Interactive Analysis using PROOF in a GRID infrastructure

Ana Y. Rodríguez Marrero[1,3], Isidro González Caballero[2], Alberto Cuesta Noriega[2], Francisco Matorras Weinig[1]

[1]Instituto de Física de Cantabria (UC-CSIC), [2]Universidad de Oviedo

[3]arodrig@ifca.unican.es

Abstract

Current high energy physics experiments aim to explore new territories where new physics is expected. In order to achieve that, a huge amount of data has to be collected and analyzed. The accomplishment of these scientific projects require computing resources beyond the capabilities of a single user or group, thus the data is treated under the grid infrastructure. Despite the reduction applied to the data, the sample used in the last step of the analysis is still large. At this phase, interactivity contributes to a faster optimization of the final cuts in order to improve the results. The Parallel ROOT Facility (PROOF) is intended to speed up even further this procedure providing the user analysis results within a shorter time by simultaneously using more cores. Taking profit of the computing resources and facilities available at Instituto de Física de Cantabria (IFCA), shared between two major projects LHC-CMS Tier-2 and GRID-CSIC, we have developed a setup that integrates PROOF with SGE as local resource management system and GPFS as file system, both common to the grid infrastructure. The setup was also integrated in a similar infrastructure for the LHC-CMS Tier-3 at the Universidad de Oviedo that uses Torque (PBS) as local job manager and Hadoop as file system. In addition, to ease the transition from a sequential analysis code to PROOF, an analysis framework based on the `TSelector` class is provided. Integrating PROOF in a cluster provides users the potential usage of thousands of cores (1,680 in the IFCA case). Performance measurements have been done showing a speed improvement closely correlated with the number of cores used.

I. PROOF

The Parallel ROOT[1] Facility (PROOF[2]) allows researchers to analyze and understand much larger data sets on a shorter time scale by enabling interactive executions in parallel on clusters of computers or many-core machines. **The main goal concerning this work is to construct and provide a user friendly environment to fully profit from the PROOF system**.

PROOF is **integrated transparently** with the used local batch system. The framework developed can be used with the CMS analysis software. The final configuration offers:

- Centralized integration between PROOF and the batch system.
- Transparent access to the PROOF cluster for the users.
- An analysis framework to ease the transition from the user analysis code to the required structure by the PROOF usage.

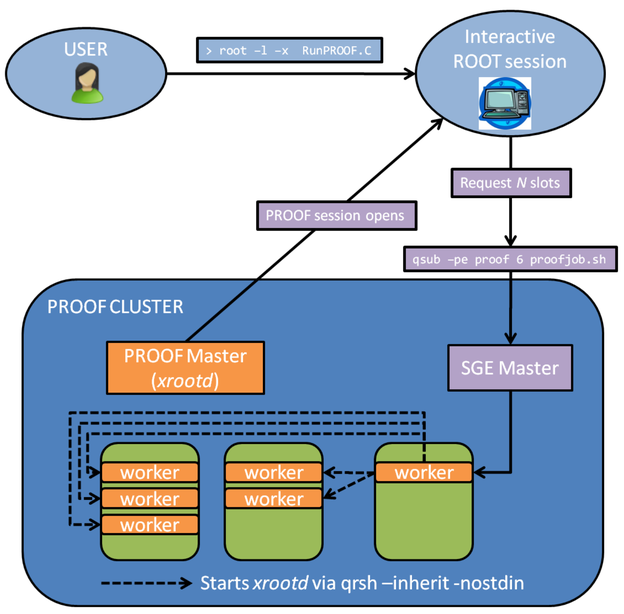
II. PROOF in the batch system

The most common way to set up a PROOF cluster is to configure a set of machines to run the PROOF servers permanently. However, a dedicated PROOF cluster is incapable of adapting to the **dynamic demand of the resources made by users**.

By taking profit of established infrastructures as are the IFCA and Universidad de Oviedo ones, our system provides:

- a fast response for the users
- makes an efficient use of resources by **only requesting the needed nodes**.

Those institutions give support to many users that ultimately execute their jobs at the corresponding batch queues. Local users at these centers profit from an **interactive queue** through which individual PROOF clusters can be dynamically enabled.



When the users start a ROOT session the following **process** occurs:

- A ***startproof*** script initializes and exports all the configuration variables that are needed at the master and workers servers, and submits a single SGE[3] or Torque (PBS)[4] job *proofjob* requiring the number of desired slots that act as workers.
- ***Proofjob*** starts the ***xrootd*** process on the primary worker and spawns through an interactive session the same process at the secondary workers. Once the process is successfully completed, a signal is sent to the client and captured by the user analysis ROOT macro.
- The client connects to the master allowing the start of the PROOF session, and thus the analysis execution using the PROOF facility.
- The master is independent to the batch system, and runs the ***xrootd*** process permanently.

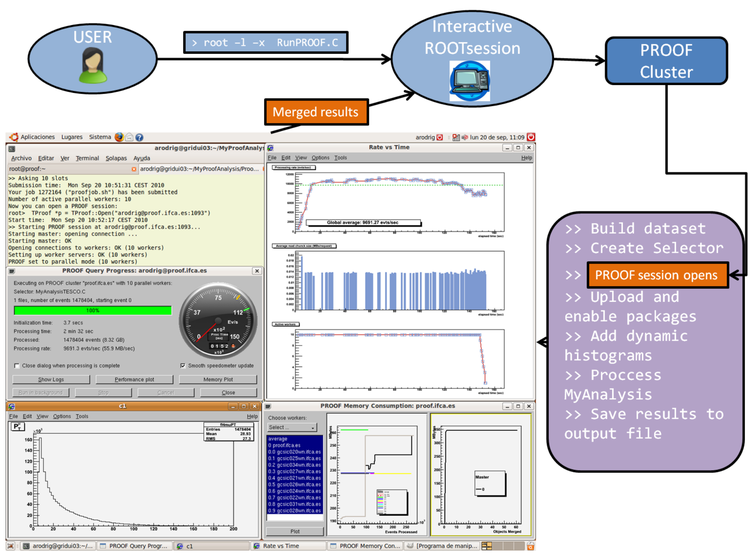
PROOF Cluster **policy**:

- The proof job is submitted to an interactive queue thus the **waiting time to get the slots is of the order of few seconds**.
- **The number of slots assigned by the job manager depends on the current load of the batch system with an upper limit based on the user requirements**.
- The dynamic PROOF cluster is **available during one hour from its initialization. Several PROOF sessions may be run throughout that time period**.

III. A CMS Analysis Framework using PROOF

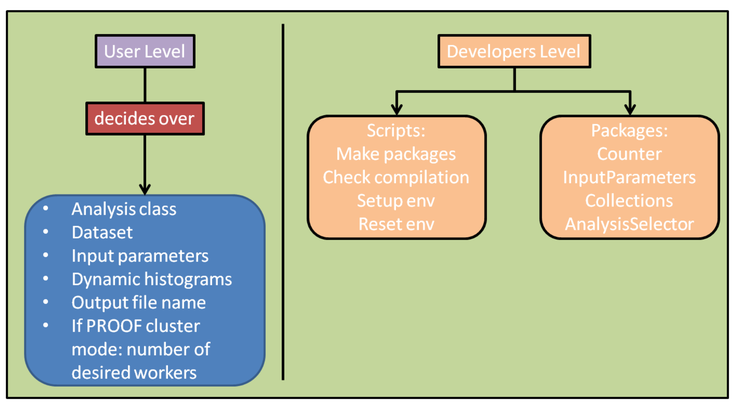
Based on the needs of our users to analyze plain TTree ROOT files with a selection of events and information extracted from the official CMS datasets we have built a user **friendly and flexible PROOF analysis framework to simplify the migration of the large amount of sequential user code** to a PROOF based environment. This kind of HEP analysis is specially suited for PROOF.

The framework is slightly more general providing means to run the exact same code in sequential mode or through PROOF.



We have gone one step forward avoiding the manual recreation of the branches every time the exact content of the data files changes by automatically producing and loading that information. The user can, therefore, concentrate the efforts on the coding of the actual selection and reconstruction algorithms for the analysis.

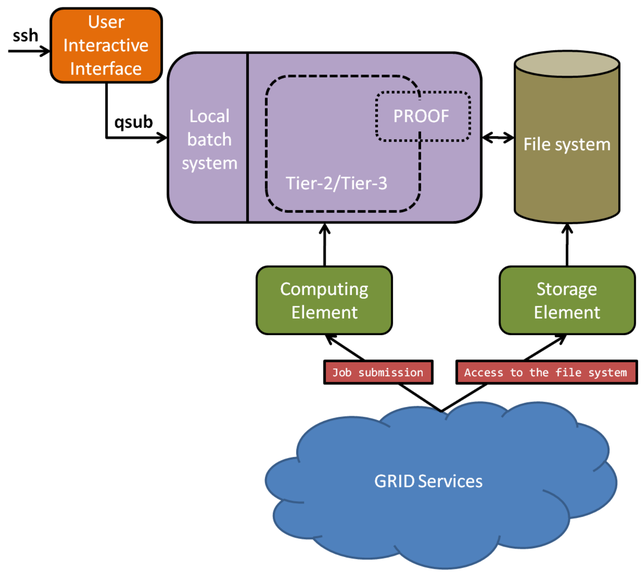
A set of **extra tools** specialized for its usage in PROOF are also provided: counters, input parameters, and collections. The handling of typical analysis objects (histograms, counters, collections) is optimized through dedicated methods in order to take care of repetitive calls.



The configuration of the PROOF session is handled through a single entry point macro. The objects created in the PROOF session are automatically stored in a root file that can be later explored. The users may also select some of the histograms to be interactively plotted as they are filled during a PROOF session. This may be very useful in early spotting mistakes in the analysis code.

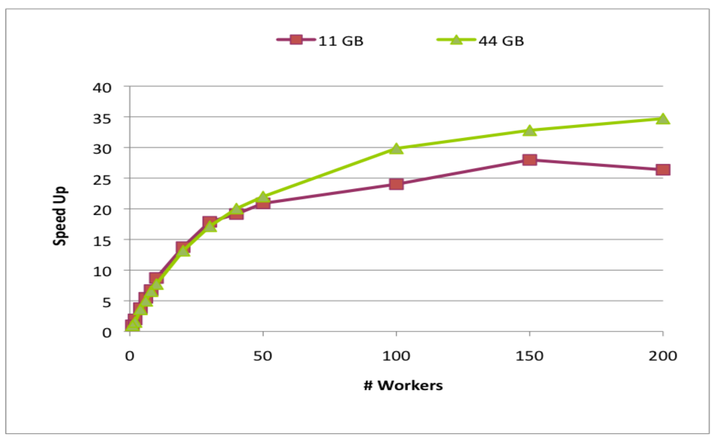
IV. Testbeds: IFCA and Universidad de Oviedo

IFCA[5] and Universidad de Oviedo[6] provide computing resources for the Spanish participation at the LHC-CMS experiment. **IFCA uses the SGE batch system, GPFS[7] as file system and StoRM[8] as storage element**. IFCA counts with **1680 cores** at its GRID cluster, and has 600 TB RAW for CMS storage. **Universidad de Oviedo uses the Torque (PBS) batch system, and Hadoop[9] as file system and storage element**. Universidad de Oviedo counts with 74 slots, and 100 TB RAW for CMS storage. The figure below describes the proof submission jobs to the batch system:



V. Performance

In order to study the performance of the PROOF cluster, several tests have been carried on. An **I/O bounded analysis** that applies cuts to the events and fills histograms was used to study how the application scales with the number of workers at IFCA.



The processing **speeds up almost linearly** with the number of workers up to 20 workers. The processing time decreases from 40 minutes (reading the 44 GB dataset) to 5 minutes for 10 workers and to 1 minutes for 200. More detailed studies are planed in order to understand the speed-up behavior. Preliminary studies at U. Oviedo show similar results.

CPU-bounded analysis are expected to show better performance.

VI. Conclusions

Dynamic PROOF clusters can be currently created at IFCA through the SGE batch system, and at Universidad de Oviedo through Torque.

CMS local users at both institutions have been using this new facility since its integration very recently. The analysis framework favored the migration from sequential analysis to parallel analysis executions.

Preliminary measurements show that the PROOF cluster performs linearly up to 20 workers.

The full setup (configuration files, scripts, etc.) to integrate PROOF into a SGE or Torque (PBS) batch system is available upon request.

References

1. ROOT: http://root.cern.ch/
2. PROOF: http://root.cern.ch/drupal/content/publications/
3. SGE: http://gridengine.sunsource.net/
4. Torque (PBS): http://www.clusterresources.com/
5. IFCA : http://grid.ifca.es/ http://grid.ifca.es/Tier2/
6. Universidad de Oviedo: http://www.hep.uniovi.es/
7. GPFS: http://www-03.ibm.com/systems/software/gpfs/
8. StoRM: http://storm.forge.cnaf.infn.it/
9. Hadoop: http://hadoop.apache.org/