Interactive Parallel Analysis on the ALICE Grid with the PROOF Framework

Marco Meoni

Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
European Organization for Nuclear Research, 1211 Genève, Switzerland
marco.meoni@epfl.ch
http://people.epfl.ch/marco.meoni

Abstract. The ALICE experiment at CERN LHC is intensively using a PROOF cluster for fast analysis and reconstruction. The current system (CAF - CERN Analysis Facility) consists of 120 CPU cores and about 45 TB of local space. PROOF enables interactive parallel processing of data distributed on clusters of computers or multi-core machines. Subsets of selected data are automatically staged onto CAF from the Grid storage systems. However, a cluster of the size of CAF can only hold a fraction of the yearly 3 PB data accumulated by ALICE. The impracticability to store and process such data volume in one single computing centre leads to the need to extend the concept of PROOF to the Grid paradigm.

Keywords: ALICE, Grid, PROOF, interactive parallel distributed processing, data movement, task colocation, resource availability.

1 Introduction

The ALICE [1] experiment at the CERN Large Hadron Collider (LHC) will accumulate data at unprecedented speed and volume. The yearly estimate is 1.5 PB of raw data from the experimental setup and additional 1.5 PB of reconstructed data and Monte-Carlo simulations. The data management and processing is done through the ALICE Environment [2] (AliEn) middleware on the Worldwide LHC Grid [3] (WLCG [4]), encompassing hundreds of computing centres with many thousands of CPUs and PB scale disk and mass storage systems. A set of unique challenges for the reconstruction and analysis software and the ways the physicists perform data analysis is offered by the data volume and distributed computing environment.

One of the most important aspects of data analysis is the speed with which it can be carried out. Fast feedback on the collected data and publication of results is essential for the success of the experiment. Since several years the ROOT [5] team at CERN is developing a software framework, the Parallel ROOt Facility [6] (PROOF), which addresses the question of synchronous fast data processing

G. Allen et al. (Eds.): ICCS 2009, Part I, LNCS 5544, pp. 114–122, 2009.

on large computing farms. This framework is widely used in High Energy Physics (HEP) and in the ALICE experiment in particular.

The goal of PROOF is to allow for transparent interactive parallel analysis of large sets of files in ROOT format, a common container for data storage in HEP. It is conceived to provide transparency with respect to a local ROOT analysis session (same code can be run locally and in a PROOF system), scalability (no limitations on the number of machines that can be used in parallel) and adaptability (handling of changing of load, disk failure and network cut on the nodes). In this context, by *interactive* it is meant that the user is able to see the results right away, contrary to a Grid job where it is necessary to wait for the job to finish. *Parallel* means that the task is split into subtasks that will be executed on many computing nodes at the same time. Commonly we refer to analysis of *data* that are the result of events reconstruction (so called ESD, Event Summary Data or AOD, Analysis Object Data), but in the future PROOF can be extended to support also large scale simulation.

2 The PROOF system

PROOF is a system 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 or event-based parallelism). It is an interactive system, therefore it can be used

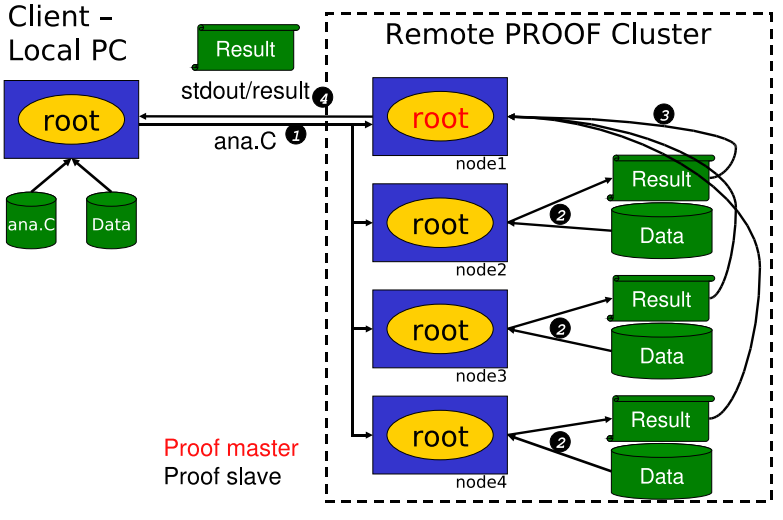

Fig. 1. Schema of the PROOF system. After the user connects to the cluster, a ROOT session is started on each node. The analysis code is sent to the head node (master) and then replicated on each worker. Each worker processes local data and produces a partial result that is sent back to the master node and merged together with the others. The final result is displayed on the user's screen.

from the ROOT prompt thus allowing for direct visualization of results. At the price of some overhead, result objects, e.g. histograms, can be monitored while they are being produced (so-called feedback histograms). Additional libraries, i.e. user code for processing, can be distributed with so-called PROOF packages.

The functioning schema of PROOF is shown in Fig. 1. A user running a ROOT session on her client connects to a PROOF master which in turn opens a ROOT session on each PROOF worker via Xrootd protocol [7]. Then the user submits a query, that consists of the analysis code and the list of files that she wants to be processed (step 1). The PROOF master splits the input dataset into data fragments and distributes them to the PROOF workers. The fragments are assigned in such a way that data local to a worker is processed first, then non-local data, if any remaining. After processing, the partial results are individually sent back to the PROOF master, merged together (step 3) and returned to the user (step 4). This workflow works automatically for mergeable results, that is the case for typical ROOT objects like histograms and trees. To this end, user objects must implement the merging functionality.

3 PROOF Concept on the Grid

PROOF is currently running for the ALICE experiment on a computing cluster called CAF [8] (CERN Analysis Facility) with 120 CPU cores. Naturally, a cluster of this size can only hold a fraction of the 3 PB data to be accumulated by the experiment. It is also not feasible, financial and support wise, to provide a computing capacity capable of handling the data volumes in one single computing centre. For these reasons the concept of PROOF necessitates to be extended to the Grid paradigm - the WLCG. Presently there are a number of research projects aiming at extending the PROOF functionality on the Grid. They are briefly introduced here below to clarify the conceptual and architectural differences as well as the goal each one wants to achieve.

- At INFN Torino, Italy, the Virtual Analysis Facility (VAF) project is currently running PROOF on a Grid Tier-2 (T2) cluster. In Grid terminology a T2 is a mid-size (few hundred CPU) computing centre for user analysis and MonteCarlo production.

  The cohabitation between batch and PROOF processes is achieved by running them on two separate virtual machines on Xen. When an interactive analysis must be prioritized with respect to batch jobs or vice versa, the operating system can directly suspend/restart or slow down/speed up an entire VM, transparently handling the processes memory footprint. Since running a VM requires administrator privileges, this kind of setup can be deployed at the level of a single computing centre, but not at the largest scale (the WLCG Grid) as in our case.

- The ATLAS experiment at CERN, in collaboration with the University of Wisconsin and BNL, USA, is developing a project to run Grid and PROOF services together on the same computing centre infrastructure using the

CONDOR [6] system to handle job priorities. CONDOR is a job scheduler mechanism allowing job submission in a local queue and provides command line API to fully control their behavior at scheduling and running time. As in Torino, the efforts are focused on how to run efficiently batch and interactive jobs on shared resources belonging to a single computing centre.

PROOF is an interactive analysis tool that, as such, may stay idle for a long period of time (for example nights and weekends). CONDOR allows for sharing the available resources between batch-like activities and PROOF, with the capability to get CPUs in a reasonably short time. In case interactive jobs are executed, running batch jobs are suspended freeing the used resources (CPU, I/O, memory) and resumed after the PROOF session is finished.

– At the GSI Heavy Ion Research Center in Darmstadt, Germany, the PROOF on Demand [9] package (PoD) is under development to perform PROOF-based distributed data analysis on the Grid and local batch systems. This successful project is already in production and has many common points with the result we want to achieve in ALICE: the ROOT/PROOF framework is used as a starting point, but the Grid access from ROOT is achieved using the gLite [10] implementation. PoD provides interfaces to submit Grid job scripts executed by the Local Resources Manager System (LRMS) on remote workers. These scripts, comparable to the pilot jobs in our proposal (section "System Architecture"), make environment recognition, upload of the necessary software packages and start the gLite-PROOF services.

  On the other hand there are two considerable differences in respect to our model that prevent its application. First of all the PROOF master is directly started on the user machine, i.e. each user connects to her own master. The choice to separate user environments has the advantage to obtain a robust architecture. The flip side is the assumption that PROOF workers may register themselves on the user machine: this is not certainly feasible on a Grid topology as the client machines are typically forbidden to accept incoming connections or neither allowed to ask for. The assumption a PROOF worker can directly register itself on the client machine leads to a 2-tiers architecture, whereas in our project we must add a gatekeeper on the site's front-end machine (optionally running also the local PROOF master), thus re-creating the 3-tiers architecture of a local PROOF cluster.

  The second difference resides in the dataset distribution. At GSI no assumptions are made on the data location, the access is up to the user code. On the contrary, the location of the file to be processed, that may be stored across several sites, is the keypoint of our computing model to run the code where data is, i.e. "bring the kB to the PB and not the PB to the kB".

In addition to the projects outlined above, PROOF is currently being used by a number of other HEP experiments. This is the case of the CMS collaboration in Oviedo, Spain, with the purpose to adapt the analysis framework to the PROOF model. Similarly, the LHCb collaboration at CERN has the aim to

adapt in PROOF ad-hoc software for analysis, as well as the RHIC experiment at the Brookhaven National Laboratory, USA.

The extension of the PROOF concept to the Grid allows for interactive processing of large volumes of data distributed over hundreds of computing centres and accessible by many thousands of CPUs. The parallelization of the process has the obvious purpose of providing short response times. A number of problems (P) have been clearly identified, as well as proper solutions (S).

Cluster Connectivity

P The distributed PROOF clusters should be interconnected in a multi level hierarchy reflecting the PROOF cluster deployments.

S Each site will run its own PROOF master connected in turn to a general public PROOF superMaster acting as the starting point for the interactive user sessions. The distributed nature of the PROOF setup is hidden.

Tasks and Data colocation

P In a distributed computing environment the data sample to be analyzed will be located at many computing centres worldwide. The analysis tasks must be executed in the computing centre hosting the data, thus avoiding (heavy) data movement.

S This can be achieved by starting PROOF workers at the computing nodes of the centres through Grid jobs. The Grid middleware classes for asynchronous analysis allow for task splitting according to dataset location.

Protected Access

P The computing centres are protected by very strict access rules, implemented through complex firewalls, minimizing the methods that can be implemented for communication between tasks running in separate centres.

S In a classic PROOF local cluster, the head node initiates communications towards all the registered workers (master-to-worker). The worker nodes must be reachable from the PROOF master where the work is initiated. In a Grid topology a PROOF master is very likely running on the front-end machine of each centre, called VO-box, whereas the PROOF workers can communicate only through this VO-box. In this scenario the communication is reversed to a registration service running on the VO-box (worker-to-master).

Dynamic Scheduler

P The PROOF setup must be adapted to the dynamic Grid topology since potential computing workers and data servers may become available or drop out at any point in time, depending on the local availability of resources.

S The PROOF master must be able to connect to workers not only when the user starts a PROOF session. In the current PROOF implementation the number of workers is known a-priori. A dynamic worker allocation feature will require the development of a workload-based scheduler.

Interactivity

P The Grid is by implementation a batch system, where a job runs with a delay with respect of the submission time. With an interactive system, user tasks should start with the initiation of the analysis session.

S A number of Grid jobs will be always kept running in the local computing centres, ready to spawn PROOF sessions. The number of such jobs should be a function of the number of Grid users who start PROOF sessions concurrently and is adjusted automatically.

4 System Architecture

By implementing the above mentioned solutions we can achieve a novel way to run the PROOF setup on distributed resources accessing large data volumes, preserving at the same time the key benefits to run interactive and parallel data processing. Fig. 2 displays the system architecture in a Grid environment.

A ProxyServer service is running on a public port on each VO-box (step 1). This module is installed together with the Grid services deployed on the front end machine of the given site. A number of pilot Grid jobs (step 2) based, among

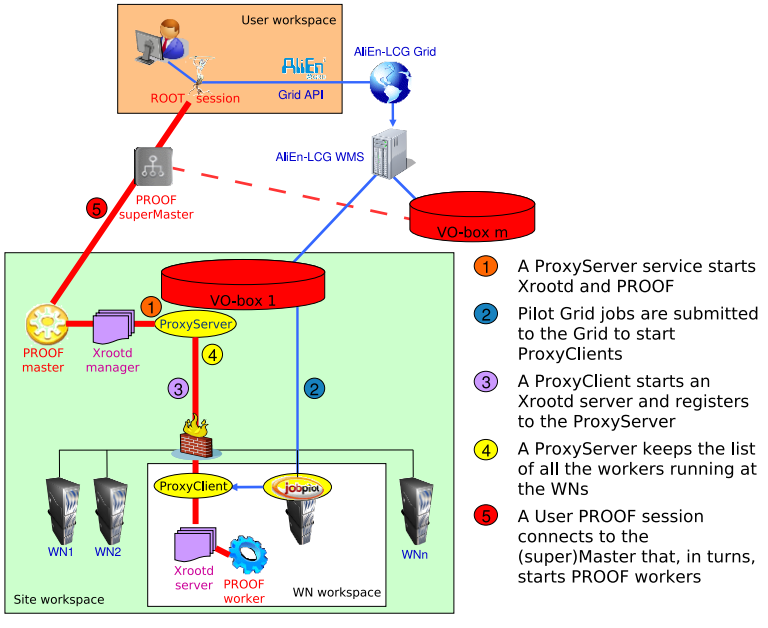

Fig. 2. Schema of the PROOF system on the Grid. New components need to be plugged into the system to allow dynamic usage of loosely-coupled distributed resources. The PROOF architecture master-to-slave is inverted to work through fairly strict firewalls preventing direct connections to computing machines at the remote Grid sites.

the others, on the cardinality and location of the user input dataset, is submitted to the Grid from the user interface with the purpose to start a certain number of ProxyClient, the counterpart of the ProxyServer. Pilot jobs are created by the task splitting capabilities of the Grid middleware and sent to the computing sites close to the Storage Element(s) (SEs) hosting the dataset. The number of pilot jobs is a function of the number of files the user wants to process and their distribution among the SEs.

ProxyClient services already running can be re-used for further user sessions. In such way, the latency of the Grid is hidden because the proxies are kept running once started, ready to serve new tasks. A ProxyClient starts an Xrootd server from the proper ROOT package installed at the site. Users might ask for different versions of a given ROOT/Xrootd package: if it is not present on the Grid Worker Node (WN), it will be automatically downloaded using the Grid Package Manager service. Whenever a ProxyClient process is started on a WN, it registers itself on the ProxyServer (step 3) and establishes an outgoing connection towards the Grid VO-box. The ProxyServer acts as a gatekeeper and keeps the list of all ProxyClients running on the WNs (step 4).

When a user starts a ROOT session (step 5) and connects to the distributed PROOF cluster, the proper workers are selected among the available ones. The local PROOF master at the VO-box accepts connections from the public PROOF superMaster (running on a public machine and coordinating the activity of the local masters) and, through the proxies, connects the PROOF workers at the protected sites (red thick line). The PROOF superMaster distributes the load among the PROOF clusters started at the Grid sites and shields the user from the underlying complexity.

A prototype of the distributed PROOF framework is currently under development and test on two Grid sites at CERN and NIHAM (Romania). These sites store recent ESD data generated by the production cycles of the ALICE Physics Data Challenge 2008 and 2009 (PDC08/09). As a proof of concept, simple analysis tasks processing ESD and reading Monte Carlo tracks have been successfully executed (Fig. 3). The current challenge consists in connecting the individual Grid sites throughout the same session whenever an input dataset is spread over different Grid SEs. Meanwhile, the robustness of the ProxyServer must be improved to support a higher number of connections (tested up to 20 TCP sockets per site). Performance tests are in progress to precisely determine the increment of load sustained by the VO-box with the introduction of the PROOF master. It is well known that this service produces negligible traffic as long as data is processed by the workers (it must only coordinate the job among them) but has peaks of CPU usage and memory consumption during the final merging. The understanding of the PROOF master performance and scalability will come with the integration of the ProxyServer in the MonALISA monitoring system [11] in use in ALICE. The ProxyServer will plug-in into the MonALISA distributed agents already deployed at each Grid site with the advantage to be completely monitored and remotely managed. MonALISA provides the capacity

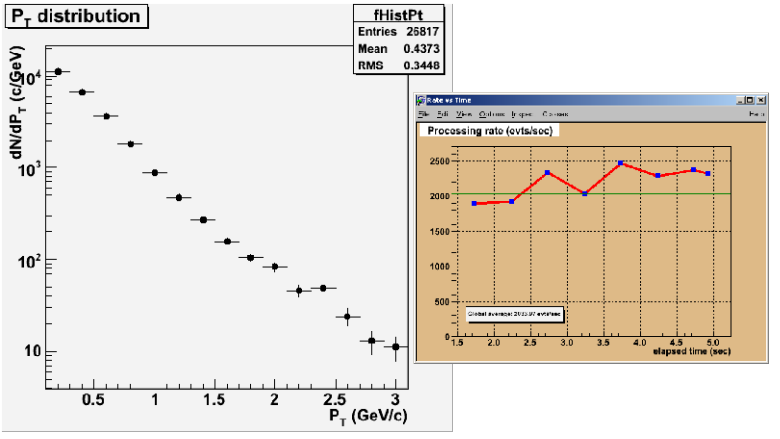

Fig. 3. Analysis task running in distributed PROOF. The plot displays the Pt spectrum out of 26k tracks with the corresponding processing rate (events/sec).

to send monitoring information to the central ALICE Web repository [12] for history view or an interactive GUI client for detailed short-time views.

5 Summary

This contribution presents the approach of the ALICE experiment to enable fast data analysis on the Grid middleware in usage, the ALICE Environment (AliEn). The PROOF framework is primarily meant as an interactive alternative to batch systems for central analysis facilities and departmental workgroups (Tier-2s) in particle physics experiments. However, thanks to a multi-tier architecture allowing multiple levels of masters, it can be adapted to a wide range of virtual clusters distributed over geographically separated domains and heterogeneous machines that form the Grid. The dynamic allocation of a Grid Analysis Facility running PROOF allows to quickly prototype user code that needs many iterations on input datasets, with the advantage given by the availability of the entire data production of the experiment instead of only a small subset locally staged. To this end, new components must be plugged into the system, i.e. a tunnel between the Grid WNs and corresponding VO-box to work through firewall protections, an hierarchy of PROOF masters coordinating the work among the Grid sites, a PROOF workers distribution based on the AliEn capabilities to split jobs according to the data location and a dynamic allocation of PROOF workers.

Acknowledgments

The author would like to thank Jan Fiete Grosse-Oetringhaus for the good teamwork during the CAF setup and maintenance in the ALICE Offline group at CERN, Latchezar Betev and Federico Carminati for the support in this project and review of this document.

References

1. ALICE Technical Proposal for a Large Ion Collider Experiment at the CERN LHC. CERN/LHCC/95-71, December 15 (1995)
2. Saiz, P., Aphecetche, L., Buncic, P., Piskac, R., Revsbech, J.E., Sego, V.: AliEn - ALICE environment on the GRID. In: Nuclear Instruments and Methods 2003, pp. 437–440 (2003), http://alien.cern.ch
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
4. The Worldwide LHC Computing Grid Project http://lcg.web.cern.ch/LCG
5. Brun, R., Rademakers, F.: ROOT - An Object Oriented Data Analysis Framework. Nucl. Instr. And Meth. A 502, 339–346 (2003)
6. Ballintijn, M., Roland, G., Brun, R., Rademakers, F.: The PROOF distributed parallel analysis framework based on ROOT. In: Proc. Conf. for Computing in High-Energy and Nuclear Physics (CHEP), La Jolla, California, 24-28 March (2003)
7. The Scalla Software Suite: xrootd/cmsd, http://xrootd.slac.stanford.edu
8. Grosse-Oetringhaus, J.F.: The CERN analysis facility: A PROOF cluster for day-one physics analysis. J. Phys. Conf. Ser. 119, 072017 (2008)
9. Manafov, A.: PROOF on Demand - An implementation of the PROOF distributed data analysis on different batch systems and/or on the Grid, https://subversion.gsi.de/trac/dgrid
10. http://glite.web.cern.ch/glite/
11. Legrand, I.C., Newman, H.B., Voicu, R., Cirstoiu, C., Grigoras, C., Toarta, M., Dobre, C.: MonALISA: An Agent based, Dynamic Service System to Monitor, Control and Optimize Grid based Applications. In: CHEP 2004, Interlaken, Switzerland (September 2004), http://monalisa.caltech.edu/monalisa.htm
12. Meoni, M.: Monitoring of a distributed computing systems: the Grid AliEn@CERN, University of Florence, Italy, December 19 (2005), http://monalisa.caltech.edu/docs/marco_thesis.pdf, MonALISA Repository for ALICE, http://pcalimonitor.cern.ch