Samba and CERNBox:

Providing online access to Windows-based users at CERN

Giuseppe Lo Presti[1,*], *Aritz* Brosa Iartza[1], and *Sebastian* Bukowiec[1]

[1]CERN - European Organization for Nuclear Research, 1211 Geneva, CH

Abstract. This paper presents the experience in providing CERN users with direct online access to their EOS/CERNBox-powered user storage from Windows. In production for about 15 months, a High-Available Samba cluster is regularly used by a significant fraction of the CERN user base, following the migration of their central home folders from Microsoft DFS in the context of CERN's strategy to move to open source solutions.
We describe the configuration of the cluster, which is based on standard components: the EOS-backed CERNBox storage is mounted via FUSE, and an additional mount provided by CephFS is used to share the cluster's state. Further, we describe some typical shortcomings of such a setup and how they were tackled. Finally, we show how such an additional access method fits in the bigger picture, where the storage is seamlessly accessed by user jobs, sync clients, FUSE/Samba mounts as well as the web UI, whilst aiming at a consistent view and user experience.

1 Introduction

At CERN, the offer of personal storage for the user community has historically been based on two independent services: on one hand the Microsoft Distributed File System (DFS) [11], for Windows-based users, and on the other hand the Andrew File System (AFS), for Linux-based users. However, cross-platform interoperability was not optimal: within the Organization, users from several universities and institutions work together, therefore collaboration is one of the key elements.

Providing a universal CERN home directory with easy access to user files from any device in any location enables and fosters collaboration and mobility. CERNBox [1] is built upon modern sync & share functionalities and it is the adopted solution to fulfill the requirements of a universal home directory service, supporting any OS on the market. But as CERNBox is built on top of Linux, the Samba service presented in this paper is instrumental to meeting such requirements.

This paper is structured as follows. After an introduction about the current DFS and CERNBox services in section 2, the architecture and motivation of the Samba service is described, and its operations and monitoring are discussed in section 3. Section 4 describes some performance measurements executed to compare the two services. Finally, section 5 presents the state and evolution of the service, and section 6 discusses the outlook for the future.

*e-mail: giuseppe.lopresti@cern.ch

2 Context and Motivation

2.1 The Distributed File System

DFS [11] is one of the central services provided to all CERN users. It provides the ability to logically group network shares on multiple servers and to transparently link shares into a single hierarchical tree-like structure, the DFS Namespace.

DFS is used at CERN for two use cases: Microsoft Windows [10] so-called home folders, including *Desktop, Documents, Music, Pictures, Videos, Favourites*, and project/work spaces, including departmental spaces and scratch areas.

In both situations, the network shares are accessed via the Microsoft Server Message Block (SMB) file-sharing protocol, also known as Common Internet File System (CIFS) protocol. The major difference is the volume of space available for the users and how many users access the shares. The home folder is strictly tied to a single user. The share is owned by the user and in most situations files are accessed only by the owner. The home folders are linked with the user by an Active Directory group policy, which sets the folder redirection to the network location where the user's home folder is stored. On laptops, the offline files policy is enabled to make network files available to the user, even if the connection to the server is unavailable. By default, a user has assigned 2 GB of storage space and they are entitled to increase it up to 10 GB using a self-service resource portal.

Project spaces are offered on a self-service basis where the user can specify the desired amount of storage space. The shares are accessed by multiple users very often with different privileges on different levels of the folders' structure. Currently, the DFS service stores around $2,700$ project spaces ranging in size from several MBs to around 40 TB.

Following the Organization's strategy to prefer open source solutions upon commercial ones and reduce license costs [4], in September 2019 the Windows home folders started to be migrated from DFS to CERNBox, and as of December 2019 DFS stopped to be used as a default solution for Windows home folders. Currently, more than $29,000$ user accounts have been migrated to CERNBox.

2.2 CERNBox

Built upon the EOS [2] storage, CERNBox is the CERN cloud storage hub and has become one of the most commonly used platforms for collaboration in the Organization. The goal is to consolidate different home directory use cases into a widely accessible unified service, whilst supporting typical Physics analysis use cases, which involve thousands of user batch jobs accessing the underlying EOS storage and asynchronously producing output files. EOS is also proposed to complement AFS as Linux-based storage.

CERNBox is based on the ownCloud [7] software stack, and allows several access methods as depicted in Figure 1.

With the DFS migration taking place, an increasingly larger fraction of users needs to access the storage from Windows via direct native methods, without synchronization to the local file system. Use cases ranging from working with large files, smooth file transfer between different operating systems, engineering simulations, automatic file generation and processing, or network shares in shared environments like Windows Terminal Servers and Linux Application Gateways that have limited local storage and hundreds of users, are only a few examples. To date, the natural solution to implement such access method is via the Samba [6] software suite, to enable Windows users access to the Linux-based EOS storage through the SMB/CIFS protocol.

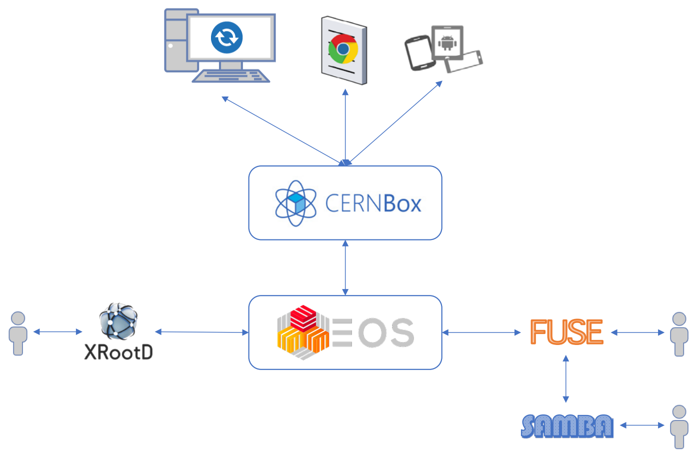


Figure 1. CERNBox can be accessed through several methods and a variety of devices and platforms.

3 Architecture of the Samba service

3.1 Overview

Developed and maintained by the open source community for nearly 30 years, Samba allows setting up a CIFS gateway for accessing a variety of Linux-based storages via dedicated VFS (Virtual File System) modules, including locally mounted file systems.

To provide a high available deployment, `HAProxy` was initially considered, owing to its simple configuration that well adapts to our on-premise OpenStack-based cloud. However, preliminary tests had shown that the latency introduced by such an extra layer was extremely penalizing the user experience. Therefore, the native high-available solution provided by the Samba suite was implemented, based on the `ctdb` daemon. This solution is based on binding the network interfaces of the nodes running `ctdb` to *floating* IP addresses: such a requirement is incompatible with CERN's cloud infrastructure due to internal constraints, and a bare-metal deployment was chosen instead, where four nodes can all bind to four additional IP addresses when being operated in a `ctdb`-managed cluster. A network alias exposing all four floating IP addresses was then made available to load balance user requests across the cluster.

An interesting property of how failures are handled by `ctdb` is the fact that at any moment in time *all* floating IPs are bound to at least one physical node, provided that there's at least one node up and running in the cluster. In other words, all IPs are always available even when up to *n-1* nodes are down for maintenance or unreachable, thus easing standard fabric management operations whilst maintaining high service availability.

Given the stateful nature of the SMB/CIFS protocol, and the intrinsic differences between Windows-managed and POSIX-compliant file systems, Samba implements a persistency layer that allows client requests to be translated to POSIX calls. In a high-available setup such layer must be shared across the nodes: this is achieved by a small CephFS-based shared volume, to avoid single points of failure. A typical alternative is to use an NFS-exported share, where all nodes mount an NFS share exported by one of the nodes. This model has been considered and tested as a failure scenario, should the CephFS share be unavailable due e.g. to extended service downtime.

Finally, at CERN several EOS instances have been deployed to serve the different use cases, including 5 instances for user home directories, 3 instances for shared project/workspace areas, and further instances for experiment-specific storage and other activities. As a Samba VFS module for EOS is not available, we have deployed the necessary configuration to FUSE-mount all required EOS instances in the Samba gateways.

3.2 Configuration and Access Control

Each Samba gateway must be configured to join the network's *Domain Controllers*, currently implemented at CERN by Microsoft Active Directory. This is achieved via a shared secret in the form of a keytab file, which represents the whole cluster at once from the Domain Controllers' perspective. Users' authentication can thus be performed against Active Directory, and their identifier (`SID`, in Windows realms) can be translated to standard Unix credentials (`uid`, `gid`). This mapping is controlled by the `idmap` configuration parameters: Samba provides an ID mapping backend named `ad` in order to interface to Active Directory.

The authorization step is delegated to EOS: the Samba nodes are declared as *Unix gateways* to all concerned EOS instances, thus allowing to act on behalf of the users. A security hazard with this model is that the Samba nodes have privileged access to EOS: however, this is mitigated by having the cluster not exposed outside the CERN firewall, owing to the fact that SMB access is not offered outside the Organization.

Advanced authorization mechanisms such as Access Control Lists (ACLs) are also supported via a dedicated VFS plugin, which implements the *Rich ACL* semantics and converts them to EOS ACLs. This plugin was developed in-house [3] and is going to be deployed soon in production for general use.

Finally, an important element of the configuration lies in how file locking is handled by Samba. A common use case is the concurrent editing of office files, and typical applications like Microsoft Office and LibreOffice rely on the ability to lock the file being edited in order to prevent multiple users from overwriting each other. To support this use case, files must be uniquely identified across the Samba cluster. The solution is to use a Samba VFS module to enable such unique `fileid`'s naming:

```
vfs objects = fileid
fileid:algorithm = fsname
```

In addition, CERNBox offers web-based editors for office files as part of its portfolio [1], therefore the interplay between Desktop applications and web-based engines must be carefully considered. To this end, appropriate hooks have been introduced in the web-based applications to take into account when files are being edited via the Samba service, and symmetrically a user accessing an office file via Samba will be denied write access if the file is already locked in a web-based application [5]. This ensures the maximum data integrity for the user, regardless the access method, providing a consistent view of the whole system.

3.3 Monitoring and Alerting

As hundreds of users access the service on a daily basis, a robust probing and alarming infrastructure has been put in place from inception, in order to intervene should issues arise either in the EOS storage backend or in the Samba software stack itself.

For this purpose, a dedicated probe was developed, that is continuously executed in all nodes of the cluster. The probe on one hand extracts useful information from the logs, and on the other hand it actively tests the backend by performing elementary operations on all FUSE-mounted storage endpoints.

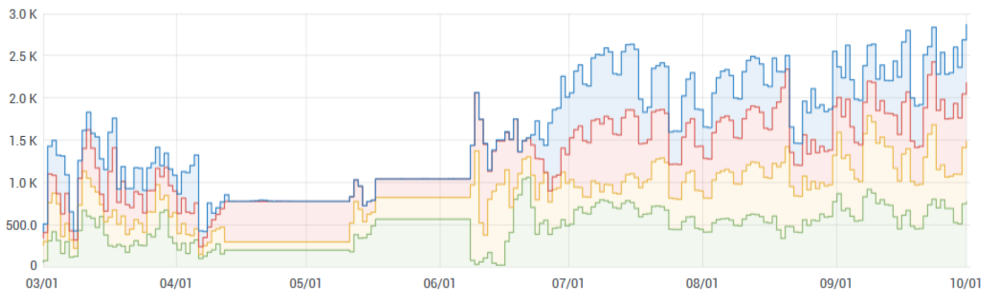


Figure 2. Service usage during 2020, in terms of counts of active connections to the four nodes in the cluster. The usage significantly increased during summer, and the service is now stably serving over 3, 000 active connections, with a total rate of I/O operations in excess of 10 kHz.

For the former, we look at specific signatures in the Samba and EOS FUSE logs as they provide insights on the status of the service, where typically ongoing issues on certain components can be identified before they make a real impact on users. The signatures identification was part of the tuning process of the system.

For the latter, active tests of the EOS backends increased the overall sensitivity: there have been cases where the probe detects issues, which are not identified by previously available alerting systems. Furthermore, as the EOS FUSE implementation keeps evolving and issues do regularly happen, an automatic mechanism for detecting blocked FUSE mounts has been put in place, and its action is fed as well to the monitoring system.

All information is gathered in a dashboard based on Grafana, a popular platform for monitoring widely used at CERN, and specific alerts have been implemented within the dashboard itself, following operational and troubleshooting experience. This provides immediate feedback about the current usage and common Service Level Indicators (SLIs), such as current number of active connections, current number of open files, or total I/O operations per second served by the cluster, as well as information on the responsiveness of the system, typically measured by latency measurements. In particular, experience has shown that latency is the most critical SLI, as issues in this area are extremely visible to users.

Finally, a real end-user test is performed as well on a regular basis from external Windows nodes, to validate the availability and the performance of the service.

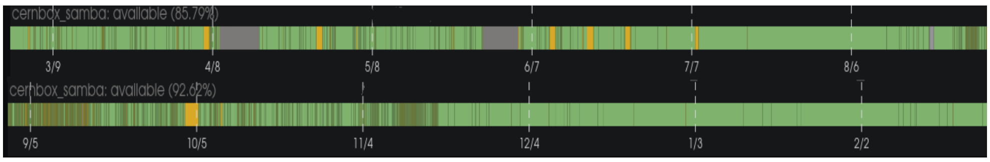


Figure 3. Samba Service Availability measured over a year (from 01/03/2020 to 25/02/2021), where yellow bands correspond to a degraded state (grey for no data). A general improvement can be noted, in particular towards end of 2020, notwithstanding the increase in its usage.

4 Performance Measurements

4.1 Comparing the Samba service with DFS

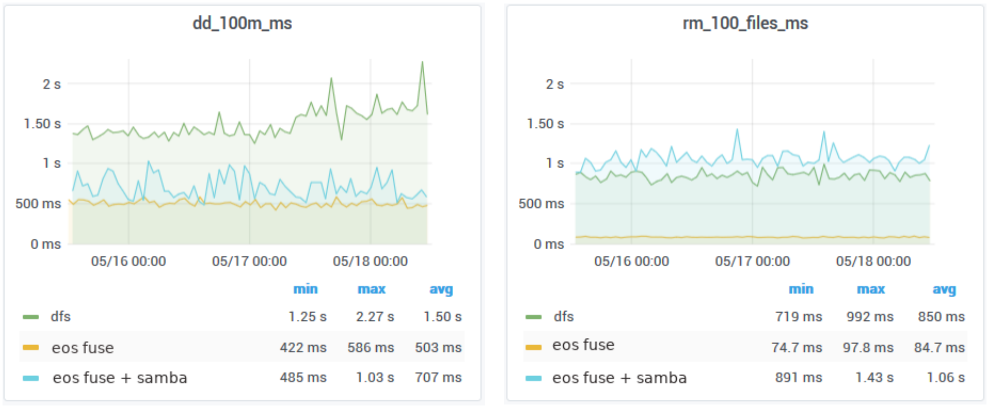
In order to compare the performance of Samba and DFS, a benchmark was established, based on tests that are continuously run against the relevant targets. These tests are based on elementary file system actions, including shell-oriented operations such as copying predefined chunks of data via `dd`, as well as `touch`, removal, listing, and moving of files, in order to exercise common metadata operations. In addition, tests involving `git`, `tar`, and `zip` are also part of the test suite.

Such tests have been run against several targets, in order to compare the different server implementations:

- a Windows share on a local Windows file server, to serve as baseline;
- the same share served by DFS, to measure the overhead introduced by the DFS Namespace;
- a Samba share on a local file system, as baseline of the Samba software suite itself;
- a Samba share over EOS.

4.2 Selection of the Benchmark

These tests have shown that some operations have a higher performance hit in Samba when compared to DFS. As shown in Figure 4, typical data-intensive operations perform better on the Samba setup, owing to the larger bandwidth offered by the underlying storage. On the contrary, DFS outperforms Samba on typical metadata-intensive operations, though the underlying storage as measured from Linux via FUSE performs remarkably better than both DFS and Samba. In particular, creating and removing a large amount of files from a given folder is among the most impacted actions: this holds true also for Samba shares on a local file system, suggesting that the SMB protocol inherently brings in a significant overhead. Therefore, we defined as a benchmark a combination of such actions, and used it as a reference to measure how each change or improvement affected the service.



dd_100m_ms

	min	max	avg
dfs	1.25 s	2.27 s	1.50 s
eos fuse	422 ms	586 ms	503 ms
eos fuse + samba	485 ms	1.03 s	707 ms

rm_100_files_ms

	min	max	avg
dfs	719 ms	992 ms	850 ms
eos fuse	74.7 ms	97.8 ms	84.7 ms
eos fuse + samba	891 ms	1.43 s	1.06 s

Figure 4. Execution time of a data-intensive (`dd`) vs. a metadata-intensive (`rm`) operation, sampled every hour over a period of 3 days.

The benchmark has been written as a PowerShell script and run from a dedicated Windows node. To be noted that measurements have shown a different server-side behavior when comparing the same actions executed via PowerShell or from the Windows File Explorer's GUI, as the script triggers more operations on the Samba servers. Yet, it makes a valid reference as it can be reproduced and automated.

4.3 Traffic Analysis

To analyze how operations such as folder listing are served by Samba we have used `wireshark` [8], a popular network protocol analyzer able to parse the SMB protocol payload. The Linux `smbclient` tool was also profiled for comparison. The analysis of the network layer was essential to understand and pinpoint the areas where the underlying software stack is exercised the most.

On the other hand, we used the `strace` [9] tool against the `smbd` process that serves the benchmark test: this tool provides insights about the system calls executed by Samba, and the latency introduced by EOS when serving each system call, thus achieving a fine grained profiling of the EOS FUSE mount implementation under a controlled environment.

4.4 Analysis Results

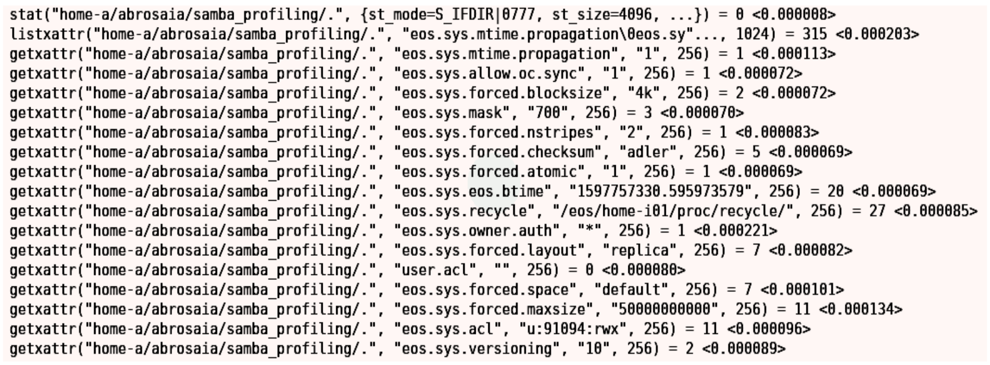
On a general note, the tests against a Windows share on a local file server have shown that the DFS Namespace implementation introduces a negligible overhead. At the same time, while it is legitimate to compare the Samba targets between themselves, an intrinsic miscalibration issue exists in comparing Samba and DFS, given that the hardware/OS stack where they run is significantly different. Nevertheless, the approach has been to compare the existing DFS infrastructure against the deployed Samba infrastructure, as this ultimately is what the users need to access.

One of the most striking evidences when analyzing the `strace` profiles was the large amount of system calls related to extended attributes (xattrs): as a matter of fact, when a Windows client requests the properties of a file, a Samba server is requested to retrieve the list of all its xattrs and their values. If a folder is listed or browsed, the same applies for each xattr of each file in the folder. We note that the same operation executed by the Linux `smbclient` tool does not require all this extra information. The EOS storage system extensively uses xattrs for internal metadata information and bookkeeping, thus making the `getxattr` system call by far the most recurrent call in any of our profiling reports.

To mitigate this performance bottleneck, we have enabled the possibility to hide the system-level extended attributes in the EOS FUSE mounts, such that only user-specific ones like the ACLs are exposed to Samba. The comparison between Figure 5 and Figure 6 shows the impact of hiding the system-level attributes: in this particular case the optimized trace spans 0.065 msec vs 0.203 msec with the default config, over a factor of 3 faster. Given the large count of such requests, this single action resulted in a remarkable boost in terms of overall latency.

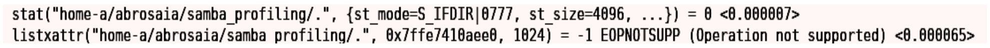
4.5 Further Optimizations

A paramount endeavour that took a considerable amount of time has been the fine tuning of the EOS FUSE implementation, given that EOS is a storage backend serving a wide range of use cases in the HEP community. The most relevant fixes have been introduced in Q4 2020, where handling of internal locks has been optimized, yielding a visible improvement in the service's performance and availability.



```
stat("home-a/abrosaia/samba_profiling/.", {st_mode=S_IFDIR|0777, st_size=4096, ...}) = 0 <0.000008>
listxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.mtime.propagation\0eos.sy"..., 1024) = 315 <0.000203>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.mtime.propagation", "1", 256) = 1 <0.000113>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.allow.oc.sync", "1", 256) = 1 <0.000072>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.forced.blocksize", "4k", 256) = 2 <0.000072>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.mask", "700", 256) = 3 <0.000070>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.forced.nstripes", "2", 256) = 1 <0.000083>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.forced.checksum", "adler", 256) = 5 <0.000069>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.forced.atomic", "1", 256) = 1 <0.000069>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.eos.btime", "1597757330.595973579", 256) = 20 <0.000069>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.recycle", "/eos/home-i01/proc/recycle/", 256) = 27 <0.000085>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.owner.auth", "*", 256) = 1 <0.000221>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.forced.layout", "replica", 256) = 7 <0.000082>
getxattr("home-a/abrosaia/samba_profiling/.", "user.acl", "", 256) = 0 <0.000080>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.forced.space", "default", 256) = 7 <0.000101>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.forced.maxsize", "50000000000", 256) = 11 <0.000134>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.acl", "u:91094:rwx", 256) = 11 <0.000096>
getxattr("home-a/abrosaia/samba_profiling/.", "eos.sys.versioning", "10", 256) = 2 <0.000089>
```

Figure 5. Profile of a *folder properties* request to EOS, with the FUSE mount running with its default configuration.



```
stat("home-a/abrosaia/samba_profiling/.", {st_mode=S_IFDIR|0777, st_size=4096, ...}) = 0 <0.000007>
listxattr("home-a/abrosaia/samba_profiling/.", 0x7ffe7410aee0, 1024) = -1 EOPNOTSUPP (Operation not supported) <0.000065>
```

Figure 6. Profile of the same request, with the FUSE mount hiding the system-level extended attributes.

The *strace*-based analysis has additionally shown that for each directory requested to be listed, its parent directory's content was retrieved as well. This bug did not have a large impact in other FUSE-based setups, but it significantly impacted the Samba nodes as Windows clients list very often their home directory, resulting in multiple listing of each user's parent directories.

Finally, the configuration of the EOS FUSE mounts has been optimized in order to boost both the data and metadata caching, which is particularly effective given the pattern of recurrent requests performed by Windows clients.

To conclude, further fixes are expected in the future: for instance, we need to force a regular remount of all EOS FUSE mounts in order to mitigate internal slowdowns and keep latency to acceptable levels. However, use cases like long-running transfers are naturally impacted by such a workaround. More in general, a continuous improvements process of the EOS FUSE mount implementation is instrumental to the long-term sustainability of the service.

5 Current State and Evolution of the Service

Initially, the CERN Samba service was based on Samba 4.8 on top of CentOS 7, as this was the official release available on CentOS in Q3 2019. In the following months, versions 4.9 and 4.10 were tested, but some performance issues were identified, which prevented their deployment.

With the wide availability of CentOS 8, a new cluster was put in place running Samba 4.11: we have repackaged the Samba software as the upstream repository did not include the latest patches, and we have deployed it in production in Q4 2020. Subsequently, version 4.12 was also deployed, but it had to be eventually reverted because of a software crash observed under a significant load. The latest Samba version 4.13 currently available is expected not to be affected by that bug, as the relevant code has been refactored, therefore we plan to deploy it soon, following further tests. In particular, the RichACL VFS module mentioned earlier is currently being tested and validated.

More in general, we are committed to follow the remarkably active upstream developments and releases, in order to keep our external code such as the RichACL module and the probe up to date, and to ensure the whole service infrastructure is properly maintained and supported.

6 Conclusions and Outlook

In this contribution the Samba-based architecture put in place at CERN was presented, detailing the deployment choices and the field experience in running the service.

We note that new use cases are coming up, thus requiring even more strengthening of this service: in particular, shared project areas hosted in DFS are being migrated, which are more often accessed via SMB. Also, we note that typical experiment workflows including Data Acquisition (DAQ), production, and analysis, have started to be requested from Windows nodes, again requiring native online access to the storage.

At the same time, a number of alternatives are expected to play a role in offering CERNBox native access to Windows users: most notably, an EOS native Windows client is being developed by Comtrade [13], which would allow a seamless integration in the Windows GUI. Furthermore, the new ownCloud client will support the Cloud File System API [12] recently made available in the Windows OS.

We plan to evaluate these alternatives as well as keep evolving the Samba service, in order to offer the most complete set of solutions to our user community.

References

[1] H. G. Labrador et al., EPJ Web of Conferences **245**, *Evolution of the CERNBox platform to support the Malt project*, 08015 (2020)

[2] A.J. Peters, E.A. Sindrilaru, G. Adde, Journal of Physics: Conference Series **664**, 062037 (2015)

[3] A.J. Peters, G. Lo Presti, R. Többicke, *Using the RichACL Standard for Access Control in EOS*, poster at CHEP (2019), https://indico.cern.ch/event/773049/contributions/3474476

[4] G. Lo Presti et al., *CERNBox as the hyper-converged storage space at CERN: integrating DFS use-cases and home directories*, poster at CHEP (2019), https://indico.cern.ch/event/773049/contributions/3474470

[5] G. Lo Presti, *Multi-lock support for Office offline and online applications in CERNBox*, EOS Workshop 2021, https://indico.cern.ch/event/985953/contributions/4238601

[6] Samba, https://www.samba.org

[7] ownCloud, https://owncloud.com

[8] Wireshark network analyzer, https://www.wireshark.org/docs

[9] The Linux syscall tracer, https://strace.io

[10] Microsoft Windows, https://www.microsoft.com/windows, used with permission from Microsoft

[11] Microsoft Distributed File System, https://docs.microsoft.com/en-us/windows-server/storage (access time: 23/02/2021)

[12] Microsoft Windows Cloud API, https://docs.microsoft.com/en-us/windows/win32/cfapi (access time: 23/02/2021)

[13] G. Molan et al., *EOS-wnc: EOS Client for Windows*, CS3 Workshop 2021, https://indico.cern.ch/event/970232/contributions/4158376