Enabling data analysis à la PROOF on the Italian ATLAS Tier-2s using PoD

Roberto Di Nardo[1], Gerardo Ganis[2], Elisabetta Vilucchi[1], Alberto Annovi[1], Mario Antonelli[1], Gianpaolo Carlino[3], Alessandro De Salvo[4], Alessandra Doria[3], Anar Manafov[5], Agnese Martini[1], Marianna Testa[1] on behalf of ATLAS Collaboration

[1] INFN Laboratori Nazionali di Frascati, via Enrico Fermi 40, IT-00044 Frascati, Italy
[2] CERN, CH-1211 Geneva 23, Switzerland
[3] INFN Napoli and Universita' di Napoli, Dipartimento di Scienze Fisiche, Complesso Universitario di Monte Sant'Angelo, via Cinthia, IT-80126 Napoli, Italy
[4] INFN Roma-1 and Universita' La Sapienza, Dipartimento di Fisica, Piazzale A. Moro IT-00146, Roma, Italy
[5] GSI Helmholtzzentrum für Schwerionenforschung GmbH, Planckstr. 1, 64291 Darmstadt, Germany

E-mail: roberto.dinardo@lnf.infn.it, gerardo.ganis@cern.ch, elisabetta.vilucchi@lnf.infn.it

Abstract. We describe our experience using PROOF for data analysis on the Italian ATLAS-Tier2 in Frascati, Napoli and Roma1. To enable PROOF on the cluster we used PoD, Proof-on-Demand. PoD is a set of tools designed to interact with any resource management system (RMS) to start the PROOF daemons. In this way any user can quickly setup its own PROOF cluster on the resources, with the RMS taking care of scheduling, priorities and accounting. Usage of PoD has steadily increased in the last years, and the product has now reached a production level quality. PoD features an abstract interface to RMSs and provides several plugins for the most common RMSs. In our tests we used both the gLite and PBS plug-ins, the latter being the native RMS handling the resources under test. Data were accessed via xrootd with file discovery provided by the standard ATLAS tools. The SRM is DPM (Disk Pool Manager) which has rfio as standard data access protocol; so we provided DPM of Xrootd protocol too. We describe the configuration and setup details and the results of some benchmark tests we run on the facility.

ATL-SOFT-PROC-2012-033
18 May 2012

1. Introduction

In the ATLAS computing model [1], Tier-2 resources are intended for MC productions and end-user analyses activities. These resources are usually exploited via the standard GRID resource management tools which are de facto a high level interface to the underlying batch systems managing the contributing clusters. While this is working as expected, there are user-cases where a more dynamic usage of the resources may be more appropriate. For example, the design and optimization of an analysis on a large data sample available on the local storage of the Tier-2 requires many iterations and

fast turn around. In these cases a 'pull' model for work distribution, like the one implemented by PROOF [2], may be more effective. In this paper we describe our experience using PROOF for data analysis on the Italian ATLAS Tier-2s. We used Proof-On-Demand, PoD [3], to enable PROOF on the resources with both the gLite and PBS back-ends, the latter being the native RMS handling some of the resources under test. Data management was provided by DPM [4] and data files were accessed by means of the XROOTD client through the DPM/XROOTD door.
This paper is organized as follows. In Section 2 the essential parts of the ATLAS computing model are recalled, in particular the organization of the resources. In Section 3 the technique to enable PROOF on Tier-2 systems is described. The first results about start-up latency and read-out rates are shown and discussed in Section 4. Finally in Section 5 we outline the future directions.

2. The ATLAS Computing Model

The computing models of the LHC experiments have been designed many years ago following the MONARC paradigm, based on a hierarchical structure of the computing centres organized in different levels or Tiers. The Tier-0 facility based at CERN is responsible of the first-pass processing and archiving of the primary raw data and their distribution to the Tier-1 centres, world-wide distributed, which have to store and guarantee a long-term access to raw and derived data and to provide all the reprocessing activities. Each Tier-1 is connected to a set of Tier-2 and Tier-3 sites grouped in regional Clouds. The Tier-2s are medium size computing centres designed for the user analysis and provide all the Monte Carlo simulation capability while the Tier-3s, small centres located in each university, are designed for the final steps of data analysis.
From the point of view of the network, in this hierarchical model the Tier-2s are connected and exchange data only with the Tier-1 in their cloud, thus very fast links are not needed. Only the Tier-1 sites need to be connected among them and with the Tier-0 with high-speed connections.
With the start-up of the LHC operations, the large amount of collected data and of copies of the same data replicated in all the clouds, showed that such a static model would have needed an increasing amount of storage resources in order to scale adequately.
On the other hand, the evolution of the network technology, made it possible to design a new data distribution model based on the network connectivity with a more efficient exploitation of the mass storage. The easy and fast data access allows to replicate only few master copies in the Tier-1s while the distribution of physics data in the Tier-2 centres can be driven by the real user needs, avoiding the over-replication of data and allowing a dynamic data caching and a continuous refresh of data at the Tier-2s.
ATLAS has identified the Tier-2 centres showing a high level of reliability, continuity of services and good network connectivity and such sites, called Direct Tier-2s (T2Ds), have been included in a mesh network structure composed of all the directly connected sites and all the Tier-1s. After a preliminary phase of study and experimentation, 14 pilot sites have been identified in ATLAS, with three Italian Tier-2 among them, in order to deploy and test this new generation network.
Analysis of data by users and physics groups is made through the submission of jobs on the Grid to the two available backend: Panda, the Production ANd Distributed Analysis system for ATLAS [5], and the Workload Management System (WMS). While local clusters are reserved to the last step of the analysis and software development.
The distributed analysis tools available for job submission are: prun and Pathena for the Panda back-end and Ganga [6] for both Panda and WMS back-end. Presently ATLAS does not encourage the use of WMS. In fact, only tasks submitted via Panda are reported in the official accounting systems [7]; although local accounting system (HLR with DGAS2Apel for the Italian cloud) records any job run in the Tiers. The tool used for this work - PoD with the gLite plug-in – submits the PROOF daemon jobs directly to the WMS back-end.

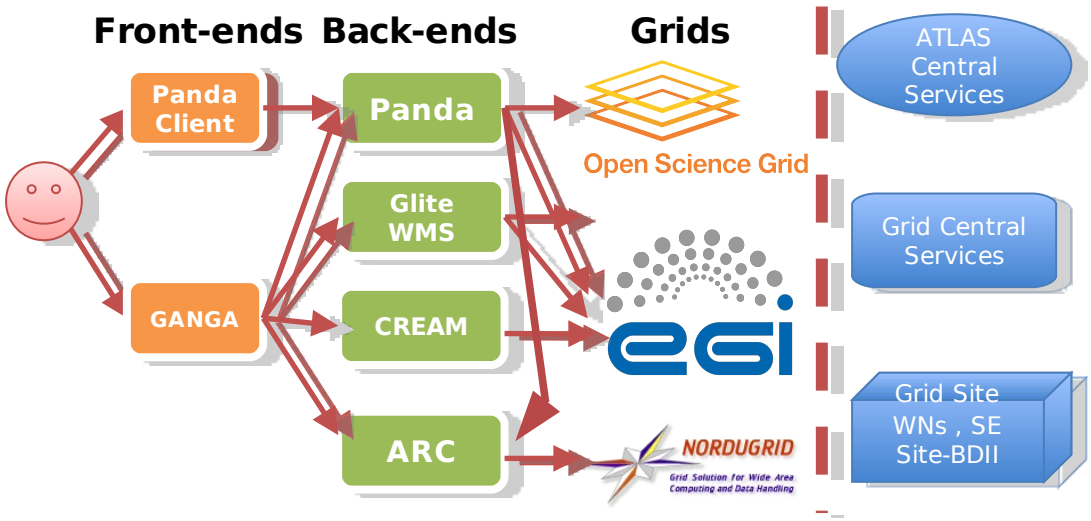

Figure 1. Analysis tools

2.1 The ATLAS Italian Cloud and Frascati testbed

The ATLAS Italian cloud is made of the Tier-1 at CNAF (Bologna) and four Tier-2s: Frascati, Milano, Roma1 and Napoli [8]. Local Resource manager systems are Portable Batch System (PBS), Condor and Load Sharing Facility (LSF). To optimize the use of the computing resources, a mechanism of job priorities and of resource sharing among the different activities inside the ATLAS Virtual Organization (VO) was implemented. The mechanism makes use of the VOViews publication in the Information System (IS) and the fair share implementation per UNIX group in the batch system [9]. In this way the WMS is able to correctly manage published VOViews and resources allocated to a generic ATLAS users, users with production role and users of the /atlas/it group correspond to the defined share. For jobs submitted to the Panda backend, the priority is managed by the system itself.

The Storage Resource Management (SRM) systems in use at Italian Tier-2 sites are the Disk Pool Manager (DPM) and StoRM. DPM a lightweight solution for disk storage management that offers the required SRM interfaces and allows the space reservation for different activities of the experiment (*space tokens)*. StoRM is a GRID SRM for disk-based storage systems developed at INFN-CNAF and designed to support guaranteed space reservation and direct access (native POSIX I/O call) to the storage [10]. It takes advantage from high performance parallel file systems like GPFS [11] and is available in Milano. The most relevant figures of the Italian Tier-2s are summarized in Table 1.

IT Tier-2s 2012 total resources	Computing			Storage	
	Job slots	HePSpec	Batch System	Capacity (TB)	Srm type
Frascati	870	8300	PBS	420	DPM
Milano (T2D)	1050	10900	PBS/Condor	1104	StoRM
Napoli (T2D)	1200	12400	PBS	1104	DPM
Roma1 (T2D)	1300	13100	LSF	1044	DPM

Table 1. Italian cloud resources

The testbed for the system development was configured at Frascati Tier-2. Frascati, being the most recent Tier-2, is the smallest one in Italy. The middleware presently used is gLite version 3.2, the LRMS is PBS and the scheduler is Maui with a fair-share policy implemented on the base of system groups that correspond to the VOMS groups and role. The SRM is DPM.
The Italian Tier-2s hosting the DPM Storage Element have been instrumented with the XROOTD access. Currently, given the implementation of XROOTD with DPM, the access is read-only and insecure, meaning that there is no authentication/authorization layer activated. In fact, the security infrastructure is currently working for the ALICE experiment only. However, while it is still not possible to open the XROOTD access in Wide Area Network, the read-only access in LAN is working correctly, so it has been enabled for the nodes of the clusters for local access.
The implementation of XROOTD in DPM will be enhanced in the future, both from the point of view of the authentication/authorization layer and for what concerns the performance. ATLAS software is available to worker nodes and user interfaces through the CernVM File System. The CernVM File System (CernVM-FS) [12] is a file system used by various HEP experiments for the access and on-demand delivery of software stacks for data analysis, reconstruction, and simulation; it is a fuse-based http, read-only file system which guarantees file de-duplication, on-demand file transfer with caching, scalability and performance. It consists of web servers and web caches for data distribution to the CernVM-FS clients that provide a POSIX compliant read-only file system on the worker nodes.

2.2 Analysis data formats

The data formats used for analysis are AOD and D3PD. The latter is a derivation of the former in the form of a flat ROOT TTree with several branches organized by name according to the reconstructed physical quantities they represent. Being a standard ROOT tree, the D3PD format is particularly adapted to be analyzed with PROOF. In fact, several high level analysis tool working on D3PDs, for example SFrame [13], use in the background interfaces with PROOF.

3. Enabling PROOF on Tier-2s with PoD

The goal of the PROOF system is to enable interactive analysis on a set of distributed resources using a multi-tier master-worker model to achieve dynamic workload-balancing. PROOF was initially addressing the case of a dedicated cluster of resources. However, since the beginning, it was clear that in many cases analysis groups would not have been in the position to afford dedicated cluster. The advocated solution was to make PROOF coexist with a standard resource management system. Several attempts in this direction were done, for example using Condor, LSF and SGE; an interface with the Condor system was even distributed with ROOT and used in the PHOBOS experiment. PoD, Proof-On-Demand, is the most complete result of these activities. PoD is a tool-kit defining the essential common interface required to setup a PROOF cluster on any Resource Management System (RMS); the different back-ends are then accessed using plug-in technology. The currently supported back-ends, i.e. the RMS for which a PoD plug-in has been implemented, are LSF, PBS, OGE, Condor, LoadLeveler and gLite-WMS. As we have seen in Section 2, the ATLAS end-user interacts with Tier-2 resources via the Panda, WMS or CREAM back-end; the PoD-gLite plug-in uses WMS submission. The development of PoD/Panda plug-in is under evaluation.

The basic idea of the exercise described in the paper is to use gLite-PoD to startup the PROOF daemons on the assigned resources and then to start PROOF sessions from the user work-station using standard network connections. In this model the PROOF master can be located on any node enabled to interact with the WMS; these are typically the User-Interface machines (UIs). The PROOF workers will be the machines assigned by the WMS, while the client machine is typically the end-user laptop/desktop or even the UI itself.

3.1 PoD and ROOT in the ATLAS distribution software

For our tests we used PoD and ROOT from the CVMFS ATLAS distribution. For PoD we used version 3.10, the latest available on CVMFS at the time of writing; this version contains some essential fixes for the gLite plug-in. We used ROOT 5.32/02 from CVMFS, the latest available at the time of writing; this version contains some important fixes for PROOF, mostly related to the fact that the user *username* on the assigned Tier-2 machines is different from the one used to submit the job.

3.2 Example of PoD at work

It is not the purpose of this paper to describe the way to operate PoD, but we think that showing an example of the basics steps can convey better the idea about how PoD works. PoD provides a simple and intuitive command line in order to simplify access to its functionality. There are basically two steps, starting the PoD server, i.e. the master, and submitting the worker jobs. These operations need to be done on the master node either from the master node itself or remotely from the user workstation. In the following we assume that we are operating PoD form the master node, i.e. the UI in the gLite case.

The PoD server is independent of the chosen back-end. The server is controlled from the command line using the *pod-server* command; to start the server just use the option *start*:

```
$ pod-server start
Starting PoD server...
updating xproofd configuration file...
starting xproofd...
starting PoD agent...
preparing PoD worker package...
select user defined environment script to be added to worker package...
selecting pre-compiled bins to be added to worker package...
PoD worker package will be repacked because "/atlashome/evilucch/.PoD/etc/xpd.cf"
was updated
PoD worker package: /atlashome/evilucch/.PoD/wrk/pod-worker
------------------------
XPROOFD [27630] port: 21001
PoD agent [27653] port: 22001
PROOF connection string: evilucch@atlas-ui-02.roma1.infn.it:21001
------------------------
```

Information about the server configuration is displayed on the screen, including the connection string; the latter is the URL to be used to start the PROOF cluster, either directly or via a proper SSH-tunnel. The server can be stopped issuing *pod-server stop*.

The next step is to start the workers nodes. This is done using the command *pod-submit*, which submits the jobs to the RMS to start the PROOF daemons. Job submission is obviously back-end-aware. To submit workers to a Tier-2, using the gLite-WMS, the CREAM computing element and the queue are required. For example, for the Frascati Tier-2 used in these tests we used

```
$ pod-submit -r glite -q atlasce2.lnf.infn.it:8443/cream-pbs-atlas_short -n 100
```

The number of workers available at any time is obtained via the useful command *pod-info*:

```
$ pod-info -n
45
```

As soon as some workers are available a PROOF session can started. Additional workers can be

picked up re-opening the PROOF session; otherwise they will not be used and the batch system will release them after a defined interval of time defined in the PoD code.

4. First results

In order to show how the systems works in a real environment we ran two kind of tests. The first test aimed at investigating the startup latency, i.e. the time need to get at least some of the required resources ready to start a PROOF session. In the second test we show what kind of readout performance can be achieved for a test analysis from the Tier-2 storage elements and its scalability versus the number of workers.

4.1 Startup latency

After first tests in Frascati's Tier-2, PoD was tested also in the other two sites with DPM: Roma1 and Napoli, giving comparable results.

The tests were made to highlight the time necessary to allocate a certain number of nodes with PoD before running PROOF analysis. This time has been conservatively taken as “startup latency”. As one might expect, this time depends on the number of nodes required and on the share allocated for the VOMS group with which the user is authenticated on the Grid. Additionally, it depends on the total number of job slots available in the Tier-2 and the average job runtime.

For an average of 10000 successfully run jobs per day (typical Frascati's Tier-2 values) one could expect 0.1 job slot available per second. If a 100% fair-share is dedicated to the PoD, one expect to allocate ~50 worker node in about 500 s. This is the same situation expected with a lower fair-share but with no other jobs pending.

The normal activity of the Tier-2 consists of: 30% of resources dedicated to the Monte Carlo production, 60% for analysis with Panda, and 10% for jobs submitted via WMS. For these tests we used a modified configuration in order ensure a 25% of the available resource fair-share to PoD submissions (via WMS), decreasing the Panda fair-share percentage.

Figure 2 shows the results of those tests performed at the Frascati Tier-2 using VOMS credential of Italian group of ATLAS Virtual Organization (VO). A job with 50 job slots request has been submitted with PoD every about 30 minutes for a total of 21 submission. The color scale is proportional to the job submission time from dark blue to light blue. As seen in Figure 2, jobs have experienced very different batch system loads, and in average about 1000 seconds are required to allocate all job slots requested with a large spread. This average startup latency is slightly better than that expected with 25% fair-share, indicating that many others parameters are involved. The system load is shown in Figure 3 where Ganglia [14] plot is reported with running and queued jobs (deep colors for running jobs and lighter colors for queued jobs) for the following VOMS role/groups: green for production role, yellow for Panda analysis with generic ATLAS role and olive green for ATLAS Italian group.

First submissions showed a large startup latency suffering from resources competitions from Panda analysis jobs (filled yellow histogram of Figure 3). As soon as Panda analysis job requests have been fulfilled, the startup latency is decreased to lower values. It can also be seen a slight increase of startup latency for late submission as expected from the 25% priority asymptotic value.

Additional tests have been performed to study the performances among users with the identical VOMS credential in competition for the same resources.

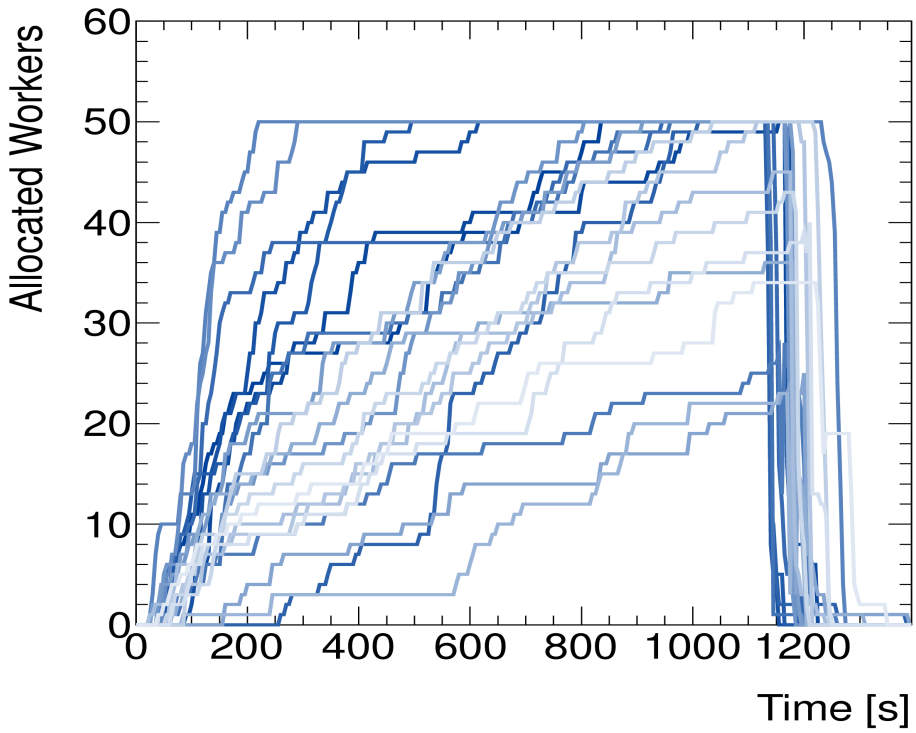

Figure 2. Results of the submission tests on the Frascati Tier-2: number of allocated slots as function of time for each bulk submission

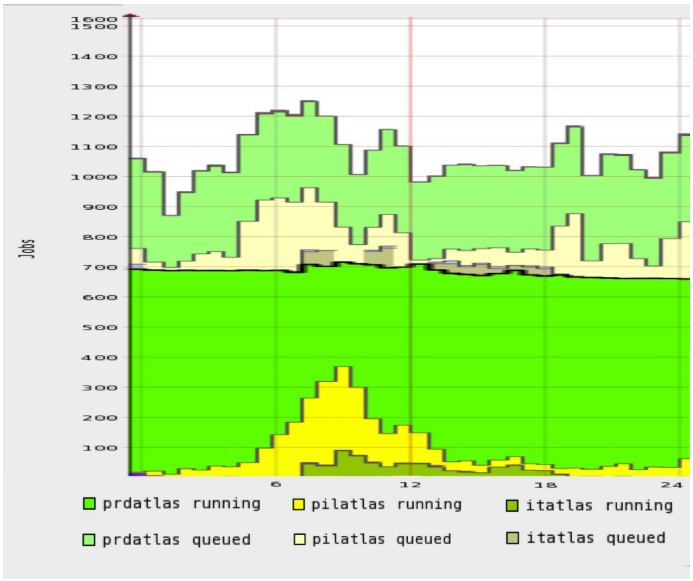

Figure 3. Ganglia plot of Frascati batch system load during the test of Figure 2.

Figure 4 shows three examples, corresponding to low, medium and high Tier-2 cluster load, of submissions by two users with proxy as ATLAS Italian group (then using the same resource share).
For a low farm load case the two users were able to allocate the requested number of nodes in the same time, without appreciable competition. As soon as the farm load increase, the users enter in

competition for the available resources, bringing to a larger startup latency. In the worst case one of the users is not even able to allocate the full requested nodes in the monitoring time window (20 minutes).

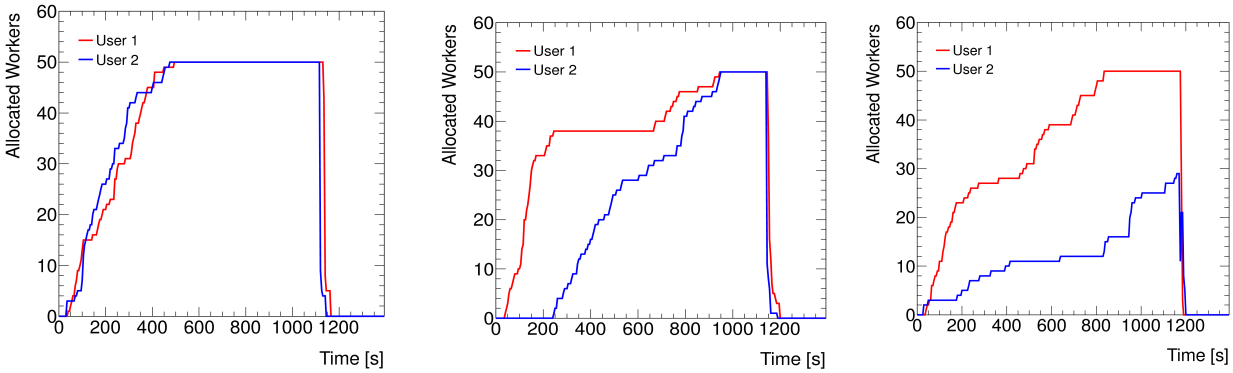

Figure 4. Number of allocated slots as function of time for two users with the same VOMS credential. The jobs for the two users have been submitted at the same time

5.2 Read-out performance

Since data analysis jobs are typically I/O bound, it is important to understand how the storage system of a given facility compares with the available number of CPU slots, i.e. of potentially concurrent processing jobs.

To investigate the data access rate we used a simple ROOT TSelector derived from standard D3PD and configured to read branches associated to tracks, electrons, muons and jets, corresponding to about 40% of the event. We measured the input rate in MBytes/second using the PROOF statistics tools as a function of the number of workers. This quantity is derived from the number of bytes effectively read out from the files by the active workers divided by the total processing time; the latter includes event decompression and construction of the event information in memory, which is the only CPU load in this simple analysis. Studying the input rate as a function of the number of workers allows to understand what are the single job requirements in terms of I/O. Since any real analysis will have a larger CPU load, the results obtained in this way are conservative.

The results obtained at the Roma1, Frascati and Napoli Tier-2's are shown in Figure 5 for three typical configurations:

Case 1: Worker processes distributed over many node, dataset files distributed over many file servers.
Case 2: Worker processes distributed over many nodes, dataset files distributed over few file servers.
Case 3: Worker processes on few nodes, dataset files distributed over many file servers.

In all the three Tier2's, the network topology is such that worker nodes and data servers are connected through 1 Gbit/s and 10 Gbit/s switches, respectively.

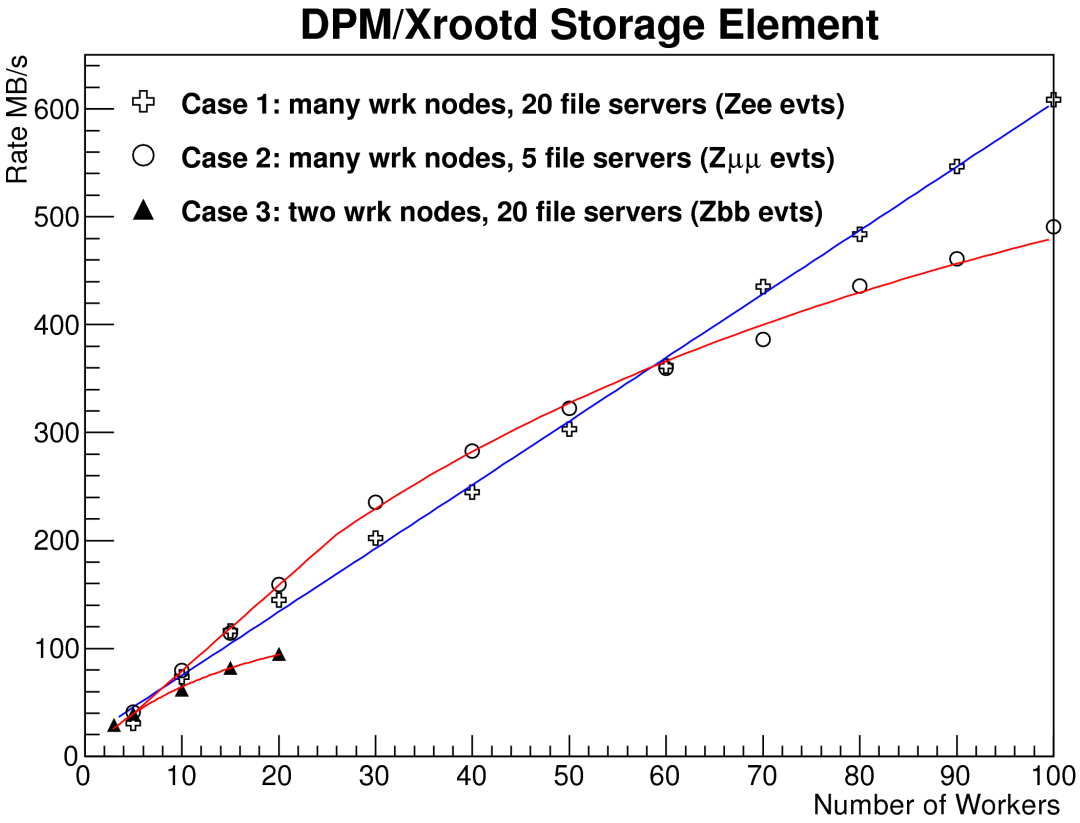

Figure 5. Results of the readout rate tests at Roma1, Frascati and Napoli Tier-2's. See text for interpretation details.

The super-imposed curves are the fits to the simple model presented in [15]:

$$Rate(N_{wrk}) = R_1 \cdot N_{wrk} \cdot \left[1 + \frac{R_1}{R_{I/O}} \cdot \left(\frac{N_{wrk}}{min(N_{I/O} \cdot N_{wrk})} - 1\right)\right]^{-1}$$

where N_{wrk} is the number of workers, R_1 is the single-process rate, $R_{I/O} \cdot N_{I/O}$ is the total I/O rate and $R_{I/O}$ is the average rate per I/O device. The results of the fits are shown in Table 2.

Parameter	Case 1	Case 2	Case 3
R_1	5.9 MB/s	7.9 MB/s	8.7 MB/s
$R_{I/O} \cdot N_{I/O}$	-	~900 MB/s	~175 MB/s

Table 2. Results of fits shown in Figure 5

The parameter R_1 measures the rate of reading and decompressing the event per worker. The measured values, here around 7-8 MBytes/s per process, depend on the type of analysis and on the structure of the event read and built in memory. The scalability for increasing number of workers

indicates how the system would react to increasing worker loads, i.e. to an increasing number of users.
In *Case 1* the scalability is good over the range tested. This is in agreement with the expectations because, with both storage and processing elements widely distributed over the resources, the effective network bandwidth is large when compared to the requirements of the number of processes under test.
The results for *Case 2*, on the contrary, shows some deviations from linear scalability, hint of the saturation phenomena described in [15]. In this case the dataset was distributed over 5 data servers, 3 out of which, at the time of the test, were temporary connected via a 1 Gbit/s network switch. The saturation value found by the fit, around 900 Mbytes/s, is in qualitative agreement with what expected by the network configuration.
For *Case 3* saturation starts at lower number of workers wrt *Case 2* because of the 1 Gbit/s network connection of workers; these configurations were obtained with a PoD fair-share of 5% giving a maximum of 25-30 workers located on two physically different nodes. Again, the saturation value found by the fit, around 175 Mbytes/s, is in qualitative agreement with what expected by the network configuration.
This result underlines the importance of a fully functional network set up for efficient data-serving to multiple processes. For the optimal configuration (*Case 1*), a back-of-the-envelop calculation shows that a SE configuration with 20 servers, like the ones available in Roma1 and Napoli, should be able to serve efficiently up to 3200 processes requiring each ~8 MBytes/s. Under these assumptions the storage system should be therefore adequate to the CPU processing power of those Tier-2s (see Table 1).

6. Future work

As mentioned in Section 3, the ATLAS community would benefit from the availability of a Panda-based PoD plug-in. Feasibility studies have been started with the aim of having a working prototype in reasonably short times.
The plan is also to continue performance measurements using real analysis and multi-user configurations.

References

[1] R.W.L. Jones and D. Barberis, *"The Evolution of the ATLAS Computing Model"*, J. Phys.: Conf. Ser. 219 072037.
[2] http://root.cern.ch/drupal/content/proof
[3] http://pod.gsi.de
[4] https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm
[5] T Maeno, *"PanDA: distributed production and distributed analysis system for ATLAS"*, 2008 *J. Phys.: Conf. Ser.* **119** 062036.
[6] J T Moscicki *et al.*, *"Ganga: a tool for computational-task management and easy access to Grid resources, Computer Physiscs Communications"*, vol. 180, Issue 11 (2009), arXiv:0902.2685.
[7] http://dashboard.cern.ch/atlas/
[8] L Rinaldi *et al.*, *"ATLAS computing activities and developments in the Italian Grid cloud "*, Proc. Conf. 183 for Computing in High-Energy and Nuclear Physics (CHEP 2012) (New York, USA)
[9] A Doria et al, "Deployment of job priority mechanisms in the Italian cloud of the ATLAS experiment", J. Phys. Conf. Ser. (2010) 219, 072001.
[10] http://storm.forge.cnaf.infn.it
[11] http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=com.ibm.cluster.gpfs.doc/gpfsbooks.html

[12] A De Salvo *et al.*, *"Software installation and condition data distribution via CernVM FileSystem in ATLAS"*, Proc. Conf. 183 for Computing in High-Energy and Nuclear Physics (CHEP 2012) (New York, USA).
[13] D.Berge, J.Haller, A.Krasznahorkay, “SFrame: A high-performance ROOT-based framework for HEP data analysis”, PoS ACAT2010:048,2010.
[14] http://ganglia.sourceforge.net/
[15] C.Aguado-Sanchez *et al*, *“Studying ROOT I/O performance with PROOF-Lite”*, 2011J. Phys.: Conf. Ser. **331** 032010, http://iopscience.iop.org/1742-6596/331/3/032010.