Journal of Physics: Conference Series **664** (2015) 042019 doi:10.1088/1742-6596/664/4/042019

Tests of PROOF-on-Demand with ATLAS Prodsys2 and first experience with HTTP federation

R. Di Nardo[2], G. Ganis[1], E. Vilucchi[2], P. Albicocco[2], M. Antonelli[2] on behalf of the ATLAS collaboration

[1] CERN, CH - 1211 Geneva 23, Switzerland
[2] INFN Laboratori Nazionali di Frascati, via E. Fermi 40, 00044 Frascati (RM), Italy

E-mail: Gerardo.Ganis@cern.ch, Roberto.Di.Nardo@cern.ch, Elisabetta.Vilucchi@lnf.infn.it

Abstract. During the LHC Run-1, Grid resources in ATLAS have been managed by the PanDA and DQ2 systems. In order to meet the needs for the LHC Run-2, Prodsys2 and Rucio are used as the new ATLAS Workload and Data Management systems.
The data are stored under various formats in ROOT files and end-user physicists have the choice to use either the ATHENA framework or directly ROOT. Within the ROOT data analysis framework it is possible to perform analysis of huge sets of ROOT files in parallel with PROOF on clusters of computers (usually organised in analysis facilities) or multi-core machines. In addition, PROOF-on-Demand (PoD) can be used to enable PROOF on top of an existing resource management system.
In this work, we present the first performance obtained enabling PROOF-based analysis at CERN and in some of the Italian ATLAS Tier-2 sites within the new ATLAS workload system. Benchmark tests of data access with the httpd protocol, using also the httpd redirector, will be shown. We also present results on the startup latency tests using the new PROOF functionality of dynamic workers addition, which improves the performance of PoD using Grid resources.
These new results will be compared with the expected improvements discussed in a previous work.

1. Introduction

The LHC Run-2 has just started and the ATLAS [1] computing system Prodsys1 was updated to Prodsys2 [2], in order to meet the increased experiment requirements. In fact the new system must be able to scale with respect to the exponential growth of the task submission rate, must allow to automate some activities performed manually in the past, be flexible, dynamic as well as be simple to use for the end user.
New functionalities were introduced by adding to PanDA [3] two new components: DEFT and JEDI to manage tasks and jobs definition and to optimize the workload [4], while the data management system changed from DQ2 to Rucio [5]. The HTTPS/WebDav protocol being now fully in production in the storage systems, the need to group distributed Storage Elements (SEs) that expose them under a single name space has led to the deployment of the Dynamic Federation. The Dynamic Federation system allows to couple distributed storage endpoints with HTTP/WebDAV and S3 protocols and exposes a dynamic unique name space, built on the fly by merging metadata items taken on demand from the endpoints (both local and remote) [6].

In this work we present the first performance obtained enabling PROOF-based analysis, [7], within the new ATLAS workload system. Benchmark tests of data access with HTTPS/WebDav protocol, using also the Federation, will be shown. We also present results on the startup latency tests using the new PROOF functionality of dynamic workers addition, [8], which improves the performance of PROOF-on-Demand (PoD) [9] using Grid resources. These new results will be compared with the expected improvements discussed in a previous work [9].

2. Running PROOF-on-Demand in the ATLAS Prodsys2

The new components of Prodsys2 are DEFT and JEDI. DEFT, standing for *Database Engine For Tasks*, is responsible for the Meta-Tasks and tasks definition, manages inter-dependent groups of tasks (Meta- Tasks) and generates the corresponding data processing workflows. JEDI, the *Job Execution and Definition Interface*, is an intelligent component in the PanDA server that has the capability for task-level workload management. In order to make the PanDA system more task-oriented and to integrate prodDB and PanDADB [4], JEDI defines the jobs to submit to PanDA optimizing the workload: it dynamically translates the task definitions from DEFT into actual workload jobs executed in the PanDA Workload Management System.
In order to be able to run PROOF-based analysis with PoD using ATLAS PanDA queues as back-end, some options of JEDI must be disabled. The PoD PanDA plug-in uses the PanDA-client software `prun` to make a bulk submission in PanDA of the required number of workers in the PanDA analysis queue. Then the plug-in was modified by adding specific options to `prun`:

```
prun $SITE --maxFileSize=3072000 --noBuild --outputs="XYZ:pod-agent.client.log"
--disableAutoRetry --skipScout --allowTaskDuplication
--outDS user.$NICKNAME.pod --nJobs=$NUM_WRK --exec $JOBSCRIPT
```

In particular, the `--skipScout` option is used to disable the scout jobs, the automatic job retry is disabled by using `--disableAutoRetry` and the `--allowTaskDuplication` option allows multiple tasks to contribute to the same output dataset without re-activating already finished tasks and creating a new task for each PoD job submission. This last option is necessary because the output dataset is used by PoD to store the log files.

3. Access tests with XRootD and HTTP protocol

In a previous work [9], we tested the performance of input data access of a PROOF based analysis, using the XRootD protocol with different types of SRM (DPM [10], StoRM [11], EOS [12]), accessing data in LAN and WAN, also through FAX, the federation of ATLAS storage with XRootD. We proved the effectiveness of the use of the federation to access data, providing a globally reliable storage system with good performance and transparent access for the user.

In this work we test the input data access of a PROOF based analysis using the HTTPS/WebDav access available for DPM [13]. Moreover, we made some tests accessing input datasets through the Dynamic Federation with HTTP/WebDav, with the federator server located at DESY. The federator gives transparent access to storage through the HTTPS/WebDav protocol, presenting a unique name space and redirecting the requests to the closest available SE, so that HTTP and WebDAV clients can browse the Federation and directly download files using the Rucio syntax. Moreover, the federator server is always aware of the endpoints' status, that are checked every 10 seconds. If the httpd daemon is offline, then the storage system is removed from the Federation. When the HTTP protocol is available again on a removed SE, the federator tests it for few minutes before enabling it again in the Federation.

For our tests we run a PROOF-based analysis on a PROOF cluster setup with PoD/PanDA in the Frascati Tier-2. The input dataset is stored both in DPM SEs at Frascati and Naples ATLAS Tier-2. We ran two types of tests: in the first one we compared the performance accessing

the input dataset with XRootD, HTTPS/WebDav directly and HTTPS/WebDav through the federator server; in the second one we test the fallback mechanism of the Dynamic Federation mentioned above, using another server from the Federation when the original one goes down.

3.1. Comparing XROOT, HTTP and HTTP Federation access technologies
The first test was made running the PROOF-based analysis addressing the input files with:

A Using XRootD, via the following path:
`root://atlasse.lnf.infn.it//dpm/lnf.infn.it/home/atlas/atlaslocalgroupdisk/$path`

B Using HTTP/WebDav protocol, via the following path:
`https://atlasse.lnf.infn.it//dpm/lnf.infn.it/home/atlas/atlaslocalgroupdisk/$path`

C Through the HTTP federation test-bed with the federator located at DESY, with the path:
`http://federation.desy.de/fed/atlas/$path`

where `$path` is for example `rucio/mc12_8TeV/NTUP_HSG2.01371016._000016.root.1`. The result of the test, in Figure 1, shows that the overhead due to the federator is small and that XRootD protocol seems to perform better than HTTP/WebDav.

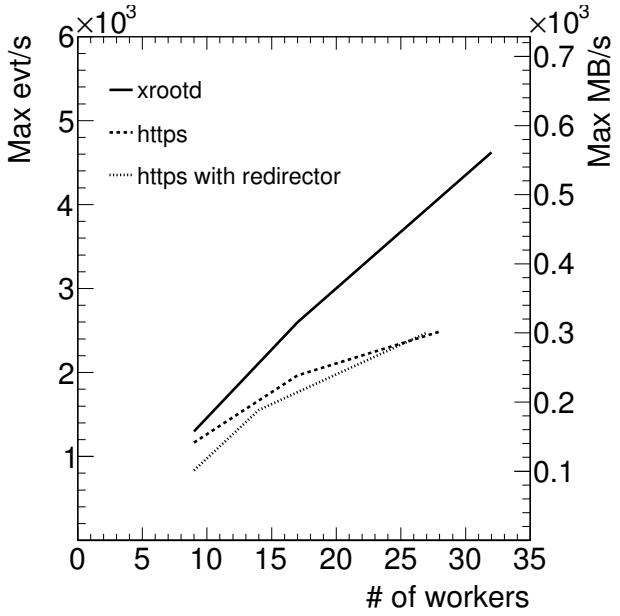
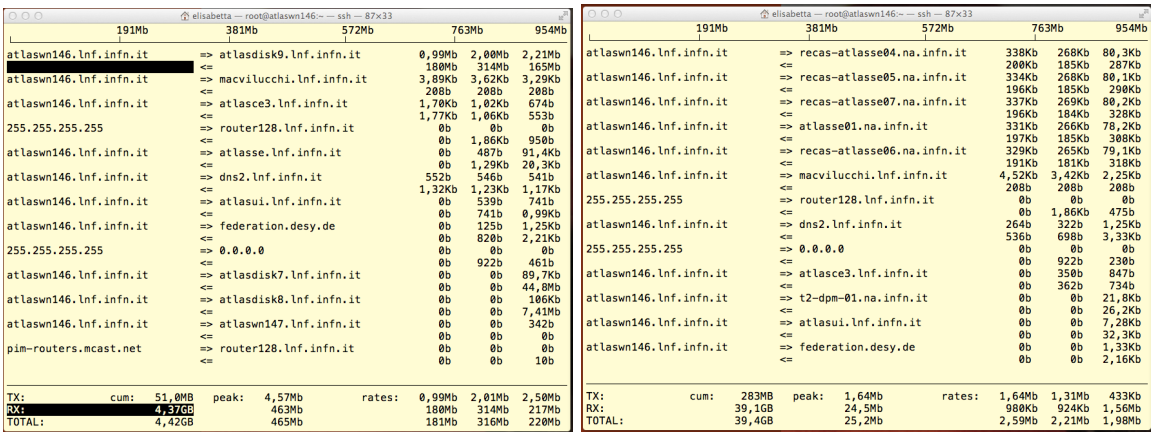


Figure 1: Access with XRootD and HTTP protocol

3.2. HTTP protocol and Dynamic Federation over LAN and WAN
The second test was performed with an analysis task running in Frascati and accessing, through the federator, a dataset available both in Frascati and Naples DPM SEs. During the tests we checked the network connections of the PROOF cluster nodes, with a focus on those towards the SEs and the federator. The first PROOF task was launched while both SEs were up and the federator effectively redirected the workers to the Frascati SE, as expected. In Figure 2(a) we report the screenshot with network connections of one of the worker nodes: we can notice the connections to the federator (*federation.desy.de*), to the Frascati DPM

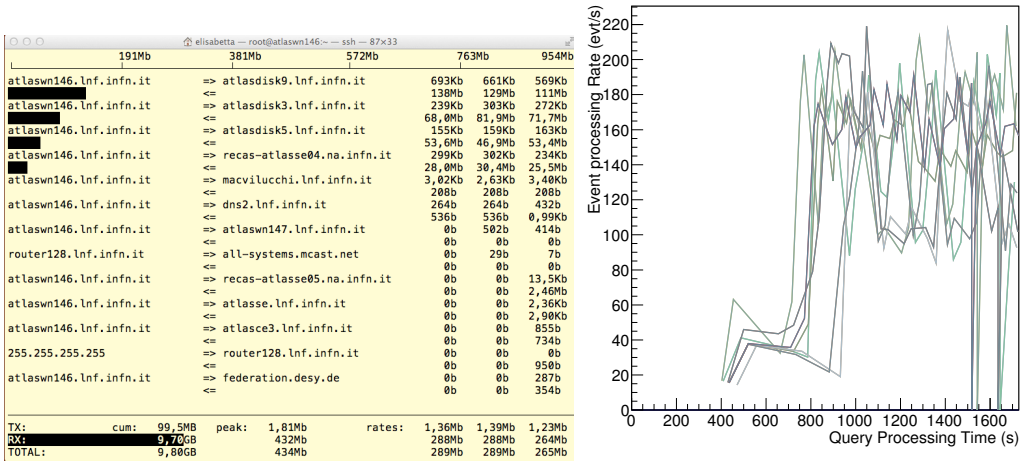
head node (*atlasse.lnf.infn.it*) and to some DPM pool nodes in Frascati (*atlasdisk9.lnf.infn.it, atlasdisk8.lnf.infn.it, ...*).
To test the federator responsiveness, the task was launched a second time after bringing down the Frascati SE; the federator then redirected the workers to the Naples SE as we can see in the screenshot in Figure 2(b), where *t2-dpm-01.na.infn.it* is the DPM head node in Naples and *recas-atlasse04.na.infn.it, atlasse01.na.infn.it, ...* are some of the DPM pool nodes in Naples. After two minutes of task execution, Frascati SE was enabled again, in about 5 minutes the



(a) (b)

Figure 2: (a) PROOF node network connections to Frascati SE and Federator. (b) PROOF node connections transition from Napoli SE to Frascati SE.

workers stopped accessing data from Naples and resumed local access. The transition phase is shown in the screenshot in Figure 3(a). The transition, with the related boost of performance in terms of the rate of processed events, is visible in the plot in Figure 3(b), where a delay can be observed as expected from the Dynamic Federation behavior.



(a) (b)

Figure 3: (a) PROOF node connections transition from Napoli SE to Frascati SE. (b) Boost of performance with HTTP from WAN to LAN.

4. PROOF-on-Demand and Prodsys2

As explained in [9], the worker availability curve as a function of the time from submission can be used to measure the *startup latency* and *ramp-up slope*, two parameters affecting the *time-to-result* for an analysis.

The startup latency time depends on the status of the site, the priority of the user and the number of job slots requested while the ramp-up slope is more affected by the size of the site. Measurements similar to those presented in [9] have been performed, where the average of single tests consisting in a user requiring 100 workers and retrieving the startup time of each worker have been used. Figure 4 shows the average latency curve obtained for the measurements performed at CERN (red full line), as an example of a large site, and at Naples Tier-2 (blue full line), an example of a smaller site. The hashed area represents the RMS of the various single tests for both cases, reflecting the impact of the different load condition of the sites during the tests. In addition, the dotted lines shows the results of the previous measurements performed in 2013 for the same two sites and reported in [9]. With respect to the previous measurements

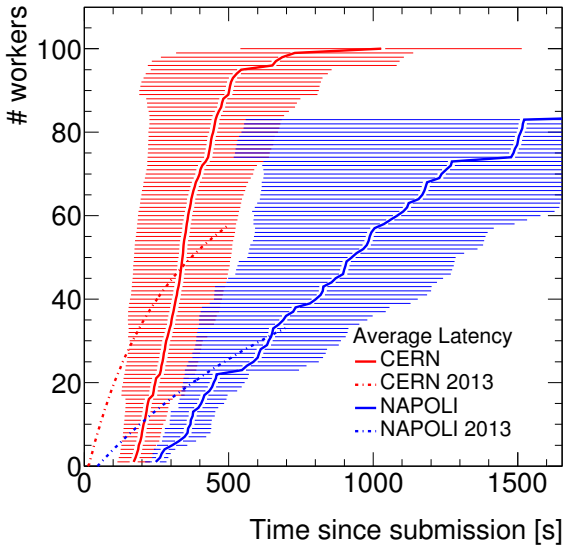


Figure 4: Average latency at Naples Tier-2 and CERN: a comparison with 2013 measurements.

done in 2013, it has been found that:

- the startup latency has significantly increased by about 100 seconds and this can be explained by the additional layers introduced in ProdSys2;
- the ramp-up slope has also increased, indicating an increase in the amount of resources available in the sites considered for this test.

5. Latency tests with dynamic worker addition

At the previous edition of the conference, an ongoing development in the PROOF system, introducing support for dynamic worker addition during a running query, was presented and discussed [8]. The development addressed the case where workers are provided by a resource management system managing scarce resources, therefore subject to inherent delays or latencies in worker availability. In [8] it was shown that in such cases, the PROOF pull model with dynamic worker addition, for short or medium length jobs can be significantly more efficient in using the resources than the push model implemented in the classical batch submission. Using

measured ramp-up and latency values from the CERN Tier from [9], the theoretical speed-up for a job requiring 10 days of serialized computing time, could be up to 30% [8]. The effect is expected to be larger for smaller sites where latencies are larger and ramp-up rate is smaller.

In the meantime the PROOF development has been brought to a state such that the prediction can be checked in practice [1] . We used a relatively small site, Naples Tier-2, and ran analysis jobs reading a data set stored locally. The analysis task required about 150 minutes of serialized time. The model in [8] predicts a processing time of 700s in pull mode and a ratio *push-vs-pull* time-to-result ratio of 0.73 or a *pull-vs-push* average speed-up of 27%.

The new functionality allows to launch the query as soon as the workers are requested to PanDa (via `pod-submit`). The PROOF job will pause waiting for the first worker to come. When the first worker comes the analysis starts. As soon as other workers come they ask for work and contribute to the analysis processing phase.

We explain this quantitatively in Figure 5. We have first measured the average latency and worker ramp-up rate for the site by running several job submissions via PoD/PanDa. In Figure 5 this is shown by the area hashed in blue. We then asked for 40 workers and launched a PROOF job with dynamic worker addition enabled. The black curve shows the number of active workers during the query time in the case of dynamic addition. We can see the ramp-up of available workers, which came in bunches, with the first worker processing for a much longer time than the last one. The dashed black line shows what worker activity would look like in push-mode, assuming the same worker availability curve and the same amount of work for each worker. The completion time of the last starting sub-job determines the time-to-result. In the example chosen the *pull-vs-push* speed-up in time-to-result is 20% . This is in good agreement with the prediction, considering that the observed ramp-up rate is on the upper side of the distribution observed for the site.

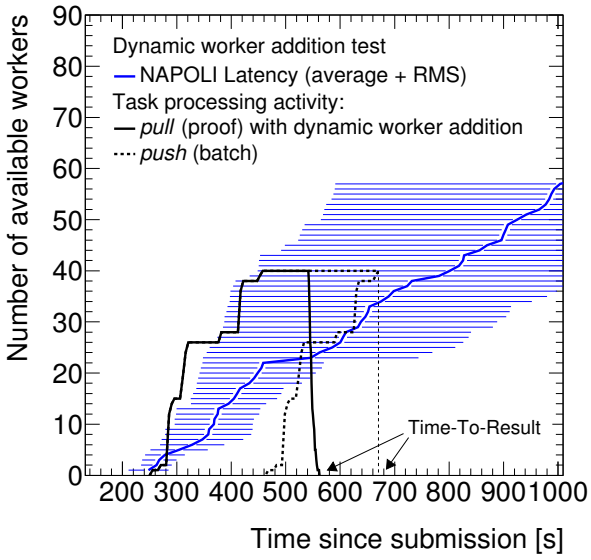


Figure 5: Latency measured with and without dynamic workers addition.

6. Conclusions

Initial tests of PoD PanDA with ProdSys2 are positive, though an increase in startup latencies is observed. Tests of data access via the HTTPS protocol seem to indicate that XRootD delivers

[1] Full functionality is available in ROOT 5.34/28 and in the forthcoming ROOT 6.04.

better overall performance. First functional tests of the HTTP federation are positive; more intensive tests are however required for an evaluation of the full functionality.

6.1. Acknowledgments
The authors would like to thank F. Furano for his advise and suggestions on the HTTP federation. This work was developed in the framework of the PRIN Project "STOA-LHC 20108T4XTM", CUP: I11J12000080001, and partly supported by it.

References

[1] ATLAS Collaboration 2008 *The ATLAS Experiment at the CERN Large Hadron Collider* JINST 3 S08003
[2] Campana S and Di Girolamo A 2015 *ATLAS Distributed Computing in LHC Run2* talk at CHEP2015
[3] PanDA: `https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA`
[4] De K *et al* on behalf of the ATLAS Collaboration 2014 *Task Management in the New ATLAS Production System* J. Phys.: Conf. Ser. **513** 032078
[5] Rucio: `http://rucio.cern.ch`
[6] Furano F 2015 *Dynamic Federation* white paper `http://svnweb.cern.ch/world/wsvn/lcgdm/ugr/trunk/doc/whitepaper/Doc_DynaFeds.pdf`
[7] PROOF: `https://root.cern.ch/drupal/content/proof`
[8] Berzano D *et al* 2014 *PROOF as a Service on the Cloud: a Virtual Analysis Facility based on the CernVM ecosystem* J.Phys.Conf.Ser. **513** 032007
[9] Vilucchi E *et al* 2014 *PROOF-based analysis on the ATLAS Grid facilities: first experience with the PoD/PanDa plugin* J.Phys.Conf.Ser. **513** 032102 .
[10] DPM: `https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm`
[11] StoRM: `http://storm.forge.cnaf.infn.it/documentation/`
[12] Peters A and Duellemann D 2015 *EOS as the present and future solution for data storage at CERN* talk at CHEP2015
[13] Ayllon A *et al* 2012 *Web enabled data management with DPM & LFC* J. Phys.: Conf. Ser. **396** 052006