PARALLEL PROCESSING OF LARGE DATA SETS IN PARTICLE PHYSICS

MARINA ROTARU[1], MIHAI CIUBĂNCAN[1], GABRIEL STOICEA[1]

[1]"Horia Hulubei" National Institute for Physics and Nuclear Engineering,
Reactorului 30, RO-077125, P.O.B.-MG6, Măgurele-Bucharest, Romania
E-mail: Marina.Rotaru@nipne.ro, Mihai.Ciubancan@nipne.ro, Gabriel.Stoicea@nipne.ro

Received December 15, 2014

The analysis of the LHC data aims to minimize the vast amounts of data and the number of observables used. After slimming and skimming the data, the remaining terabytes of ROOT files hold a selection of the events and a flat structure for the variables needed that can be more easily inspected and traversed in the final stages of the analysis. PROOF has an efficient mechanism to distribute the analysis load by taking advantage of all the cores in modern CPUs through PROOF-Lite, PROOF Cluster or PROOF on Demand tools. In this paper we compared performance of different methods of file access (NFS, XROOTD, RFIO). The tests were done on Bucharest ATLAS Analysis Facility.

Key words: particle physics, PROOF, protocols, data analysis.

1. INTRODUCTION

The increasing complexity of high energy physics (HEP) detectors and the high luminosities achieved in colliders like the LHC produce large amounts of data in the range of several tens of petabytes per year (PB/year). During the first LHC (Large Hadron Collider) run, the ATLAS experiment [1] collected more than 140 PB of data. By filtering the real data according to some trigger or combination of triggers, and by splitting the MC (Monte Carlo) samples according to the process simulated each analysis can use only those data sets that are relevant to them. The output of the reconstruction of both experimental and simulated data is the Analysis Object Data (AOD). The total size of a single version of AODs is of the order of PBs. The AODs are used as primary format for analysis by about a quarter of ATLAS physicists. From AODs, the physics groups derive reduced data formats, Derived Physics Data (DPD) [2] are the result of dedicated slimming, skimming and thinning procedures. Skimming is defined as the reduction of events, whereas slimming and thinning involve the reduction of objects. The most common derived format is the D3PD, a ROOT [3] ntuple. The total size of a single version of D3PDs is also of the order of PBs. Given the huge rate of events collected at the LHC, the processing time of these ntuples can be very large and depends on the complexity of the physics events under study. Also the final analysis has to be run for a significantly high number of times in order to tune cuts and to make histograms and graphs. All these characteristics can result in

a very large processing time. It is therefore very important to have the possibility to fully exploit the local processing resources with the maximum efficiency using all the processor cores. To this purpose some tests have been done on the PROOF (Parallel ROOT Facility) [4] architecture using Bucharest ATLAS Analysis Facility (BAAF; for the detailed description see [5], [6]).

This paper is organised as follows. Section 2 describes the parallel ROOT facility in high energy physics. Section 3 illustrates different protocols for accessing files. The results are presented in sections 4. Finally, the conclusions are given in section 5.

2. PARALLEL ROOT FACILITY

Parallelization [7] of an application is not an easy task and several issues should be taken into account. In this paper parallelism is achieved at the level of events, the computing load is distributed among the available computing units by having them perform the whole set of operations over a given subset of the events. The algorithms applied to each event are not parallel themselves. PROOF is the parallel facility integrated in the ROOT toolkit. It provides a simple framework to traverse ROOT main collections (TTree) in parallel with a small effort. PROOF has been designed with the following goals in mind:

- Transparency: the analysis code should need as little modifications as possible to be run on PROOF with respect to being executed sequentially.
- Scalability: the basic architecture should not put any implicit limitations on the number of computers that can be used in parallel.
- Adaptability: the system should be able to adapt itself to variations in the remote environment (changing load on the cluster nodes, network interruptions, etc.).

PROOF is therefore a natural solution to speed and improve the last stages of high energy physics data analysis.

One of the most important characteristics of a parallel system is the way it handles load balancing. PROOF implements a load balancing mechanism that dynamically distributes the load based on the performance of each computing unit. Since data will not be replicated in each worker node, rather split over them, it needs to be efficiently accessed remotely. ROOT supports several protocols such as RFIO, XROOTD, http and posix compliant file system. That means that most of the storage systems used in HEP can be used from PROOF. A wrong or poor selection of the storage technology can produce a bad performance for a PROOF based analysis. Special attention should be paid to this aspect when designing the analysis strategy. Finally, merging the results from the worker nodes might be an annoying task of any

distributed system. PROOF takes care of merging objects from the worker nodes provided they inherit from the ROOT base class (TObject) and implement the right method. This is the case for all main ROOT objects like histograms and collections.

PROOF consists of a 3-tier architecture, the ROOT client session, the PROOF master server and the PROOF slave servers. The user connects from the ROOT session to a master server on a remote cluster and the master server in turn creates slave servers on all the nodes in the cluster. Queries are processed in parallel by all the slave servers. Using a pull protocol the slave servers ask the master for work packets, which allows the master to distribute customized packets for each slave server. Slower slaves get smaller work packets than faster ones and faster ones process more packets. In this scheme the parallel processing performance is a function of the duration of each small job, packet, and the networking bandwidth and latency. Since the bandwidth and latency of a networked cluster are fixed the main tunable parameter in this scheme is the packet size. If the packet size is chosen too small the parallelism will suffer due to the communication overhead caused by the many packets sent over the network between the master and the slave servers. If the packet size is too large the effect of the difference in performance of each node is not evened out sufficiently. This allows the PROOF system to adapt itself to the performance and load on each individual cluster node and to optimize the job execution time.

The PROOF-Lite is the case of a user having a powerful private multicore machine - a desktop or a laptop - where the user would run PROOF-Lite to analyze the data or to perform the tasks needed by the analysis. In addition to setup ROOT - and therefore PROOF-Lite - what one needs in this case is the ability to install smoothly the additional code required, for example, by the experiment and to be able to access the data, which typically are stored remotely.

The standard PROOF is the case of a cluster of - typically homogeneous - worker nodes, with a non-negligible amount of local storage. Also in this case access to potentially available software binaries needs to be smooth. However, the master, *i.e.* the node chosen to be the entry point to the system, will serve also as build/retrieve machine and will make binaries available to the workers.

PROOF on Demand [8], PoD, is a toolkit to set up PROOF on any resource management system.

3. DATA ACCESS IN LAN

One of the biggest challenges in LHC experiments at CERN is data management for data analysis. Event tags and iterative looping over data sets for physics analysis require many file opens per second and (mainly forward) seeking access. Analyses will typically access large data sets reading terabytes in a single iteration. A large user community requires policies for space management and a highly per-

forming, scalable, fault-tolerant and highly available system to store user data. While batch job access for analysis can be done using remote protocols experiment users expressed a need for a direct filesystem integration of their analysis (output) data to support file handling via standard Unix tools, browsers, scripts, etc.

There are several ways to access data on the disk storage from a job running on the worker node, WN. One is a staging, in which a job downloads a file (or multiple files) onto the WN from the storage before starting processing. This method is reliable, but needs a large disk space on the WN and it takes time to download. Another method is the use of the direct I/O from an application. The RFIO and the XROOTD can be used from an application with the client library. Partial file reads can be done with this but could be inefficient due to a read-ahead feature. The Network File System (NFS) [9] is a traditional service to allow a remote file access.

Solutions for efficient remote access to data exists and are very discussed. One of the most favored in conjunction with PROOF is the one based on SCALLA/ XROOTD [10], an efficient remote file server system, providing Virtual Mass Storage capabilities and is also adopted as data access protocol for LHC. In PROOF clusters, XROOTD is used for remote files direct access.

RFIO [11] is the protocol used by physics analysis code before XROOTD in order to read data stored within a DPM (Disk Pool Manager [12], [13]) instance and is one of the protocols used to perform the tests in this paper.

4. RESULTS

To examine performance of the file access between RFIO, XROOTD, and NFS using PROOF on the BAAF, a PROOF enabled analysis has been run several times on PROOF-Lite or on PROOF clusters. The analysis used for these tests is based on RootCore framework [14]. RootCore is the package which just provides a common infrastructure to be able to easily use different packages, compile them with one command and link the analysis package against them. The benchmark analysis is the general search for new phenomena with the ATLAS detector in proton-proton collisions at $\sqrt{s} = 8$ TeV [15]. It is a cut based analysis which reads ntuples (ATLAS D3PDs) of the ATLAS SUSY stream. In this analysis 350 out of 3300 branches are read. The analysis performs cuts and combinatorial operations and creates several histograms and a tree with 200 branches as output. The D3PDs from the ATLAS SUSY stream are flat ROOT trees partitioned in ROOT files of ≈ 700 MB. In the tests presented, two different dataset sizes have been used. The smaller dataset consists of 60 files with 600,000 events in total (66 GB), the larger dataset consists of 410 files with 4,100,000 events (500 GB). The data files were read from the NFS server or from the grid site RO-02-NIPNE for the tests.

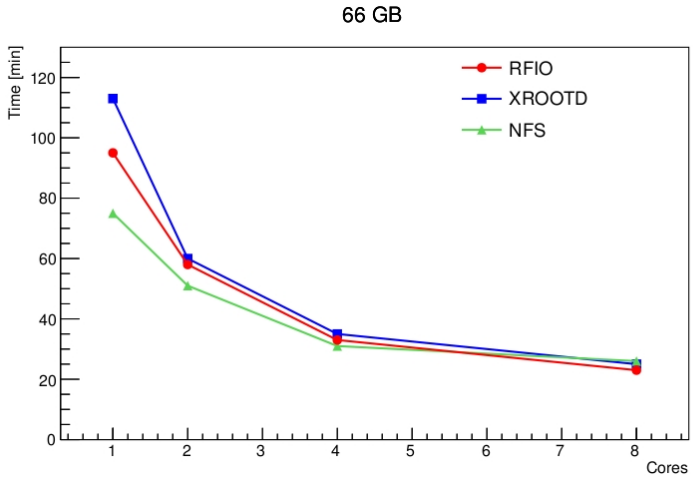

Fig. 1 – Execution time per protocol required for 66 GB data set analysis using PROOF-Lite.

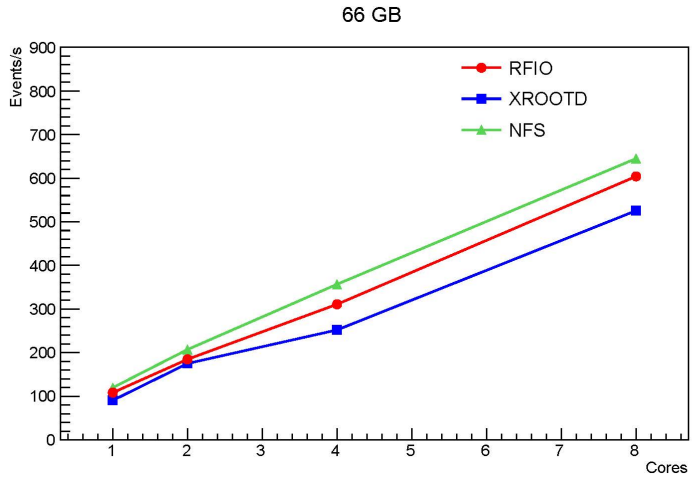

Fig. 2 – Processing rate as function of the number of cores for 66 GB data set analysis using PROOF-Lite.

4.1. SMALL SIZE DATA SET

Figure 1 presents the execution time of the analysis as a function of the number of processing cores using PROOF-Lite. For one and two cores, a better time of processing is obtained when we acces files from NFS server, but for 8 cores, the time is similar for all the three protocols used.

First performance measurements show a very good and efficient scalability for

NFS and RFIO, but a poorer performance with XROOTD, Fig. 2.

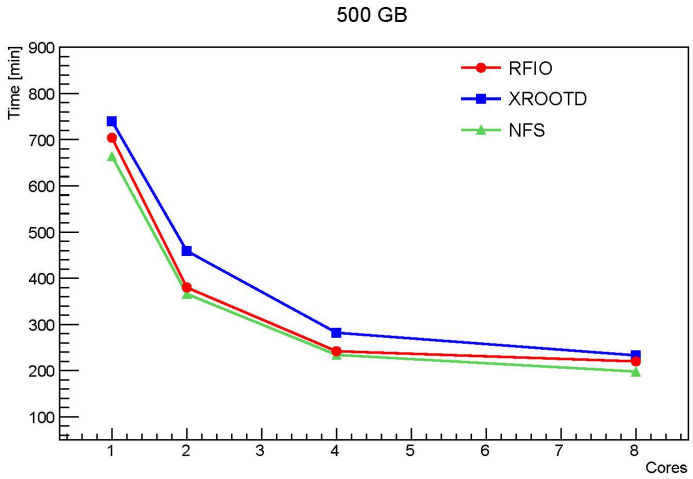

Fig. 3 – Execution time per protocol required for 500 GB data set analysis using PROOF-Lite.

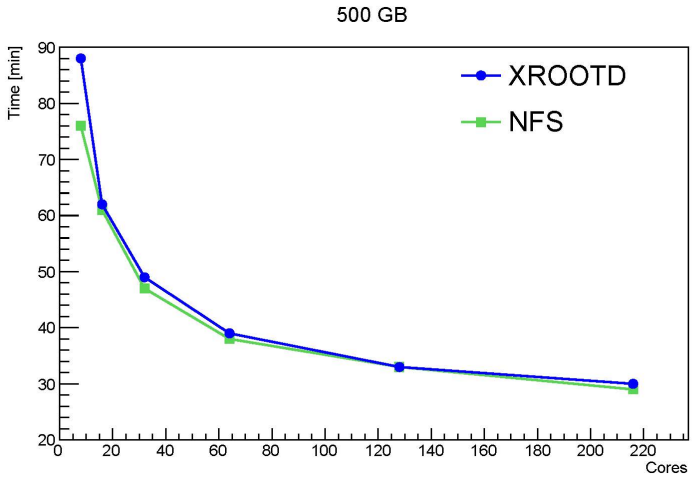

Fig. 4 – Execution time per protocol required for 500 GB data set analysis using PROOF on Demand.

4.2. LARGE SIZE DATA SET

Figure 3 shows the processing time using up to 8 cores. From the tested protocols, the fastest is NFS shortly followed by RFIO and XROOTD is the slowest.

We also ran tests using up to 216 nodes in the cluster for XROOTD and NFS with PoD (Fig. 4). This showed the same as we observed before that for a small number of cores the NFS give us a better time of processing, but after 60 cores the results are almost the same.

5. CONCLUSIONS

We have described the current status of a fully-featured PROOF-based analysis on the Bucharest ATLAS Analysis Facility. Although it is clear from our tests that there are some remaining issues with the use of the studied protocols, the best performance is given by NFS for a small number of processing cores. Otherwise the results are similar for all the three studied protocols and we suggest that is better to access files directly from the grid site RO-02-NIPNE because it has more storage capacity than the NFS server.

We demonstrated that the PROOF system, part of the widely adopted ROOT analysis environment, extends the amount and range of data that can be interactively analyzed. The PROOF system is becoming a full solution for LHC analysis, in particular for Tier-2/Tier-3 clusters and/or multi-core machines.

Acknowledgements. This work was presented at the 9th edition of the Workshop on Quantum Field Theory and Hamiltonian Systems, 24-28 September 2014, Sinaia, Romania.

REFERENCES

1. G. Aad *et al.* [ATLAS Collaboration], JINST **3**, S08003 (2008).
2. P. Laycock *et al.*, "*International Conference on Computing in High Energy and Nuclear Physics*" (Amsterdam, 2013).
3. `http://root.cern.ch/`.
4. M. Ballintijn *et al.*, Nucl. Instrum. Meth. A **559**, 13 (2006).
5. M. Cuciuc, M. Ciubăncan, V. Tudorache, A. Tudorache, R. Păun, G. Stoicea, C. Alexa, Rom. Rep. Phys. **65**, 122 (2013).
6. M. Ciubăncan, G. Stoicea, C. Alexa, A. Jinaru, J. Maurer, M. Rotaru, A. Tudorache, Rom. Rep. Phys. **67**, 386 (2015).
7. J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, "*Sourcebook of Parallel Computing*" (Morgan Kaufman, 2003).
8. `http://pod.gsi.de`.
9. `http://www.freebsd.org/doc/en/books/handbook/network-nfs.html`.
10. C. Boeheim, A. Hanushevsky, D. Leith, R. Melen, R. Mount, T. Pulliam, B. Weeks, "*Scalla: Scalable Cluster Architecture for Low Latency Access Using xrootd and olbd Servers*".
11. G. A. Stewart, G. A. Cowan, B. Dunne, A. Elwell, A. P. Millar, "*International Conference on Computing in High Energy and Nuclear Physics CHEP07*" (J. Phys.: Conf. Ser. 119 062047, 2008).
12. `http://twiki.cern.ch/twiki/bin/view/LCG/DpmAdminGuide`.

13. Lana Abadie *et al.*, "*DPM Status and Next Steps CHEP07*" (Victoria, 2007).
14. https://twiki.cern.ch/twiki/bin/view/AtlasComputing/RootCore?redirectedfrom=Atlas.RootCore.
15. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CONFNOTES/ATLAS-CONF-2014-006/.