Integrating PROOF Analysis in Cloud and Batch Clusters

Ana Y. Rodríguez-Marrero[1], Isidro González Caballero[2], Alberto Cuesta Noriega[2], Enol Fernández-del-Castillo[1], Álvaro López García[1], Jesús Marco de Lucas[1], Francisco Matorras Weinig[1]

[1]Instituto de Física de Cantabria, Universidad de Cantabria – CSIC, Santander, Spain
[2]Physics Department, Universidad de Oviedo, Oviedo, Spain

E-mail: arodrig@ifca.unican.es

Abstract. High Energy Physics (HEP) analysis are becoming more complex and demanding due to the large amount of data collected by the current experiments. The Parallel ROOT Facility (PROOF) provides researchers with an interactive tool to speed up the analysis of huge volumes of data by exploiting parallel processing on both multicore machines and computing clusters. The typical PROOF deployment scenario is a permanent set of cores configured to run the PROOF daemons. However, this approach is incapable of adapting to the dynamic nature of interactive usage. Several initiatives seek to improve the use of computing resources by integrating PROOF with a batch system, such as Proof on Demand (PoD) or PROOF Cluster. These solutions are currently in production at Universidad de Oviedo and IFCA and are positively evaluated by users. Although they are able to adapt to the computing needs of users, they must comply with the specific configuration, OS and software installed at the batch nodes. Furthermore, they share the machines with other workloads, which may cause disruptions in the interactive service for users. These limitations make PROOF a typical use-case for cloud computing. In this work we take profit from Cloud Infrastructure at IFCA in order to provide a dynamic PROOF environment where users can control the software configuration of the machines. The Proof Analysis Framework (PAF) facilitates the development of new analysis and offers a transparent access to PROOF resources. Several performance measurements are presented for the different scenarios (PoD, SGE and Cloud), showing a speed improvement closely correlated with the number of cores used.

1. Introduction

Interactive analysis of the large data sets produced by the current High Energy Physics experiments is possible thanks to the parallel execution provided by PROOF [1]. This tool executes embarrassingly parallel analysis by distributing the work load among a set of workers deployed as hosts in a cluster or as cores in a single multi-core computer. PROOF is designed to run the user analysis with as little modifications as possible with respect to sequential versions. However, it requires the code to be written following some conventions and requires a working PROOF deployment that will actually run the user analysis.

PROOF is primarily meant to be deployed in a cluster of machines where a *master*, which can be multi-layered, acts as entry point to the facility and distributes the work to a set of *workers*. All these machines must run a properly configured PROOF daemon. The installation and configuration of such clusters requires certain level of expertise and understanding of the system, mostly due to the flexibility of the PROOF system that allows it to adapt to a wide range of resource setups. Alternatively, PROOF-Lite is a dedicated version of PROOF that takes advantage of multi-core machines, but is limited to a single host.

Tools like PoD [2] or PROOF Cluster [3] free the user from the setup and configuration of the system by providing a dynamic construction of the PROOF infrastructure using a batch system. This integration of PROOF into existing production services forces the adaptation of the system to specific software configurations, which may not completely fulfill the requirements of the user. Moreover, sharing the hosts with other jobs may cause a degradation of performance and can lead to problems in the configuration of PROOF. The dynamic nature of the analysis and the need for isolation make PROOF a typical use-case for cloud computing, where virtual machines are started on demand to create a PROOF cluster that can be accessed interactively with the exact software configuration required and without any interference from other jobs or users.

The PROOF Analysis Framework (PAF) [4] was created to allow users to run their analysis with PROOF without caring about the PROOF details. The framework has two main objectives: hide the PROOF details as much as possible, so the researcher can concentrate on the analysis development and not on the setup of the infrastructure, and provide ready-to-use skeleton and tools for developing new analysis or easily migrating existing ones that can be run in parallel or sequential modes. This work describes the PAF backend that is in charge of deploying the PROOF infrastructure on behalf of the user. This backend is able to use several platforms: PROOF-Lite, batch systems (using PoD and PROOF Cluster) and Cloud.

2. PROOF Analysis Framework Deployment

PAF transparently handles the automatic deployment of the PROOF cluster for the user providing a fast response while trying to make an efficient use of the resources. PAF supports five different deployment modes, that the user selects according to the resources available:

Sequential. This mode runs the code without any parallelization. It is intended for debugging purposes only and allows to isolate any potential problems caused by the analysis code from the PAF tools.

Lite. The Lite mode starts the PROOF-Lite environment on a single machine. It allows to perform analysis of small data sets on login machines or desktops when no computer farm is available.

PoD. In this case, PAF uses the PoD tools to deploy the cluster in a batch system. The master daemon is created dynamically in the user machine, which can cause a bottleneck when merging large files or a big number of users are sharing the same machine for running analysis.

Cluster. The Cluster mode uses a fixed master configured and tuned for the specific characteristics of the site where it is installed. The workers however are dynamically configured and started by submitting jobs to a SGE [5] or Torque [6]/PBS [7] batch systems.

Cloud. This new experimental mode creates complete PROOF clusters on demand, both master and a set of workers, by starting virtual machines using a IaaS (Infrastructure as a Service) Cloud platform.

Cluster and Cloud deployment backends are completely developed by the IFCA and Universidad de Oviedo teams and detailed in the following sections.

2.1. PAF Cluster

The PAF Cluster mode takes profit of the existing production infrastructure at IFCA and Universidad de Oviedo. Those sites give support to both grid and local users that ultimately execute their jobs at the corresponding resources, managed by a batch system scheduler.

Under this schema, a dedicated host acts as master of the PROOF farm. This master is independent of the batch system and is the fixed entry point that PAF uses for establishing the analysis sessions. The workers are created by submitting jobs requesting several slots to the batch system. This job configures and starts the daemon on all the allocated slots. Once the daemons are started they are added to the list of workers for the user and a signal is sent to PAF, that can connect to the master to execute the user code. Data is stored in a shared file system mounted by all nodes. Fig. 1 shows the process flow for creating a PAF Cluster session.

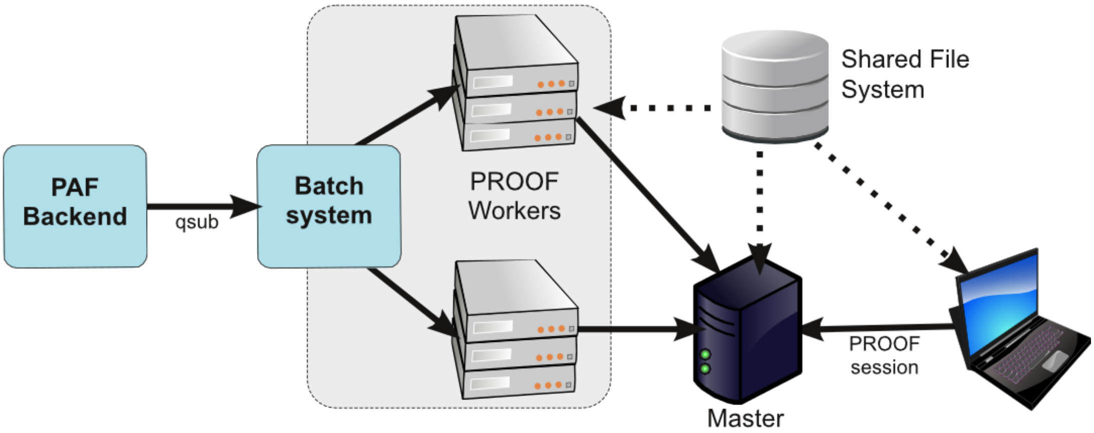

Figure 1. Deployment of PAF on a Batch Infrastructure.

In order to improve the service level of the PAF Cluster users, the batch policies are tuned to enhance the interactive experience:

- PROOF jobs are limited to two hours. This time is enough to execute most of the analysis even with large data sets. PAF reutilizes whenever possible existing jobs within those two hours, and users are warned when only a few minutes are left. The job limitation policy enables the use of backfilling in the batch system and avoids starvation of batch jobs by freeing resources when they are not needed anymore. Users may finalize the PAF jobs at any time by invoking the *endproof* command.
- A range based slot allocation policy is enforced. PAF sets an upper limit for the number of slots to allocate in a single job and the batch system using a fair use policy will adapt the actual number of slots given to the job to the current load.
- PROOF jobs have a high priority and are started within seconds from the submission. If it is not possible to allocate resources for the job immediately, batch low priority jobs running in the cluster may be suspended.

2.2. PAF Cloud

The Cloud mode takes advantage of the ability to start virtual machines on demand provided by IaaS Clouds, like the pilot OpenStack [8] service at IFCA. The current implementation is based on OpenStack and the OpenStack API, but its architecture is general enough to be adapted to similar public or private Cloud offerings, such as Amazon EC2 [9], with minor effort.

The Cloud mode assumes that the following services are available:

- A Virtual Machine (VM) Image repository, where golden copies of the VMs to be executed are stored. For the use of PAF, a VM image was created with Scientific Linux 5 and a complete ROOT installation with the PROOF daemons.

- A Volume server, that provides permanent storage that can be attached to the running VMs. Data to be analyzed by the users is stored in such volumes. PAF maintains a list of available data sets and the volumes where they are stored.
- A Virtual Machine Orchestrator, that allows users to start, stop and manage instances of VMs. PAF contacts this service for setting up the PROOF cluster by instantiating VMs that are taken from the image repository. The system provides a set of VM *flavors*, where each flavor has a fixed number of CPU cores, RAM memory and disk space. The number and characteristics of these flavors depend mostly on the capabilities underlying physical infrastructure.
- A Network Management Services for the assignment of public IPs to running instances and the provisioning of VPNs to connect securely to those instances. The current PAF implementation assumes a VPN is readily available and the user machine is correctly configured to use it, but it might be easily enhanced to request a public IP and assign it to the PROOF master.

Fig. 2 depicts the process for deploying a PROOF cluster in the Cloud infrastructure. Once the user selects the Cloud mode and starts a PAF analysis the next steps are taken:

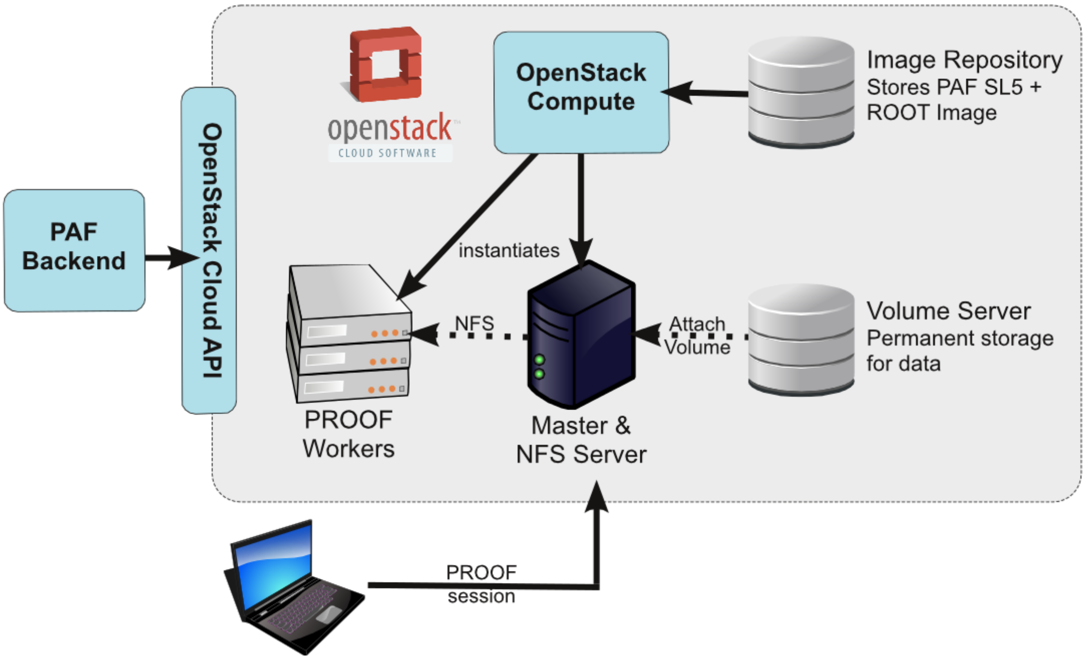

Figure 2. Deployment of PAF on a Cloud Infrastructure.

- A master VM is instantiated in the cloud infrastructure. Currently, this VM is instantiated using a flavor that has a single core and 2GB of memory.
- The volume (or volumes) containing the data to be analyzed is attached to the master VM. Once attached, it is mounted in the VM.
- A set of worker VMs is instantiated in the the infrastructure. Worker VM are 8-core machines with 14GB of RAM. The number of workers to start is determined by the user and is limited to a maximum of 32 cores per session in order to avoid abuses.

- A NFSv4 server that exports the data from the volumes is configured and started in the master VM. This filesystem is mounted at all the worker VMs and data becomes available at all the VMs.
- The PROOF cluster is created by building a configuration file that contains both the master and worker VMs, and then starting the daemons at each VM. The system is ready for accepting connections from the clients. The IP of the master is returned to PAF and the analysis session can start.

Users can reuse their PROOF infrastructure during several sessions. Once they have finished all their analysis tasks, an *endproof* script terminates the instances and frees all the resources.

3. PAF in Production

Users have incorporated PAF in their daily analysis routine using Cluster and PoD modes on the computing infrastructures at IFCA and Universidad de Oviedo respectively. Both centers provide resources for the Spanish participation at the LHC-CMS experiment [10]. IFCA uses the SGE batch system for managing more than 2000 cores. All the nodes in the cluster mount a GPFS [11] shared file system that contains user homes and the CMS data. Universidad de Oviedo has a 100 cores cluster managed by Torque and a Hadoop [12] filesystem for the data.

The Cloud Pilot at IFCA is currently at a testing phase for advanced users and consists of a OpenStack service which manages 8-core hosts configured with the Xen hypervisor for running the Virtual Machines. No shared filesystem is provided by the infrastructure, but as shown previously, a NFS server is used for sharing the data between the PROOF VMs.

PAF is used as the main framework for the final stages of the physics analysis carried out by the researchers involved in the CMS Collaboration at IFCA and Universidad de Oviedo. Fig. 3 shows the performance of a PAF analysis on real data sets from the CMS top physics group. Two different samples of events, with 19 GB and 38 GB were used, for each of the computing infrastructures available: PAF Cluster at the SGE production system at IFCA (labeled as IFCA), PAF PoD at the Universidad de Oviedo Torque cluster (labeled as Oviedo) and PAF Cloud using the OpenStack resources at IFCA (labeled as Cloud). The scalability of the analysis is limited by the I/O bandwidth and no performance gains are found beyond 20 nodes. Users are satisfied with the time reduction of their executions, that allows them to analyze big amounts of data in a few minutes.

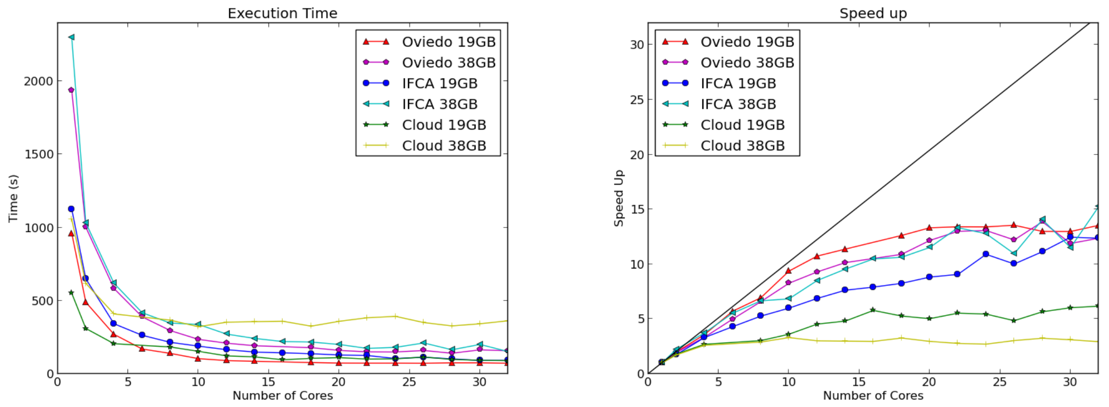

Figure 3. Time spent (left) and speedup factor (right) using PAF over a different number of workers.

4. Conclusions

We have developed a framework based on PROOF to facilitate the parallel execution of the researchers codes taking profit from the existing computing infrastructures at the sites. The framework handles transparently the automatic and dynamic deployment of the PROOF system for the user in different types of resources. PAF provides an environment for performing interactive analysis of large data sets using the production infrastructures of IFCA and Universidad de Oviedo. Private cloud services, like the OpenStack installation at IFCA, can also be accessed using PAF thanks to the Cloud mode that creates the PROOF cluster on top of virtual machines.

CMS local users at both institutions have been using PAF since its early integration, and they are satisfied with the facility. The outcome of the analysis using PAF have ended in the publication of results.

References

[1] Ballintijn M, Biskup M, Brun R, Canal P, Feichtinger D, Ganis G, Kickinger G, Peters A and Rademakers F 2006 *Nucl. Inst. & Meth. in Phys. Res. A* **559** 13–16

[2] Malzacher P and Manafov A 2010 *J. Phys.: Conf. Ser.* **219** 072009

[3] Rodríguez Marrero A Y, González Caballero I, Cuesta Noriega A and Matorras Weinig F 2011 *J. Phys.: Conf. Ser.* **331** 072061

[4] González Caballero I, Rodríguez Marrero A, Fernández del Castillo E and Cuesta Noriega A 2012 A PROOF analysis framework, *to be published at the Proc. for CHEP2012*

[5] Son of Grid Engine (SGE): https://arc.liv.ac.uk/trac/SGE

[6] TORQUE: http://www.adaptivecomputing.com/products/open-source/torque

[7] Portable Batch System (PBS): http://www.pbsworks.com/Product.aspx?id=1

[8] OpenStack: http://openstack.org

[9] Amazon Elastic Compute Cloud (EC2): http://aws.amazon.com/ec2

[10] The CMS Collaboration 2008 *J. Inst.* **3** S08004

[11] IBM General Parallel File System (GPFS): http://www.ibm.com/systems/software/gpfs

[12] Hadoop distributed file system: http://hadoop.apache.org