Comparative Investigation of Shared Filesystems for the LHCb Online Cluster

S. Vijay Kartik, Niko Neufeld
CERN (European Organization for Nuclear Research), Switzerland
E-mail: vijay.kartik@cern.ch

Abstract. This paper describes the investigative study undertaken to evaluate shared filesystem performance and suitability in the LHCb Online environment. Particular focus is given to the measurements and field tests designed and performed on an in-house OpenAFS setup; related comparisons with NFSv4 and GPFS (a clustered filesystem from IBM) are presented. The motivation for the investigation and the test setup arises from the need to serve common user-space like home directories, experiment software and control areas, and clustered log areas. Since the operational requirements on such user-space are stringent in terms of read-write operations (in frequency and access speed) and unobtrusive data relocation, test results are presented with emphasis on file-level performance, stability and “high-availability” of the shared filesystems. Use cases specific to the experiment operation in LHCb, including the specific handling of shared filesystems served to a cluster of 1500 diskless nodes, are described. Issues of prematurely expiring authenticated sessions are explicitly addressed, keeping in mind long-running analysis jobs on the Online cluster. In addition, quantitative test results are also presented with alternatives including NFSv4. Comparative measurements of filesystem performance benchmarks are presented, which are seen to be used as reference for decisions on potential migration of the current storage solution deployed in the LHCb online cluster.

1. Introduction
As one of the four large experiments on the Large Hadron Collider (LHC) at CERN, LHCb houses a very large Online cluster, comprising around 4000 computers (including about 1500 diskless nodes). The cluster is a homogenous mixture of Windows and Linux clients and servers, hosting home directories for LHCb users in addition to the experiment project space, including software, control tools and dedicated data areas. A total of approximately 120 TB of data is served through the current storage infrastructure, which employs a distributed filesystem (NFSv3) over a commercial storage solution based on CVFS [1]. The current storage system has a home-grown NFS server cluster setup that re-exports all project-, satellite- and home-areas. The low-level changes that have been made to enable this are non-standard and thus require careful tuning and upkeep. While this keeps costs low, the administrative effort is sizeable and often requires interruptions and downtime for users, which is not ideal. Since the Online cluster is managed by a fairly small team of system administrators, the need for a shared filesystem with verified stability of functioning and ease of management (viz. backups, server migration/replacement) is clear. This paper describes initial efforts in the direction of evaluating stable alternatives with comparable performance. The goal of the evaluation is to find a solution that scales well, provides long durations of stable running, keeps interventions
invisible to clients, and incurs minimal administrative labour. OpenAFS is a viable alternative that is being evaluated, since it provides all the required functionality and moreover, has been deployed at CERN since a long time [2]. Brief performance results about NFSv4 are also listed here, and comparisons with NFSv3 are made to provide perspective for our specific case in the Online cluster.

2. Filesystem requirements foreseen for LHCb
Since the LHCb Online cluster has its own particular client usage pattern, potential solutions for filesystems need to provide a minimum set of functionalities that ensure acceptable performance under this usage, in addition to the more general requirements from shared filesystems.

(i) High availability: Interruptions (due to regular maintenance as well as problem resolution) on the fileserver end should be invisible to users.

(ii) Low administrative burden: Management of the filesystem should be inexpensive in man-hours, particularly in connection with backups and server movement or hardware changes. Ease of installation and setup is desirable in addition.

(iii) Exports on diskless nodes: Since a sizeable proportion of the cluster is in the form of diskless nodes, the shared filesystem visible on them must be identical to the regular working nodes for the end-users. This means that only limited amounts of memory can be spared for local file caching as the diskless nodes usually run memory-intensive processes for other computations.

(iv) Equivalent filesystem exports to Windows and Linux clients: Many users of the Online system run Windows clients, and it is imperative to expose an identical filesystem irrespective of the operating system being used.

(v) Handle long-running jobs gracefully: An important requirement for the filesystem is to retain user sessions through extended authenticated lifetimes, in order to avoid abruptly interrupting/killing long-running analysis jobs that are launched in the Online cluster.

(vi) Handle high volume of small files: A typical use case of the Online cluster involves a high number of read operations on several configuration and project files for trigger processes at the start of data-taking in LHCb. The shared filesystem thus must be able to handle voluminous read requests over a short duration of few minutes.

3. State of the art in LHCb
The current shared filesystem infrastructure in the LHCb Online cluster is based on NFSv3. Approximately 120 TB are served to all client machines, running through a self-managed cluster of six exporting nodes hosting files from an underlying CVFS-based commercial storage system.

![Figure 1. Current storage solution setup in the Online cluster.](image-url)
The system works well for current needs in LHCb Online, owing to multiple iterations of specific tuning in the past and several handmade setups to ensure the desired performance figures. The motivation for the current setup was due to a) low cost, b) ease of administration of CVFS and c) hand-made setup for server cluster. This has a direct influence on the structure of the shared filesystem, making it a) more strenuous to maintain and b) prone to higher downtime. Regular periods of downtime for maintenance are usually foreseen in typical clusters deployed for high performance computing, but this is undesirable at LHCb where 24-hour availability is essential for data-taking as well as continuing analysis work. Consequently, the interventions on the current storage infrastructure leading to service interruption are one of the major reasons for this evaluation of alternatives.

4. Current test setup
The performance of the existing storage infrastructure was measured against the results obtained from different test setups configured to host shared filesystems based on OpenAFS and NFSv4 (among others like IBM’s General Parallel File System (GPFS)), as these two options seemed to meet most of the requirements of the Online cluster, as listed in section 2. The test setups were installed in the Online cluster in order to replicate typical running workloads to make relevant performance measurements.

![Figure 2. Structure of test setups for OpenAFS and NFSv4 in the LHCb Online cluster.](image)

For OpenAFS, the current configuration contains two fileservers, one database server, and ten regular client nodes. Additional diskless client nodes were included for tests to reproduce identical loads on the filesystem as seen during peak periods of storage access during normal running of LHCb. To complete the supporting structure needed for a fully functional OpenAFS setup, an independent Kerberos domain (with a separate authentication realm) and an isolated DNS zone were set up, ensuring that network traffic from the tests did not perturb the rest of the network. The file servers had a 1 Gb network link each, and exported the OpenAFS filesystem to at most 50-60 clients during peak periods of testing. All tests were performed on an ‘invisible’ OpenAFS cell, continuing the theme of total isolation from the rest of the Online cluster and the world.

A small test setup using NFSv4 was also configured on file servers with similar hardware, and performance measurements were made with the same storage back-end to avoid any effects due to hardware configurations. Fig. 2 represents these test setups. It is worth mentioning at
this point that the hard disks used as the storage back-end for these tests consistently measured at around 50 MB/s during local sequential writing tests.

In addition, an ad-hoc system based on GPFS was briefly exercised (while concurrent tests for evaluating GPFS were being performed by another group at LHCb Online), and some preliminary numbers were obtained to enable a global overview of the performance of GPFS.

For NFSv3, the current storage infrastructure (including the handmade cluster setup) at LHCb was used to perform tests.

5. Filesystem performance evaluation

To ensure fair comparisons while evaluating different options for a shared filesystem solution, it was decided to move away from a purely benchmark-driven focus and also include custom tests to replicate ‘real’ situations that are encountered during normal running in LHCb Online. Simple test results from the standard benchmarks IOzone and Bonnie++ are presented here (in addition) for completeness.

Initial testing also involved making comparisons with measurements on the local disks (with ext3 filesystems) of both server and client machines; this was necessary in order to account for the performance of underlying hardware, and also to aid in tuning configuration parameters of the shared filesystems to maximise performance. Hand-made performance tests measured the time taken to complete typical operations undertaken in the Online cluster, including scalability tests with the computing farm of diskless nodes.

![Figure 3. Comparative measurements on different shared filesystems - write speeds on sequential I/O [IOzone, Bonnie++].](image1)

![Figure 4. LHCb configuration time with different filesystems running varying number of trigger tasks on the diskless computing farm.](image2)

Fig. 3 shows sequential I/O performance for clients accessing the OpenAFS filesystem. Taking into consideration the initial performance boosts from intermediate caches, one can notice that for reasonably sized files (under 1 GB), OpenAFS performance remains at writing speeds of ~80 MB/s and reading speeds of ~65 MB/s. Similar measurements for an NFSv4 setup based on the same hardware were seen to be around 75 MB/s and 55 MB/s respectively. An initial test setup based on IBM’s GPFS filesystem demonstrated much higher performance measures of ~90 MB/s, and is a good candidate for further investigation. The impact of these measures on the real performance of the filesystem is unclear, though, since large proportions of the operations on the current shared filesystem during normal operation in LHCb are of a random nature.
Since the goal of the evaluation was to help choose a shared filesystem that would cope well with the particular running needs of LHCb during regular data-taking, more detailed test setups were put in place to mimic running conditions of LHCb Online; a brief overview of the test setup is detailed here:

- export project data to diskless nodes configured as OpenAFS (and alternatively, NFSv4) clients
- start regular “data-taking” tasks on diskless nodes
- measure configuration and running performance of complete project space (involving heavy random I/O on the filesystem in initial stages)
- make comparisons with different configurations/parameter tuning, and with the current NFSv3-based setup
- stress-test the setup to identify potential bottlenecks due to configuration, and also to test the stability of the filesystem

The configuration time of the diskless computing farm is an indication of the performance of the underlying filesystem for a high volume of simultaneous random I/O from thousands of clients, since the configuration includes all tasks on all nodes requesting different software libraries and configuration files. The results for OpenAFS are compared with earlier tests done by LHCb Online in the same vein, and are presented in Fig. 4.

The comparison is not completely fair as the other figures quoted are for NFSv3 and ‘RFS’, a homegrown filesystem with an inbuilt caching mechanism designed specifically (and solely) for the purpose of running the LHCb project [3]. The performance of these filesystems is indicated from previous benchmarks, but cannot be repeated now because the configuration procedure has changed entirely. What is clear, though, is that OpenAFS has comparable figures for a sizable number of diskless nodes from a much smaller storage back-end, and can be expected to scale well similarly for a further increase in configuration tasks.

6. Comments and proposed modifications
The need for a modified filesystem is evident, particularly in light of decreasing the administrative burden of maintaining regular backups and performing uninterrupted site-wide interventions. The tests that have been completed are indicative of general usage of the Online infrastructure, and the performance measurements are reasonable indicators for this. A few comments are required, though, to qualify the measured values reported here. Valid inferences can only be drawn after taking into consideration the effects of the different setups for the filesystems.

The OpenAFS test setup includes one database server, which redirects all file access requests from all clients to the different file servers. Since each file server is expected to comfortably serve 100 clients or more, it is clear that the performance would increase if the number of database servers were increased. This is especially true for a disjointed setup based on file access patterns (discussed later, in section 7). The NFSv4 test results show performance similar to OpenAFS, but current figures for its performance on diskless clients are unavailable as the diskless computing farm now uses RFS. Preliminary tests performed on GPFS show very attractive performance values, and highly stable operation, with the added note that this ad-hoc test setup was based on a SAN-based storage back-end with multiple high performance disks, with large amounts of server-side caching (in the vicinity of 10 GB of memory-cache per file server). GPFS is currently being evaluated for other high performance requirements in the Online cluster as well.

Keeping in mind the major uses for LHCb Online (listed in section 2), OpenAFS has shown to lend itself well to requirements, and the performance seems to be reasonable for most tasks involving regular user operations. A proposed structure for a shared filesystem setup
Figure 5. Proposal for a modified structure of an OpenAFS-based filesystem in LHCb Online.

is presented in Fig. 5. Usage patterns in LHCb are well defined - read-‘only’ accesses for files in the ‘histograms’ and ‘software’ areas, write-‘only’ accesses in the DAQ (data acquisition) area, and regular read-write accesses in the ‘group’ area and user home directories. Of course, small volumes of combined read/write accesses occur in all areas of the filesystem, but the major proportions of the filesystem usage are easily separable. With this view, a modified shared filesystem structure is proposed here, which makes use of these (mostly) isolated usage patterns. As indicated in Fig. 5, this leads to a stable file server backend with the easy option of regular backups on separate nodes which may be then exported online at all times, to redirect much of the client access traffic, thus avoiding heavy loads on a few machines. Multiple database servers can also be tuned to list file servers with varying priority depending on the file access patterns, thus further distributing the overall load. Since OpenAFS provides inbuilt facilities for most operations needed to achieve this modified, ‘live’ tuning, the administrative effort for this proposal is also minimal.

7. Conclusion
There is a clear indication that distributed filesystems like OpenAFS and GPFS have a marked advantage for I/O of a fragmented nature. The test results reported above show consistently high read/write speeds with OpenAFS for files up to 1 GB in size, in comparison to local disks as well as older filesystems like NFSv3. GPFS has very high measured values, and is certainly worth analysing further in-depth, since it is proprietary and requires a license for deployment. NFSv4 is said to be a sizeable improvement over NFSv3, with the inclusion of AFS-like access control and security, but needs maintenance and stable back-end storage clusters (like the current cluster in LHCb Online) for continued usage.

OpenAFS meets many of the needs of LHCb Online, and has added benefits of tunable
authentication tokens, which are critical for LHCb where long-running analysis jobs are common on the cluster. This tuning has been found to work very well and provides an extra mode of control over user authorization on the cluster. A positive side-effect of a migration to OpenAFS is the ability to integrate better with CERN IT for user databases and general site-wide authentication. The proposed structure may be studied and tested to conclude about the viability of the scheme for the Online cluster.

8. References

[1] Quantum Corporation, “StorNext Technical Documentation”, retrieved from http://www.quantum.com/ServiceandSupport/SoftwareandDocumentationDownloads/SNMS/Index.aspx, 2012.

[2] CERN Information Technology Department, “Functional Element Description: AFS”, retrieved from http://information-technology.web.cern.ch/services/fe/afs, 2012.

[3] J. M. Caicedo Carvajal, R. Schwemmer, N. Neufeld, “Optimising the HLT Farm at the LHCb Experiment”, Proc. IEEE NPSS Real Time Conference, 2010.

[4] J. M. Caicedo Carvajal, N. Neufeld, R. Schwemmer, “Evaluation of Cluster File Systems for the LHCb Experiment”, Proc. Int. Conf. Computing in High Energy and Nuclear Physics, 2010.

[5] IBM Corporation, “OpenAFS Administration Guide”, OpenAFS Documentation, retrieved from http://docs.openafs.org/AdminGuide/index.html, 2000.