Status of the ALICE CERN Analysis Facility

Marco Meoni\textsuperscript{1}, Jan Fiete Grosse-Oetringhaus\textsuperscript{2} and Federico Carminati\textsuperscript{2} for the ALICE collaboration

\textsuperscript{1} Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
\textsuperscript{2} CERN, 1211 Geneva 23, Switzerland

E-mail: Marco.Meoni@epfl.ch, Jan.Fiete.Grosse-Oetringhaus@cern.ch, Federico.Carminati@cern.ch

Abstract. The ALICE experiment at CERN LHC is using a PROOF-enabled cluster for fast physics analysis, detector calibration and reconstruction of small data samples. The current system (CAF - CERN Analysis Facility) consists of some 120 CPU cores and about 45 TB of disk space distributed across the CAF hosts. One of the most important aspects of the data analysis on the CAF is the speed with which it can be carried out. The system is particularly aimed at the prototyping phase of analyses that need a high number of iterations and thus require a short response time. Quasi-online quality assurance of data can be obtained. The paper describes the design principles of the PROOF framework and presents the current setup, performance tests and usage statistics. Subsets of selected data can be automatically staged in CAF from the Grid storage systems, therefore data distribution and staging techniques are described in depth. A fairshare algorithm to adjust the priorities of concurrently running sessions is also examined. Furthermore, the adaptation of PROOF to the AliEn/gLite Grid middleware is described. This approach enables a dynamic startup of PROOF nodes worldwide with the purpose to process much larger physics datasets.

1. Introduction

The ALICE\textsuperscript{1} experiment at the CERN Large Hadron Collider (LHC) will accumulate data at unprecedented speed and volume. The yearly estimate is 1.5PB of raw data from the experimental setup and additional 1.5PB of reconstructed data and Monte-Carlo simulations. The data management and processing is done on the Worldwide LHC Grid\textsuperscript{2, 3} (WLCG), encompassing hundreds of computing centres with many thousands of CPUs and PB scale disk and mass storage systems. The data volumes and distributed computing environment presents a set of unique challenges for the reconstruction and analysis software and the ways the physicists perform data analysis.

However, this computing model has an intrinsic delay in providing results from the data that are processed. In the AliEn model, the Grid is a large distributed batch system where user jobs typically process very large data volumes and must pass through several steps. They are submitted to a central queue, then split and assigned to computing farms with suitable resources, scheduled within the local batch systems and executed on the worker nodes. Outputs are finally stored on the user’s sandbox. This approach is not efficient for code prototyping where tasks have generally short execution time and need several iterations over a dataset. Typical examples are the tuning of cuts during the development of an analysis as well as calibration and alignment.
Figure 1. PROOF schema on a computing cluster. A user connects to the PROOF master that in turn starts a ROOT session on each worker node. The user analysis code is sent to the master and replicated on each worker. Each worker processes data fragments and produces a partial result out of them. Results are sent back to the master and merged. The final result is displayed on the user’s screen.

The ALICE experiment offers to its users a cluster for quick interactive parallel data processing called CERN Analysis Facility [4] (CAF) which runs the PROOF [5] software (Parallel ROOT Facility). CAF is operational since May 2006 and has been heavily improved and extended over the last three years. These proceedings describe the functioning schema of PROOF and examine the current status of the ALICE CAF and its components. Usage statistics on user grouping, CPU fairshare and disk quota are presented. Staging techniques to trigger the copying of files from the Grid upon user requests are discussed. The last section introduces an ongoing development: the adaptation of the PROOF framework to the AliEn/gLite [6, 7] Grid middleware. This allows to dynamically startup PROOF nodes worldwide and process physics datasets much larger than what is available on a local PROOF farm.

2. The PROOF framework

PROOF is a framework developed to enable interactive parallel data processing on a local cluster. It has been designed as an interactive alternative to batch systems for central analysis facilities and departmental workgroups (Tier-2s). PROOF is particularly suited for processing event-based data produced in high-energy physics experiments. In ALICE the system is suitable for interactive analysis of pp and A+A Event Summary Data (ESD), calibration and alignment of detector components as well as fast event reconstruction.

Events produced in HEP experiments are independent and data can be processed using the trivial or event-based parallelism. Events can be processed in an arbitrary order without changing the final result. PROOF splits the input data into fragments and processes them in parallel on many CPUs. Partial results are finally merged together producing the final result as on a single machine. Users can interactively run their code from an interface (the common ROOT [8] prompt), get the tasks executed and view results directly on the screen. At the price of some overhead, result objects, e.g. histograms, can be monitored while they are being produced. Additional libraries, i.e. user code for processing, can be distributed via the PROOF package feature that allows to compile and load user code on each worker.

Conceptually, PROOF is complementary to the Grid paradigm. The Grid is a large
distributed batch system with the goal to achieving a high throughput on large amounts of data. PROOF allows to execute code interactively and provides fast response on smaller subsets.

PROOF runs as a plugin to xrootd [9] and uses its communication layer. The PROOF schema is shown in Fig. 1. A user starts a ROOT session on a client and connects to a PROOF master that in turn opens a ROOT session on each PROOF worker. The master is the PROOF machine which coordinates the work between the workers, whereas each worker processes one part of the input dataset. A query submitted to PROOF consists of the analysis code (called selector) and the list of files (called chain) to be processed (step 1). Input data are split into packets (each packet consists of a certain number of events) and distributed to the PROOF workers by the a component called packetizer. Data stored on the internal disk of the PROOF worker is processed first, then non-local data, if remaining. When processing completes, the list of individual objects is sent back to the PROOF master and the partial results are merged together (step 3) and returned to the user (step 4). Evidently this requires mergeable results, as in the case for typical ROOT objects like histograms and trees. Custom objects must implement the merging functionality themselves.

3. Current status of CAF

The CERN Analysis Facility is a computing cluster hosted by the CERN IT department and has run PROOF since May 2006. It is planned to provide about 500 CPU cores and 100 TB of local disk space. At the moment it consists of 120 CPU cores and 35 TB of space for local staging. Fig. 2 shows the CAF schema and its interface to external components. The ALICE experiment produces raw and simulated data which are stored on the Grid Storage Elements (SEs) worldwide. For example, one of this is the CASTOR tape-based Mass Storage System at CERN.

The CAF currently comprises 15 8-core machines. One of them is the head node running the PROOF master and the xrootd redirector for file serving. We keep a second node as a backup master. The remaining 13 machines act as PROOF workers and xrootd disk servers at the same time. This allows to benefit from the advantage of processing data that is local to the PROOF worker themselves. Files can be automatically staged from the AliEn SEs on the local disks, but the worker machines can also access Grid files directly. The CAF monitoring is achieved with the help of the MonALISA [10] framework. Users are authenticated to the system via the Global Security Infrastructure (GSI). CAF components and related activities are described in more details in the next sections.

3.1. Monitoring

The CAF is monitored on different levels with the help of MonALISA. Host monitoring, query statistics, user sessions, disk quota and CPU priority can be consulted at http://pcalimonitor.cern.ch in the CAF monitoring section. The host monitoring provides a table of the resource utilization. For each machine the status of the services, the current load, the number of PROOF users, CPU usage, memory consumption, network traffic and number of hosted files with total size is presented. Histories are provided per user session, showing an average of seven connected users. Being the CAF a system for quick interactive analysis, more important session indicators are the connectivity of usage since May 2006 and the peak of 23 users concurrently using the system.

Query statistics are recorded at the end of each query executed by the users. Examples are the amount of bytes 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 display the utilization and performance of the cluster as well as for a CPU fairshare mechanism described further below. Fig. 3 shows the amount of data processed per user group (color coding). The bars indicate the usage per interval (left scale) and the line the accumulated usage (right scale).
Figure 2. The CERN Analysis Facility. It comprises 15 8-core machines configured as PROOF masters/workers and xrootd redirectors/servers. Data can be staged on local disks from the Grid Storage Elements worldwide. Alternatively, remote file access can be performed directly. CAF monitoring is done via the MonALISA framework and user authentication is based on the GSI protocol and LDAP account management.

The time axis is split into two periods corresponding to different CAF setups. CAF1 was the initial cluster consisting of more machines but slower CPUs (33 dual core CPUs), CAF2 is the current configuration described above. The table indicates that the average number of bytes processed has decreased by 21%, which is due to smaller events in recent productions. On the other hand, the number of processed events has increased by 17%.

3.2. Data distribution and staging

The CAF disk space is principally used as a cache space for data imported from Grid storage systems. It is not meant for permanent storage because data for user analysis might change quickly and, moreover, only a subset of the data produced by the experiment may be significant for fast interactive parallel analysis. The CAF provides an automatic mechanism to stage files stored in the AliEn SEs upon user request. Presently about 110000 files have been staged, corresponding to 23 million p+p events with a total size of 8 TB.

The concept of dataset is used to group and process files describing the input for analysis as well as for data staging from the Grid. Datasets have replaced the initial publication of available files in the form of a text file. Data staging is performed through datasets. Datasets are lists of files registered by users for processing with PROOF. They may share the same physical file, i.e. files that are in several datasets are stored only once, allow to keep file information consistent and take care of disk quotas.

Data staging from AliEn is performed by a datastager script that is plugged into the cmsd service on each disk server. Cmsd (cluster management service daemon) is part of the
Figure 3. Data processed in CAF over the last three years with two different cluster configurations. The bars indicate the usage per interval (left axis) and the line the accumulated usage (right axis). The table compares amount of data, events and queries processed.

Scalla/xrootd software suite. A correspondence between AliEn datasets and PROOF datasets exists, i.e. a PROOF dataset can be created from an AliEn dataset. As a user registers a new dataset, the xrootd redirector selects the suited xrootd disk servers and forwards the request to stage each single file. The disk servers send the request to their datastager which performs the staging. When the disk usage reaches a high-water mark (90%), a garbage collector is triggered to delete files with the oldest access time that have not been accessed for a certain period, e.g. one day. This condition preserves the consistency of the datasets. The garbage collection stops when a low-water mark (80%) is reached. The two water marks are currently in use and open for optimization.

Fig. 4 shows the components used to perform data staging from the Grid. The PROOF master registers and remove datasets upon user request. Each dataset description is stored in a file on the master machine. A data manager daemon loops regularly over the datasets and checks whether the files listed in the datasets are available. If files are missing, staging is triggered by sending a stage request to the redirector’s cmsd service. Once a file is staged, meta information is extracted and stored in the corresponding dataset, e.g. the file’s location, the size and the number of the contained events. Files are regularly touched thus preventing to be cleaned by the garbage collector on the disk servers. If a dataset is removed, files are not deleted automatically but just not touched: in this way the garbage collector will permanently remove files after a defined amount of time. This simple approach has the advantage of not requiring to loop over the existing datasets to verify whether a file is used. In case files are part of a new dataset shortly after deletion, these do not need to be re-staged.

3.3. User grouping, quota and fairshare

ALICE aims to assure a fair usage of the available disks and CPUs. Users are grouped into sub-detectors and physics working groups (PWGs). Each user can belong to one or several
groups. Presently about 120 users are registered to be authenticated to the CAF (see later the section on authentication) and they are split among 19 groups. Each group has a quota on the disk which is used to stage datasets from the Grid and a CPU fairshare target (priority) to regulate concurrent queries. The fairshare algorithm adjusts the priorities of concurrently running PROOF sessions to allow the consumed CPU time to reach a certain target per group.

Fig. 5 shows the history of the allocated CPU, while the small overlay represents the structure of the fairshare mechanism: the relatives priorities of the groups that use the system are defined in the CAF configuration, while the consumed CPU time are retrieved from MonALISA. The algorithm measures the difference between usages and quotas and computes the new priorities privileging groups with less usage than the defined quota and punishing groups which are using too much CPU. New priorities are sent to PROOF, which renices the processes accordingly. The graph in the figure shows the percentage of allocated CPU for a period of seven months. The four major physics working groups have assigned the higher CPU priority. Naturally the allocated percentages are meant to regulate the CPU usage when users are running concurrently.

3.4. Authentication
User authentication is based on the Globus Security Infrastructure (GSI) and uses X509 certificates and an LDAP-based configuration management. Grid certificates are used to authenticate the users to the system. In this way, the same mean of authentication is used either for Grid and the CAF. An additional advantage is the possibility to access directly Grid files from the CAF workers. The framework for fast parallel reconstruction of raw data has been recently developed relying on this new feature.
Figure 5. CPU fairshare algorithm. The priorities of concurrently running PROOF sessions are adjusted to allow the consumed CPU time to reach a certain target per group. The input is defined in ALICE as the relative priority of the groups that use the system. The output is calculated applying a correction formula on consolidated usage values retrieved from MonALISA.

4. PROOF on the GRID

High number of potential users, available CPUs and storage resources on the Grid, necessity to stage relevant subsets of data are only few elements which lead to the idea of exploring the usage of PROOF on a larger and dynamic scale. Furthermore it is not feasible, financial and support wise, to provide a computing capacity capable of handling the data volumes produced by the ALICE experiment in one single computing centre. For these reasons, an ongoing development in ALICE aims to extended PROOF to the WLCG paradigm.

The adaptation of the framework to the Grid presents new challenging problems. A not-comprehensive list follows:

- A network of distributed PROOF clusters must be interconnected in a PROOF multi-level hierarchy reflecting the deployment of Grid sites.
- In a large distributed environment like the Grid the data sample to be analyzed is stored in many storage centres worldwide and not in a single place. Tasks and data co-location can be achieved using the Grid middleware classes for asynchronous analysis to split tasks according to the dataset location.
- Grid nodes are firewall protected from incoming connections thus preventing to start PROOF workers from the corresponding PROOF master running on the site front-end machine. These nodes are required to advertise their presence and initiate an outgoing connection towards the PROOF master.
- Worker nodes and data servers may become available or drop out at any point in time, depending on the local availability of resources.
- PROOF is an interactive system, a characteristic that makes it very well suitable for fast code prototyping, whereas the Grid is by implementation a batch system where a job starts running after an unpredictable delay with respect of the submission time.
A prototype of the PROOF framework running across few test Grid sites is sketched in Fig. 6. Each Grid site exposes a front-end machine for a given Virtual Organization (VO) which is called VO-box [11]. A VO-box runs the Grid services, a PROOF master and an additional component called ProxyServer (step 1). The computing nodes of each site, called Worker Nodes (WNs), can communicate only towards the VO-box. The Grid middleware used by our model is the AliEn/gLite software and provides the task splitting capabilities to configure and start pilot jobs (step 2). Pilot jobs are sent to the WNs close to the Storage Element (SE) hosting the dataset to be processed with PROOF. The number of PROOF workers that will be started is a function of the cardinality and location of the user input dataset. Each pilot job has the aim to start a ProxyClient, which is the counterpart of the ProxyServer and connect to it. ProxyClient services are kept continuously running, have a limited lifetime and are used to start PROOF workers. This is the solution to overcome the Grid latency and run PROOF tasks right away across distributed sites. In fact, ProxyClients can immediately start the necessary PROOF workers.

More precisely, each ProxyClient starts an xrootd server from the ROOT package installed at the site. Users might ask for different versions of a given ROOT package: if it is not available on the WN it is automatically downloaded using the Grid Package Manager service (PackMan). Whenever a ProxyClient process is started, it registers itself on the ProxyServer (step 3) and establishes an outgoing connection towards the Grid VO-box. The ProxyServer keeps the list of the ProxyClients running on the WNs (step 4).

As a user initiates a PROOF session (step 5) on a client machine, it is connected to the distributed PROOF cluster and PROOF workers are selected among the available ones. The client connects to a public top PROOF master that coordinates the activity of the local PROOF masters. The selected local PROOF master reads the list of registered nodes from step 3 and, through the xrootd layer and the pair of proxies, starts the needed PROOF workers at the site (thick red line). The top PROOF master has the aim to connect each individual local PROOF master, distribute the load among the PROOF clusters started at the Grid sites and shield the user from the underlying complexity.

4.1. Current prototype
A first prototype of the PROOF framework adapted to the Grid has been under test since early 2009 at CERN and NIHAM (Romania). As a proof of principle, experiment specific analysis tasks have been successfully executed as shown in Fig. 7.

So far, we have generically referred to the term "dataset" in PROOF. A dataset is meant to be a logical collection of files, but we must distinguish between a dataset in a local PROOF cluster and in PROOF on the Grid. In the former case, datasets are registered to stage files from the Grid and process them with PROOF, may share same physical files and allow to keep them consistent on disk. In the latter case a dataset is not used for staging because file consistency is not an easy principle to maintain on federated Grid clusters, not to mention that the Grid file catalogue has already the purpose to register file locations worldwide. Nevertheless, a dataset can be used as a container to trigger data processing on the Grid using the ROOT classes for converting a Grid (AliEn) collection into a PROOF dataset. A user must solely indicate the name of the Grid collection to be processed.

A typical use case consists of querying the AliEn file catalogue from the Grid interface and create an Alien collection. The collection name and the user analysis macro are the only mandatory parameters to be specified. Afterwards, PROOF workers are started at the Grid sites accordingly to the file locations in a number which is a function of the number of files stored in the collection. A user can also specify his own splitting options (e.g. by storage element, file etc..). The AliEn collection is transparently converted into a PROOF collection and the user gets notified as PROOF workers are started.
Figure 6. Adaptation of the PROOF framework to the Grid. A pair of proxy services deployed via the Grid WMS allows for dynamic usage of protected loosely-coupled distributed resources. The usual PROOF architecture master-to-slave is inverted to work through fairly strict firewalls, with the PROOF workers advertising themselves to the local PROOF master running on each site front-end machine (VO-box).

5. Summary

These proceedings present the current status of the CERN Analysis Facility (CAF) running PROOF for the ALICE experiment at CERN. Interactive analysis with PROOF is a valuable addition to local analysis and batch analysis on the Grid. The CAF enables fast interactive data processing on a reasonably large amount of data. A setup is in place since May 2006 and it has been upgraded at the end of 2008. More than 100 users are presently registered and about 10 users per day run concurrently analysis code, but the number of active users can clearly be increased. Along with the setup phase of the system, ALICE has identified desired feature requests and contributed to the PROOF development, examples are user authentication, dataset handling, CPU fairshare, automatic merging of output files and raw data reconstruction.

Furthermore, an exploitation of the Grid resources for quick interactive data processing is also applicable and desired. Main reasons are the impossibility to handle the data volumes produced by ALICE in one single computing centre and the option to run interactive tasks at the Grid sites hosting the data. PROOF allows to configure federated clusters building a multi-level architecture of master nodes, each of them responsible of coordinating the work among its workers. This schema perfectly fits the deployment of the Grid sites exposing a public front-end machine and potentially hiding thousands of CPUs.
Figure 7. Analysis task running in distributed PROOF. The plot displays the Pt spectrum computed out of 26k tracks. The progress window shows some statistics, e.g. the corresponding processing rate (events/sec) and overall time.

6. Acknowledgments
The authors would like to thank Gerardo Ganis for the good teamwork during the PROOF developments, Pablo Saiz for the support during the development of the Grid pilot-jobs and the team for PROOF maintenance at GSI, Darmstadt, for the profitable discussions and ideas on common problems about the adaptation of PROOF to dynamic distributed environments.

References
[1] ALICE collaboration 1995 Technical Proposal for a Large Ion Collider Experiment at the CERN LHC CERN/LHCC/95-71
[2] Foster I., Kesselman C. 1999 The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco
[3] http://lcg.web.cern.ch/LCG
[4] Grosse-Oetringhaus J.F. 2008 J. Phys. Conf. Ser. 119 072017
[5] Ballintijn M., Roland G., Brun R., Rademakers F. 2003 The PROOF distributed parallel analysis framework based on ROOT Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP), La Jolla, California, 24-28 Mar 2003
[6] Saiz P., Aphecetche L., Buncic P., Piskac R., Revsbech J. E., Sego V. 2003 AliEn - ALICE environment on the GRID, Nucl. Instrum. Meth. 437–440 alien.cern.ch
[7] http://glite.web.cern.ch/glite
[8] Brun R and Rademakers F 1997 ROOT: An object oriented data analysis framework Nucl. Instrum. Meth. A 389 81.
[9] http://xrootd.slac.stanford.edu
[10] Legrand I. C., Newman H. B., Voicu R., Cirstoiu C., Grigoras C., Toarta M. and Dobre C. 2004 MonALISA: An agent based, dynamic service system to monitor, control and optimize Grid based applications Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP), Interlaken, Switzerland 27 Sep - 1 Oct 2004
[11] http://glite.web.cern.ch/glite/documentation/userGuide.asp