# A prototype of a Virtual Analysis Facility: first experiences

**S Bagnasco**[1]**, D Berzano**[2]**, S Lusso**[1] **and M Masera**[2]

[1] Istituto Nazionale di Fisica Nucleare, Via Pietro Giuria 1, 10125 Torino, Italy
[2] Istituto Nazionale di Fisica Nucleare and Dipartimento di Fisica Sperimentale, University of Torino, Via Pietro Giuria 1, 10125 Torino, Italy

E-mail: `stefano.bagnasco@to.infn.it`

**Abstract.** Current Grid deployments for LHC computing (namely the WLCG infrastructure) do not allow efficient parallel interactive processing of data. In order to allow physicists to interactively access subsets of data (e.g. for algorithm tuning and debugging before running over a full dataset) parallel analysis facilities based on PROOF have been deployed by the ALICE experiment at CERN and elsewhere. Whereas large Tier-1 centres may afford to build such facilities at the expense of their Grid farms, or exploit the large number of jobs finishing at any given time to quickly collect a number of nodes to temporarily allocate for interactive work, this is likely not to be true for smaller Tier-2 centres. Leveraging on the virtualization of highly performant multi-core machines it is possible to build a fully virtual analysis facility on the same Worker Nodes that compose an existing LCG Grid Farm. Using the Xen paravirtualization hypervisor, it is then possible to dynamically move resources from the batch instance to the interactive one when needed, minimizing latencies and wasted resources. We present the status of the prototype being developed, and some experience from the very first users.

## 1. Introduction

The computing model of the ALICE experiment [1], similarly to most HEP experiments, provide for both batch and interactive analysis modes. At this time most of the computing resources in ALICE are provided as Grid computing elements (CEs) through the AliEn middleware [2], mainly developed for batch analysis and mostly unsuitable for a production-grade interactive infrastructure, although several efforts to interactively use the Grid actually do exist—see for example [3] for a medium-scale EU project and [4] for an analysis of limitations and perspectives of the current EGEE deployment.

An extension of the ROOT framework [5] called PROOF [6] was adopted by the ALICE community in order to provide interactive resources to the ALICE community: the first example of such a facility is the Central Analysis Facility (CAF) at CERN [7].

PROOF is becoming very popular among ALICE users, and the increasing demand of interactive resources is unlikely to be satisfied by a single computing centre at CERN: several PROOF facilities are being built elsewhere, but many Grid computing centres can't afford the cost for new machines to be set up and maintained along with the Grid ones. This is why sharing resources currently dedicated to the Grid seems to be a good solution.

The computing model in ALICE comprises three "tiers" of computing centres [8] that differ in size and quality of service specifications: the Tier-0 is CERN, Tier-1s are centres with thousands of job slots and custodial storage capabilities (usually achieved with tape units) and Tier-2s are the smallest ones, which provide disk-only storage and hundreds of job slots. Tier-1s are mostly used for centrally-managed scheduled activities, while "chaotic" user analysis is primarily run on Tier-2s.

The largest centres have a high rate of finishing jobs, and one could think of temporarily dedicating the free slots for interactive analysis by running several parallel PROOF daemons as Grid jobs in a high-priority queue (Proof On Demand, developed at GSI, formerly known as gLiteProof, does that [9]). In smaller centres, such as the Tier-2 in Torino, the rate of finishing jobs is too low to have enough available parallel PROOF daemons in a reasonable amount of time.

Our solution is to share resources between the Grid and PROOF in a layer closer to the underlying architecture by using *virtualization*: we particularly look for a virtualization platform that allows us to dynamically reallocating resources (both CPU and memory) in order to dedicate them to PROOF only upon request.

## 2. Prototype implementation

Virtualization allows to run several virtual nodes providing different services as completely isolated machines within a single physical node: in our prototype each node runs a Grid worker node (where the WLCG middleware [10] is provided by gLite [11]) and a PROOF slave.

Our prototype consists of four HP ProLiant 360DL equipped with two Intel Xeon quad-core CPUs each that support Intel-VT hardware extensions [12], plus an extra node for access, management and monitoring. Three machines have 8 GB of RAM installed, one has 16 GB. The virtualization framework running on them is Xen 3.3, a free and open-source hypervisor originally developed at the University of Cambridge and now distributed by Citrix Systems, Inc. [13], because it is widely considered suitable for a production environment and it can dynamically reallocate not only CPU resources (via two parameters: the maximum CPU efficiency, where 100% corresponds to one CPU, called *cap*, and a relative priority, called *weight*), but RAM memory as well. Moreover, the Xen platform is capable of *paravirtualize* instead of *fully virtualize*, a technique that in our use case should minimize performance losses, as we will see in the following.

Our idea is that when a user needs to use the interactive facility, we reallocate the majority of the resources (RAM and CPU) to PROOF; when the user has finished running the analysis, resources are completely returned to the WNs. This dynamic scenario is schematically represented in figure 1. During this period Grid jobs slow down for a small amount of time compared to the average job duration, but they do not terminate abnormally, as shown by a specific test.

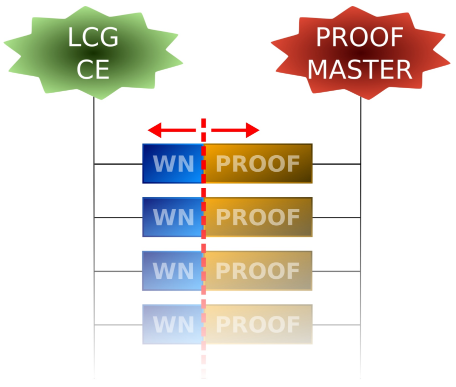

**Figure 1.** A PROOF facility (managed by the PROOF master) is run by sharing resources with a LCG Grid facility (managed by the Grid computing element): resources can be reallocated at run-time.

By removing memory from a heavy-loaded WN we expect it to start swapping to disk: since heavy swapping of the WN can potentially affect the PROOF node, each physical node is equipped with two *separated* 146 GB SAS disks, one for each virtual machine. By using both Xen and separated disks we can achieve both performance and failure isolation: this means that a failure in PROOF node does not affect running jobs on the WN.

### *2.1. Data access models*

Two different data access models can be used with PROOF. Since PROOF is developed as a Scalla/xrootd [14] plugin, the PROOF nodes can also share their disk space in a single pool where data should be staged before running the analysis. Thanks to the PROOF integration with xrootd, the PROOF master knows on which node a certain file to be analyzed is located, and sends the job close to its data instead of transferring data through the network.

This access model is certainly efficient for its low network load but it may be unfeasible: data to analyze requires a large amount of disk space, and this requirement can't be met by nodes equipped with only a few slots for small SAS disks. Moreover, if we want to maximize the slots usage we should renounce to any form of data redundancy. We should note that the staging procedure, that basically consists on replicating data from a storage element (SE) to the local PROOF pool, is a time-consuming task that results in data not immediately available to the user for analysis—data that needs to be re-staged after a disk failure. Furthermore, staging policies and procedures need to be manually managed by system administrators: human resources may not be available in small Tier-2s for such a maintenance effort.

For Tier-2s it would be more practical to use the already existing local Grid storage element, and access data directly from there, like WNs already do: maintenance would be easier for system administrators because no extra storage pools exist apart from the Grid one. A user that needs to access its data should only take care of mirroring the data on the local storage element, instead of staging it on the PROOF pool.

Despite these considerations, all tests on our prototype were run on sample data with the "PROOF pool" access model because, for testing purposes, the nodes were located on a network different from that of the local Torino SE: the SE is only visible through NAT, making a speed comparison between the two models pointless with this network topology—the PROOF pool would clearly turn out to be much faster.

Data access from the local SE will be implemented in the near future; the most important configuration feature required in order to access data from SE—*GSI authentication*—was already implemented, as explained in the next section.

### *2.2. GSI authentication*

Among several available authentication methods available for PROOF, the GSI certificate authentication has been choosen: users, in order to connect to the virtual analysis facility, need to obtain Grid-level security clearance obtained by presenting a valid x509 digital certificate to PROOF.

This authentication method has been preferred because the certificate is the same used to access Grid services (submit jobs and access data), and users are generally accustomed to it. Moreover, it provides a strong security level and, as seen before, enables us to authenticate to the SE to access data directly from there.

### *2.3. Management software*

Management of the operating system installation is performed by using Cobbler [15], a mass operating system network installation server that relies on PXE and a web server; Cobbler also manages DNS and DHCP on the virtual analysis facility LAN.

Control scripts are wrappers around Xen utilities: commands are cast on the specified machines without explicitly connecting to them, since the scripts simply use encrypted SSH passwordless connections (by using public key authentication).

A web interface based on AJAX and PHP is currently under development.

## 3. Feasibility tests

A series of tests have been conducted on the prototype in order to verify the feasibility of the virtualization approach and to stress the whole infrastructure under heavy load, in a both simulated and

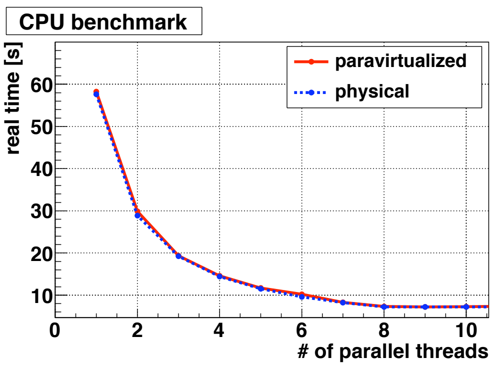

**Figure 2.** A CPU-bound task shows no performance difference between physical and virtual node.

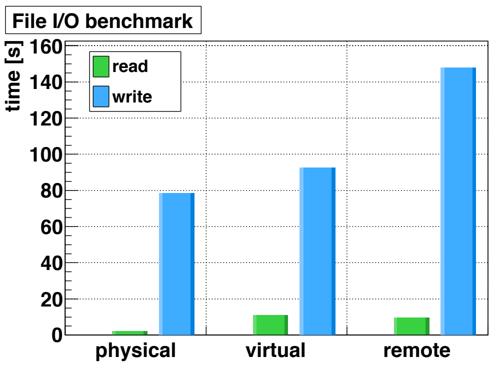

**Figure 3.** Disks of virtual machines are slower than the physical ones; remote I/O shows the slowest performances.

real production environment.

## *3.1. Specific benchmarks*
Specific benchmarks were run in CPU-only and I/O-only tests in order to compare the performance of a physical node with the one of an equally configured virtual machine: sysbench [16] was used for both tests.

The CPU-only test is a scalable prime number finder algorithm used to test the prototype with a different number of non-concurrent CPU-bound threads. Results are plotted in figure 2: a minimum is reached on both machines when the number of threads equals the number of cores, and the two superimposing lines show no or minimal performance loss in the paravirtualized host with respect to the physical one.

Disk I/O is tested by running 10 concurrent threads that read (or write) 5 GB of data, subdivided into 40 smaller files: the test has been run on a physical disk, on a virtual disk and on a remote NFS support. Results, plotted in figure 3, show that the virtual disks are slower than the physical ones, but we should consider that in a production environment the majority of I/O is done on a remote storage, that is the slowest of the three I/O methods tested, so that performance impact of the virtual disk compared to the physical one is irrelevant.

## *3.2. Behaviour under run-time resources reallocation*
This test measures the memory usage as seen by the virtual WN while lowering and increasing resources. The test was run on production-like events reconstruction that use the AliRoot [17] framework: while the event reconstruciton is running, a script alternatively removes or gives resources from a minimum of 256 MB to a maximum of 3.5 GB every two hours for a total running time of 16 hours.

Although the running jobs become extremely slow and swap space usage increases when RAM is reduced to the minimum, no job failure occurs during the whole test.

When in a production environment we remove memory from the WN to give it to the PROOF node, it takes time for the operating system to transfer data to the swap space and free the RAM as requested. The time required to reduce memory from 3.5 GB to 256 MB is about 1.5 minutes as seen in figure 4, a time that proves the feasibility of an on-demand resource reallocation.

Another test was the measurement of the average CPU efficiency (ratio between CPU time and total running time) of the sole ALICE jobs running on the four virtual WNs during 60 hours with all the CPUs fully used. Since each job uses only one core, the maximum CPU efficiency is 1. During the test, three different configuration sets are used. Only the jobs that started and terminated inside the

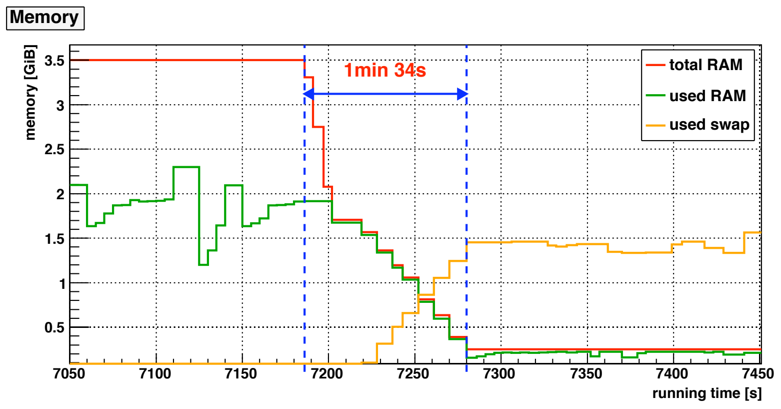

**Figure 4.** The time needed for the WN to yield memory to the PROOF node, by transiting from 3.5 GB to 256 MB, is ∼1.5 minutes, acceptable for an on-demand reallocation.

measurement window are considered. Results and configurations are shown in figure 5: at the time of each job completion, a dot indicating the average CPU efficiency is plotted.

We note that the more swap is used, the more the jobs turn from being CPU-bound to I/O bound: this fact, in addition to the reduced cap to be shared by the same number of jobs, increases the average time needed to complete each job (that can be seen by the lower density of the dots) and decreases CPU efficiency. This is clearly visible in the third resource configuration (512 MB of RAM and a cap of 100%). Again, no abnormal job terminations are observed.

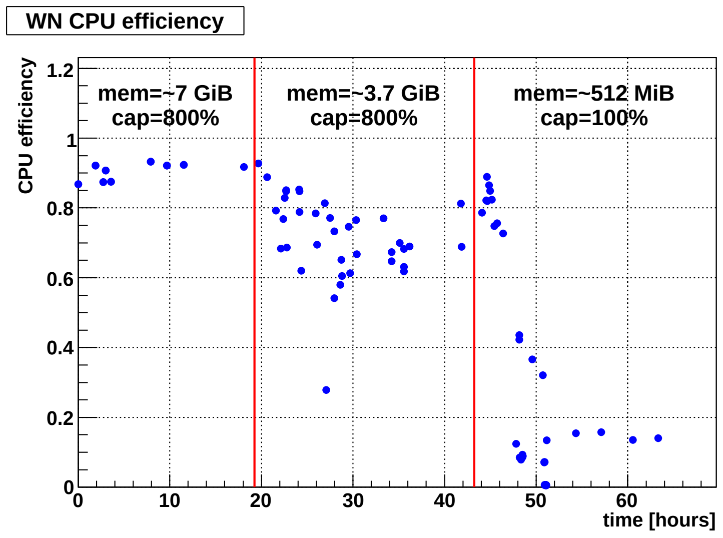

**Figure 5.** CPU efficiency by each ALICE Grid job whose execution started and terminated during interactive analisys stress test. Resources *(memory, cap)* assigned to the WN vary and are indicated on the plot. Vertical lines indicate that resources changing occurs there.

## 4. First results

The main advantage of using a virtualization layer is that no particular development effort is needed in order to make Grid and PROOF coexist: script code was only written to control the facility machines at the same time from the head node, and to implement a simple monitoring web interface (under development).

The feasibility of the virtualization layer with dynamic resources allocation is proven to be feasible, and the four virtual Grid nodes are in production along with the ordinary physical nodes since late December 2008. Mass installation software and control scripts make the facility easily and quickly scalable.

### *4.1. First performance results in production*
The facility has been tested with real interactive tasks on PROOF while the WNs were active and working, in two different configurations. Firstly, the maximum "PROOF event rate" was measured with the WN turned off: this rate is then used as reference rate. In the first configuration PROOF and the WN have the same resources allocation (same memory, same CPU weight and cap); the second configuration is the "PROOF mode", where the WN is shrunk and heavily swaps while PROOF is working.

The plot presented in figure 6 clearly shows that, in "PROOF mode", Xen makes the PROOF virtual machine behave like it were the only virtual machine on the system, since the WN is slowed very much and swaps on its own disk: real performance isolation is truly achieved with Xen and separate disks.

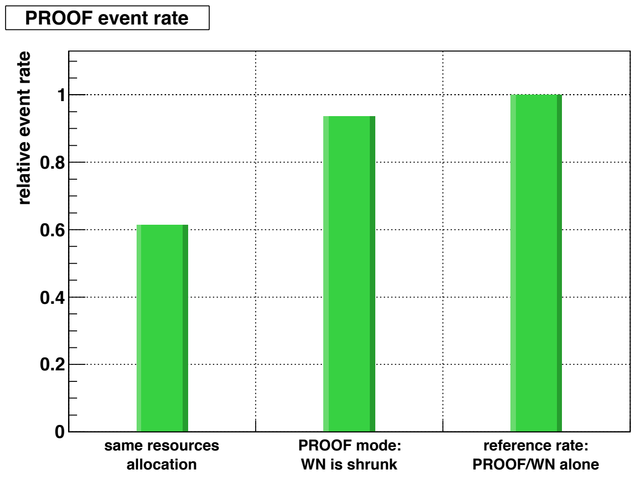

**Figure 6.** PROOF relative processing rate in two different resources allocations, while real ALICE Grid jobs are running on the WN on the same physical node; a reference maximum event rate is indicated in the third column.

## 5. Conclusions and outlook
To cater for the need of parallel interactive analysis facilities at medium-sized computer centres which already provide WLCG computing resources we developed the prototype of a system in which resources can be dynamically and temporarily reallocated between two virtual clusters whose nodes share the same physical host, one being a standard Grid worker node and the other an interactive facility running PROOF.

This paper presented a summary of both early feasibility tests and production tests, that clearly demonstrate the viability of the concept by using the Xen paravirtualization features that allow run-time reallocation of memory as well as CPU scheduling priority.

Some minimal steps must be accomplished before opening the facility to a larger number of users: network configuration needs to be revisited in order to make the local storage element directly visible to the PROOF nodes and tests should be subsequently done to compare the PROOF pool access mode with the SE access. We expect this model to be slower than the PROOF pool, but not in a relevant way, since it is already widely used by the WNs.

Although GSI authentication is already implemented, authentication credentials must be propagated in a secure way to every PROOF worker to make them capable of directly authenticating to AliEn in order to read data from the SE.

These two issues will be solved in the near future; after opening the facility to a larger base of potential users we will hopefully have a realistic feedback on their analysis experience.

Although the utilities used to manage the facility are stable and easy to use it would be better to rewrite them in order to support *different* virtualization methods: a valid abstraction layer that interacts with several hypervisors is libvirt [18]. Also, the web interface needs further developement and needs to be integrated with libvirt.

At this time resources are statically assigned to the PROOF nodes via a manual command: a better approach would be to “turn on” dynamically the facility when users need it, without neither system administrator nor user intervention. This kind of approach probably requires the PROOF Scalla plugin to be modified in order to trigger an event when a user connects to PROOF.

A last issue is represented by the current resource accounting used on the Grid: the virtual analysis facility should be integrated with existing Grid resource usage accounting infrastructures in order to avoid reallocated resources to be marked as wasted.

## References

[1] Carminati F, Foka P, Giubellino P, Morsch A, Paic G, Revol J-P, Safarik K, Schutz Y and Wiedemann U A 2004 ALICE: Physics Performace Report, Volume I *J. Phys G: Nucl. Part. Phys* **30** 11 1517–763
[2] Bagnasco S, Betev L, Buncic P, Carminati F, Cirstoiu C, Grigoras C, Hayrapetyan A, Harutyunyan A, Peters A J, Saiz P 2008 AliEn: ALICE environment on the Grid *J. Phys.: Conf. Series* **119** 6 062012
[3] The Interactive European Grid Project (http://www.i2g.eu/ and references therein)
[4] Germain-Reanud C, Texier R, Osorio A and Loomis C 2006 Grid Scheduling for Interactive Analysis *Challenges and opportunities of HealthGrids: proceedings of Healthgrid 2006* Studies in Health Technology and Informatics **120** 25–33
[5] Brun R and Rademakers F 1997 ROOT—An object-oriented data analysis framework *Nucl. Inst. & Meth. in Phys. Res.* **A 389** 81–6 (see also http://root.cern.ch/)
[6] Ballintijn M, Roland G, Brun R and Rademakers F 2003 The PROOF distributed parallel analysis framework based on ROOT *Arxiv preprint physics/0306110*
[7] Grosse-Oetringhaus J F 2008 The CERN Analysis Facility—a PROOF cluster for day-one physics analysis *J. Phys.: Conf. Ser.* **119** 072017
[8] The ALICE Collaboration 2005 ALICE technical design report of the computing *Technical report, CERN* CERN-LHCC-2005-018
[9] Malzacher P, Manafov A and Schwarz K 2008 RGLite, an interface between ROOT and gLite—PROOF on the Grid *J. Phys: Conf. Ser.* **119** 072022 (see also http://proof-on-demand.blogspot.com/)
[10] http://lcg.web.cern.ch/lcg/
[11] http://glite.web.cern.ch/glite/
[12] Neiger G, Santoni A, Leung F, Rodgers D and Uhlig R 2006 Intel virtualization technology: hardware support for efficient processor virtualization *Intel Technology Journal* **10** 3 167–77
[13] http://www.xen.org and http://www.citrix.com/
[14] Dorigo A, Elmer P, Furano F and Hanushevsky A 2005 XROOTD/TXNetFile: a highly scalable architecture for data access in the ROOT environment *TELE-INFO ’05: Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics* 1–6 (see also http://xrootd.slac.stanford.edu/)
[15] https://fedorahosted.org/cobbler/
[16] http://sysbench.sourceforge.net/
[17] http://aliceinfo.cern.ch/Offline/
[18] http://libvirt.org/