PROOF Analysis Framework

J. Delgado Fernández[1],E. Fernández Castillo[2],I. González Caballero[1],A.Y. Rodríguez Marrero[2]

[1]U. de Oviedo – Spain, [2]Inst. de Física de Cantabria (UC-CSIC) – Spain

Introduction

The PROOF Analysis Framework (PAF) is a PROOF based tool for the **last phases of a typical HEP analysis** when the data is usually stored in the format of ROOT flat (or close to flat) trees. PAF is designed to make the use of possible **distributed computing resources** easy. Therefore, the same PAF based analysis code can be run on a single node, a cluster or a cloud thanks to the transparent integration of tools like PROOF Lite, PROOF on Demand (PoD), PROOF Cluster or PROOF Cloud. In the latest release we have added a higher level of **modularity** to further enhance the ability of analysis developers to **collaborate**. We have also re-implemented most of the code to improve PAF **performance** in view of the higher needs of LHC Run-II.

Design

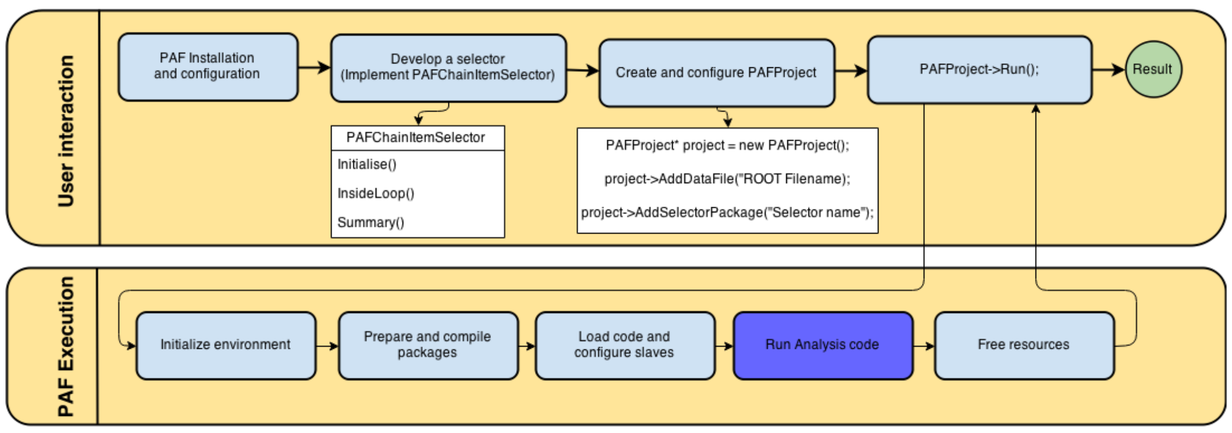
PAF common code can be downloaded, installed and configured for **personal use** or in a **central location** through a well-defined documented procedure. Its modular design provides high adaptability to new requisites that may arise in the future. We have developed a clean and pure object oriented framework that can adapt to several different workflows and user preferences.

The code has been structured in four modules: computing, environments, utilities and settings.

- **Computing**: Core computing code, base project and analysis classes.
- **Environments**: Integration of PoD, PROOF Cluster, ...
- **Utilities**: Additional PAF tools to improve the user experience.
- **Settings**: PAF requires some parameters to be configured to be able to run. Some of these settings are the path to the PAF binaries and libraries, the path to additional package repositories or the behavior of the logger. These settings are encapsulated in a strategy pattern which although, by default, takes these values from environment variables, it can be easily extended with additional mechanisms.

Workflow

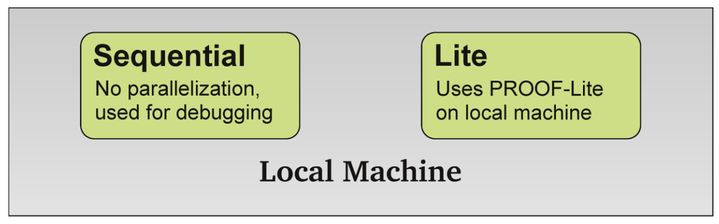
In the chart below a schematic representation of a **normal analysis development cycle** with PAF is shown. PAF installation and configuration should be done once and may be reused as many times as needed. The actual physics analysis code is encapsulated into at least one selector or analyzer that can be enriched with more packages and modules. All this is added to a project in a ROOT macro that once executed sets up the selected environment, transparently prepares and compiles all the code, loads it into the slaves, and starts the distributed processing.

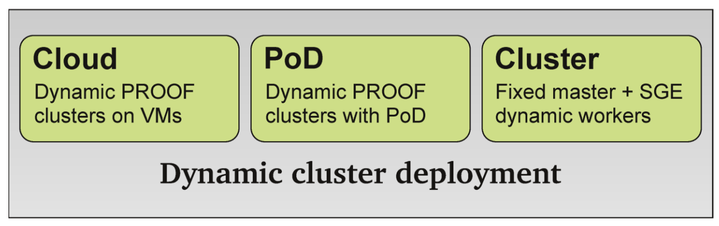


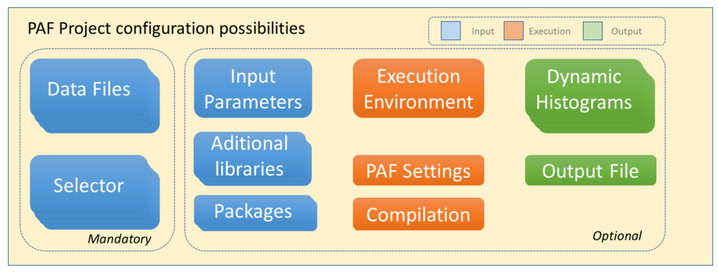
Features

A **large number of functionalities** have been implemented in PAF while trying to keep the configuration task as **simple and automatic** as possible. It has been developed with the idea of getting the **best performance with an easy way to interact with it**.

- **Modular analysis framework**: In our latest release, we have introduced the concept of a project made of several analyzers providing easy communication abilities to all of the elements. This way specialized analyzers (ex: Electron selector, MET estimator,...) can be easily shared among different analysis developers.
- **Processing environments**: Including sequential processing, PROOF Lite and other dynamic PROOF cluster builders (PoD, PROOF Cluster, PROOF Cloud). To change from one to another just a single line needs to be edited. New environments can be easily added.
- **Transparent lazy loading** of branches: A smart dynamic mechanism allows PAF to instruct ROOT to only load those branches used in the analysis resulting in processing rates up to 10 times faster depending on the complexity of the analysis.
- **Support for local or distributed package compilation**: Through a single switch PAF can be instructed to compile analysis packages locally (ideal for homogenous environments) or in the slaves (needed in the case of heterogeneous clusters).
- **Logger**: We have developed an extensible logger to improve PAF runtime information.
- **Dynamic histograms**: One or several histograms produced in the analysis can be selected to be plot during the data processing.
- **Homogenous and efficient variable passing**: A simple mechanism has been implemented to pass information from the main process to the analyzers, and between analyzers.
- **Tree inspector**: An independent tool to easily inspect and find out the content (branches) in a ROOT file The tool provides cut and copy code very useful in the development of a new analysis class.







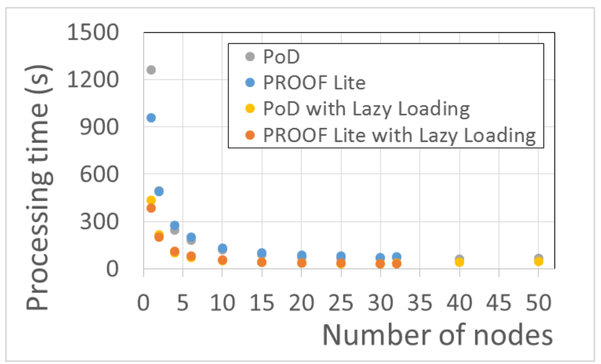
Performance

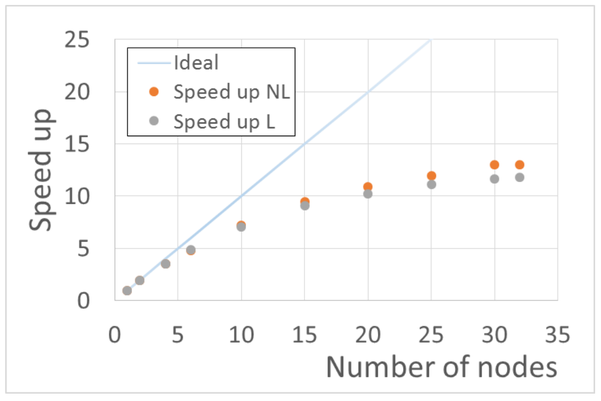
Our testbed is made of HP DL360p Gen8 servers with 2 Xeon E52650 (2.00GHz) connected to a HUS 110 storage server. A typical CMS diboson (WZ) cross section estimation analysis code was run over a 23 GB Z+Jets sample.

We show on the left the **improvement obtained by the lazy download functionality** as well as the *speed up* measured with PROOF Lite and PoD environments. The scalability of the **processing time** shows a reasonable trend for a small number of slaves, while it fails at higher values due to the limits of the hardware configuration.

The **initialization time grows linearly** with the number of nodes. PROOF Lite is 5-10 times faster in this phase since everything happens locally. Setting PoD can take up to 10 seconds but PoD dynamic clusters can be reused.

The time spent in code preparation, compilation and loading shows no correlation with the number of nodes and amounts to about 10 seconds.





Conclusions

- PAF has been **successfully used in the analysis of the LHC Run-I data** taken by the CMS detector.
- The new implementation enhances the previous version providing **better scalability, flexibility and performance**.
- The **modularization** capabilities allow for better sharing of analysis algorithms.
- Further improvements will aim at additional **reduction of the initialization and compilation time**, support for other tree formats and better handling of new workflows.
- Support for **ROOT 6** will also be provided in the short term.
- More information can be found in:

http://www.hep.uniovi.es/PAF

Universidad de Oviedo