An alternative model to distribute VO software to WLCG sites based on CernVM-FS: a prototype at PIC Tier1

E Lanciotti\textsuperscript{1,3,4}, G Merino\textsuperscript{1,5}, A Bria\textsuperscript{1,3} and J Blomer\textsuperscript{2}

\textsuperscript{1}Port d’Informació Científica (PIC), Universitat Autònoma de Barcelona, Edifici D, ES-08193 Bellaterra (Barcelona) Spain
\textsuperscript{2}CERN, European Organization for Nuclear Research, Geneva, Switzerland
E-mail: elisa.lanciotti@cern.ch

Abstract. In a distributed computing model as WLCG the software of experiment specific application software has to be efficiently distributed to any site of the Grid. Application software is currently installed in a shared area of the site visible for all Worker Nodes (WNs) of the site through some protocol (NFS, AFS or other). The software is installed at the site by jobs which run on a privileged node of the computing farm where the shared area is mounted in write mode. This model presents several drawbacks which cause a non-negligible rate of job failure. An alternative model for software distribution based on the CERN Virtual Machine File System (CernVM-FS) has been tried at PIC, the Spanish Tier1 site of WLCG. The test bed used and the results are presented in this paper.

1. Introduction
The Worldwide LHC Computing Grid (WLCG) [1] provides a distributed infrastructure for processing the data of the Large Hadron Collider (LHC) [2]. In this computing model it is of fundamental importance to have an efficient and reliable system to distribute the virtual organization (VO) specific software to all sites participating in WLCG. The model currently in use consists in executing some special jobs at each site which install locally the software in a shared area. The WNs can then access it through the local network through NFS protocol (or possibly other protocols). Several issues have been observed with this model and some alternative solutions have been explored at PIC WLCG Tier1 in order to streamline the process of software distribution. One possible model is based on deploying the CernVM-FS [5] on the site WNs. According to this model the WNs access the software deployed in a central repository at CERN through http protocol by mounting a network file system (the CernVM-FS). CernVM-FS provides efficient caching functionality which reduces to the minimum the network latency when the software is accessed more times. Some of the advantages of the proposed solution are the elimination of the complicated task of installing the software site by site, and the fact that the site does not need to maintain a shared area and mount it through some protocol in all WNs.

\textsuperscript{3} Also at Institut de Física d’Altes Energies, IFAE, Edifici Cn, Universitat Autònoma de Barcelona, ES-08193 Bellaterra (Barcelona), Spain
\textsuperscript{4} Currently at CERN, European Organization for Nuclear Research, Geneva, Switzerland
\textsuperscript{5} Also at Centro de Investigaciones Energéticas Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain

Published under licence by IOP Publishing Ltd
In this paper we briefly explain the current model for software distribution (section 2) mentioning its main limitations (section 2.1), then we introduce the new model based on CernVM-FS (section 3) and we describe the tests carried on at PIC Tier1 (section 4) and the results obtained, especially in relation to LHCb software (section 5). Some very preliminary results are shown also relatively to ATLAS software. Finally some conclusions are drawn in section 6.

2. Current model for software distribution at WLCG sites
The procedure currently used to install the experiment software at grid sites consists in running some special jobs (software installation jobs). Such jobs are executed under particular conditions in order to give them high priority, as they are essential for the usability of the site and they must be assigned to a particular WN of the farm for whom the shared area is mounted in write mode. A schematic representation of the shared area setup is shown in figure 1.

2.1. Limitations of the current model
Problems related to this procedure for software distribution happen very often and make the site unavailable for running jobs. The most frequently observed issues concern the shared area availability and the software installation jobs.

The shared area is made available to the WNs through NFS (or AFS) protocol. This kind of setup presents several limitations: any outage of the NFS/AFS server makes the content of the shared area inaccessible and immediately causes the failure of all the running jobs. Or in case of high load, the NFS/AFS server can become very slow, causing the jobs to time out in the execution of the application and eventually fail. Even if in principle sites set up an infrastructure for the shared area able to serve concurrent connections from all the WNs of the computing farm, it cannot be excluded that there will be peaks of load for the NFS/AFS server. This can be due to concurrent execution of application software especially demanding for the file system hosting the software, like scripts with recursive listings of directory or code compilation and thus many system calls. These situations are difficult to foresee, as they might depend on activity of private users which is totally uncoordinated, and they are rather difficult to manage, since setups based on NFS/AFS servers are not easily scalable.

Another very frequent source of issues resides in software installation jobs. Almost daily some of these jobs fail either because there is no space left on the shared area, or because the

---

6 The type of protocol is a choice of the particular site. At PIC Tier1 the chosen protocol is NFS, this is why we will refer more often to this protocol.
path to the shared area is not correctly set, or because network connection is too slow and the execution exceeds the job timeout. Also permission problems have been often observed, i.e. in case the job has no right to write on the shared area, due to some lock on the NFS file system (known issue accessing SQLite files on an NFS file system).

The extent of the problems related to the software distribution and installation can be estimated considering the high number of tickets reported in GGUS system [3], which was about 40 for the LHCb experiment during second quarter of 2010 (last quarter as of writing), representing a non-negligible fraction of the total job failures.

3. An alternative model to distribute software based on CernVM-FS

The alternative model that we will describe in this paper is based on the CernVM-FS, a network file system developed in the framework of the CERN virtual machine (CernVM) [4] project at CERN. According to this model the WNs access the software deployed in a central repository at CERN through http protocol by mounting the CernVM-FS. CernVM-FS provides efficient caching functionality of files data and metadata, and reduces to the minimum the network latency when the software is accessed more times. During job execution the WNs access the software through the http protocol. In this way, only the programs and libraries that are actually used during run time would be accessed, and then cached locally on the WN local disk. The optimized caching mechanism of CernVM-FS limits the network traffic between the node and the origin web server, which is furthermore reduced by mean of a zlib compression of the transferred files. Additionally, one or more http proxies can be deployed at the site, in order to cache locally the software packages and reduce the load on the central software repository of CERN, as well as for reducing latency due to data transfer through the WAN.

Clear advantages of this solution are the elimination of the complicated task of installing the software site by site, and the fact that the site does not need to maintain a shared area and mount it through some protocol in all WNs.

4. Testing activity at PIC Tier1

4.1. Test bed description

At PIC Tier1 16 WNs have been put off line from the computing farm and configured to be visible under a test queue. Each node is a 8 core system, running SL5[7] and gLite 3.2 WN middleware [8].

4.2. Installation and configuration of CernVM-FS client

The used version of CernVM-FS for the tests reported in this paper is 2.47 released in August 2010. Installation and configuration of CernVM-FS client on the WN are straightforward and don’t require any change at kernel level, as the development of the file system has been made on SLC5 OS systems. Installation is done using YUM installation tool, which will install the client with a default configuration file, where the following parameters have to be set: the location of the CernVM-FS client binary; the mount point for the file system; the path to the CernVM-FS local cache; the http proxy hostname, or a list of them (for load balance and for reliability by the client’s fail-over mechanism); the remote repository URL where the software is hosted. Optionally, more repositories corresponding to different VOs can be specified, allowing the WN to execute simultaneously jobs of different experiments. All the remaining parameters were left with the default values preset in the configuration file [5].

To complete the setup, an http proxy has been deployed at the site in order to reduce the network traffic over the WAN and limit the load on the origin web server. A Squid server has been installed for this purpose at PIC following the standard configuration used for the Squid caches deployed at CMS sites for jobs to access the Conditions Data through FroNTier [9] [10].
4.3. Test description
The test consists in executing a LHCb DaVinci\textsuperscript{7} test job, after setting the environment in order to read the software through the CernVM-FS. This was done by setting the \texttt{MYSITEROOT} environment variable pointing to the mount point of the CernVM-FS instead of the usual path to the shared area. This simple test job is made up of two parts: first it executes the LHCb script necessary to set the environment previous to the job execution, and then it runs the DaVinci application on an empty dataset, therefore just spending time in reading the software, and without making any computation. A complete test run consists of submitting to the queue an increasing number of concurrent jobs, first submitting one job per WN, and then increasing up to the saturation of the node cores, resulting in 128 total jobs (the total number of job slots available for 16 WNs with 8 cores each). In figure 3 the execution time is displayed separately for the initial setup script and for the DaVinci execution versus the total number of jobs submitted to the test queue. For the first point the total jobs are 16, which corresponds to one job per WN, and so on up to the last point corresponding to the saturation of the available job slots. The error bar represents the standard deviation on the execution time.

![Graph 3: Execution time versus number of concurrent jobs for the setup script execution (green points) and for the DaVinci application execution (red) versus the total number of jobs in the test bed.](image)

5. Results
As a first objective the execution time of a test job has been measured and its dependance with the number of parallel jobs.

![Graph 4: Network traffic for the Squid server in correspondence with a run started at 18:10.](image)

![Graph 5: Network traffic for a node of the test bed for a run started at 18:10 (the peak on the left should be ignored, as it is due to a previous test).](image)

For the LHCb test job it is interesting to watch separately the execution time of the preliminary setup script and the execution time of the Davinci application. This is because the setup script is the most demanding part in terms of system calls as it has to list a huge

\textsuperscript{7} LHCb package for data analysis
number of directories to search for libraries and binaries. In case the NFS/AFS server is under high load and the access to the shared area is particularly slow the setup script can time out and cause the job to fail. Currently the time out is set to 600s and this type of failures are observed quite often, as mentioned in section 2.1.

The basic features of the CernVM-FS setup are illustrated in figures 4 and 5. Figure 4 shows a screenshot of the outgoing (blue) network traffic of the http proxy (Squid server) when a new DaVinci release is requested and it is not present in the local WN caches. The plot corresponds to the 16 WNs requesting for the new release simultaneously. Figure 5 shows the incoming (green) network traffic on one of the WNs. It is illustrative to note that no network traffic appears after 18:15 even if runs were executed every 15 min showing the effect of the local caches on the WNs.

The results relative to the execution of the setup script for the LHCb test job are shown in figure 6, accessing the software through CernVM-FS, and in figure 7, using the current NFS setup currently in production. The plot shows a very low execution time for CernVM-FS, and no dependance with the increasing number of parallel jobs. On the other side, reading the software from the shared area through NFS protocol there is a clear dependance with the number of parallel jobs running on the same node. Several runs have been taken, when the load of the NFS server (quantified by the CPU percentage usage) was at different levels, from almost 0% to 80%, depending on the load on the Tier1 at that moment, since the production NFS server was used to do the measurements. However this has been verified to be an effect on client side, as the total number of jobs (128) is by far too low to generate a load on the NFS server. Moreover the same behaviour has been observed submitting an increasing number of parallel jobs to only one node.

Some preliminary tests have been done also with an ATLAS job, which runs ATLAS Athena application. The results are shown in figure 8 for several runs, accessing the software through CernVM-FS, and with NFS (see legend). The execution time is of the same order with both protocols.

An important remark is that the results obtained with the LHCb test job and the ATLAS test job cannot be directly compared, as they make a different use of the system calls. A thorough study of the behavior of CernVM-FS with respect to the amount of system calls has been done [6], and goes beyond the scope of this paper.

During the tests also the disk space needed on the WN for the CernVM-FS cache was carefully monitored. Cleaning the cache on the WN and then executing different versions of the DaVinci
application showed that the cache size is of order 900MB for one version, and any additional version of the same application adds some 100MB. Considering that LHCb computing activities make use of four different software packages which have similar structure and size, and that more versions of DaVinci might be in use\(^8\) the total cache size needed per WN can be estimated to less than 10GB.

6. Conclusions
The use of the CernVM-FS for distributing VO specific software to a Tier1 site has been tried with a prototype set up at PIC Tier1. Functionality tests performed with LHCb and ATLAS software show promising results: the job execution time is equal or lower than the one observed using the current setup based on a shared area accessed through NFS software. The network traffic seen between the WNs and http proxy and between this and the central servers at CERN are modest, illustrating the efficiency of the local cache and data compression policies of CernVM-FS. Also, the size of the local disk cache needed for a realistic application has been found to be a small fraction of the size of the local disk available in a typical WN today. PIC was the first Tier1 site to test in a systematic way this model for software distribution and prove its functionality. Next step would be systematic scalability tests to prove that this model can cope with the use case of LHC experiments at realistic load levels and can be safely put in production.

Acknowledgements
The Port d’Informació Científica (PIC) is maintained through a collaboration between the Generalitat de Catalunya, CIEMAT, IFAE and the Universitat Autònoma de Barcelona. This work was supported in part by grant FPA2007-66152-C02-00 from the Ministerio de Educación y Ciencia, Spain.

References
[1] Worldwide LHC Computing Grid http://lfc.web.cern.ch/
[2] The Large Hadron Collider http://lhcb.web.cern.ch
[3] J. Phys.: Conf. Ser. 219 062032 Authors: T Antoni at al.
[4] J. Phys.: Conf. Ser. 219 042003 Authors: P Buncic at al.
[5] J Blomer, https://cernvm.cern.ch/project/trac/cernvm/browser/cvmfs-tr/cvmfstechpreview.pdf
[6] J Blomer “CernVM File System v3: Features and Future” http://indico.cern.ch/conferenceDisplay.py?confId=89681
[7] Scientific Linux http://www.scientificlinux.org/
[8] gLite middleware http://glite.cern.ch/
[9] http://frontier.cern.ch
[10] B Blumenfeld et al. 2007. CMS conditions data access using FroNTier. FERMILAB-CONF-07-526-CD. Proceedings CHEP 07.

\(^8\) DaVinci is the application used for private user analysis, who could use versions older than the last released version.