd, Costin Grigoras for the implementation of the visualization of the data sent to MonALISA and the CERN IT staff for maintaining the CAF machines.

References
[1] Ballintijn M, Roland G, Brun R and Rademakers F The PROOF distributed parallel analysis framework based on ROOT 2003 Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP), La Jolla, California, 24-28 Mar 2003; Rademakers F, Ballintijn M, Bellenot B, Brun R, Ganis G, Iwaszkiewicz J Latest developments in the PROOF system 2007 These proceedings
[2] Boeheim C, Hanushevsky A, Leith D, Melen R, Mount R, Pulliam T and Weeks B The Scalla software suite: xrootd/olbd, http://xrootd.slac.stanford.edu
[3] Aphecetche L, Buncic P, Piskac R, Revsbech J E and Sego V AliEn - ALICE environment on the GRID 2003 Nucl. Instrum. Meth. A 502 437; Saiz P, Bagnasco S, Betev L, Buncic P, Carminati F, Cirstoiu C, Grigoras C, Mendez Lorenzo P, Peters A, Rademakers F, Saiz P AliEn2: the ALICE Grid environment 2007, These proceedings
[4] Legrand I C, Newman H B, Voicu R, Cirstoiu C, Grigoras C, Toarta M and Dobre C MonALISA: An agent based, dynamic service system to monitor, control and optimize Grid based applications 2004, Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP), Interlaken, Switzerland 27 Sep - 1 Oct 2004
[5] Baud J P, Couturier B, Curran C, Durand J D, Knezo E, Occhetti S and Barring O CASTOR status and evolution 2003 Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP), La Jolla, California, 24-28 Mar 2003
[6] Carminati F, Buncic P, Hristov P, Morsch A, Rademakers F, Safarik K, The AliRoot framework, status and perspectives, Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP), Interlaken, Switzerland 27 Sep - 1 Oct 2004; Carminati F, The ALICE Offline framework 2007, These proceedings