A Blueprint for a Contemporary Storage Element, building a new WLCG storage system with widely available hardware and software components: Ceph, XRootD, and Prometheus.

Roger Jones[1,*], *Matt* Doidge[1,**], *Gerard* Hand[1,***], *Peter* Love[1,****], and *Steven* Simpson[1,†]

[1]University of Lancaster, Lancaster, UK

Abstract. When a new long-term storage facility was needed at the Lancaster WLCG Tier-2 Site, an architecture was chosen involving CephFS as a failure-tolerant back-end volume, and load-balanced XRootD as an endpoint exposing the volume via the HTTPS/DAVS protocols increasingly favoured by the WLCG and other users. This allows operations to continue in the face of disc/node failures with minimal management, and enables good utilization of network connectivity for remote access.
We deployed a Prometheus/Loki/Grafana monitoring/alerting stack for timely detection and resolution of failures in such a production environment. Some custom scripts were required to adapt the off-the-shelf functional components with monitoring. With such a monitoring system in place, failures such as disc defects, data corruption and resource exhaustion in long-running processes can be anticipated, and their management planned.
We describe the hardware platform and our requirements on it, and detail the software architecture from initial design, through adaptations to face challenges encountered during production, to present condition. Developments and contributions to related projects that help to fully exploit our design decisions are described. We include performance metrics of the system, the lessons learned during production, and our future plans.

1 Introduction

The WLCG Tier-2 Site "UKI-NORTHGRID-LANCS-HEP" at Lancaster University (UK) required a new solution for the site's storage element (SE), with an opportunity to start with a fresh installation "from the ground up".

It was decided that a modular system, with data storage separate from the technologies used to serve the data, would be optimal. A comprehensive monitoring infrastructure is desired, both to provide alerting to problems and allow measurement of performance. All components are desired to be from standard industry or community sources, i.e., "off the shelf".

[*]e-mail: roger.jones@cern.ch
[**]e-mail: m.doidge@lancaster.ac.uk
[***]e-mail: g.hand@lancaster.ac.uk
[****]e-mail: p.love@lancaster.ac.uk
[†]e-mail: s.simpson@lancaster.ac.uk

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

The basic requirements were:

- Data Server: Provide at a minimum an HTTPS endpoint, and be capable of authenticated, integrity-checked third-party transfers.
- Data Storage: Allow for node-level failures without data loss, be capable of self-repairing, and provide a mountable POSIX-like interface.
- Monitoring: Record real-time metrics for all components, and alert on them.
- All aspects: Cope with the data access requirements for the site, in both the transfers to and from the site and feeding the 8000 job slots that Lancaster provides to WLCG VOs.

2 Data Server - XRootD

Any storage would need to be fronted by a gateway service, providing an authenticated endpoint for transfers to and from the space, both within the site and from external sources. The solution we chose for this data server was XRootD. Our data server was required to have the following minimal functionality:

- HTTPS/DAVS access
- Third Party Copy (TPC) support
- X.509/VOMS authentication
- Flexible scalability
- (For future-proofing) Token authentication support

XRootD [1] fulfilled all these criteria. It is a well known and well established software framework for scalable and fast data access, originating from SLAC. There is a large amount of expertise and experience in deploying, tuning and operating XRootD instances in the UK and the WLCG communities. In addition, XRootD is being adopted by communities and fields beyond High Energy Physics, such as astronomy.

2.1 Redirector Setup

A single XRootD host alone cannot provide the throughput to utilise the available site bandwidth of 40Gb/s. Transfers can be resource-intensive, and the post-transfer integrity check particularly requires a considerable amount of RAM. This was confirmed in a test involving the rapid transfer of ATLAS data to the site. [2]

In order to overcome this, we deployed XRootD[1] in a redirector/cluster fashion. At the time of writing this was 4 nodes—but as noted in the HEPiX study, the improvements to throughput performance were almost linear in proportion to the number of deployed nodes, so it would be possible to scale up until the bottleneck was moved to the site network link.

2.2 XRootD/CephFS Integration

XRootD can be configured to use a POSIX filesystem, and CephFS presents one, so integration is mostly trivial. However, access control had to be carefully considered, and we optimized some data flow thanks to a feature enhancement of an external file-management tool (Rucio).

[1] https://github.com/mdoidge/lancsxroot

2.2.1 Authentication and ACLs

The XRootD services all write and read files under the `xrootd` user. Access control for XRootD or HTTPS/DAVS is controlled through the XRootD AuthDB.

For users accessing through the CephFS mount, this posed additional challenges. One was how to prevent unrecorded writes, or even more worrying, deletions. The best solution to this problem was straight-forward—make the CephFS mount across the worker nodes (WNs) read-only. To prevent users reading from other directories, we turned to extended ACLs, to provide group-level control over reading of directory trees.

This posed one more problem—DAV directory creation (`MKCOL`) does not allow for setting ACLs or permissions, and directories created via DAV did not inherit from their parent directories. The solution to this final problem was to use the XRootD `ofs.notify` functionality to call a "fix permissions" script on each `mkdir` event. In future releases of XRootD, there are plans to switch to a multi-user capacity, and we look forward to integrating any developments in this area.

2.2.2 Rucio Modifications

Most of the data at Lancaster is for ingesting by ATLAS jobs running on our WNs. During the initial deployment, these jobs just accessed their files over the internal XRootD service. However, as the files were available through POSIX access on the WNs' CephFS mounts, this was incredibly inefficient!

In response, the UK Atlas Cloud Manager[2] (re-)implemented a Rucio module to simply symlink to the file's location within the CephFS mount, rather than copy the file locally. This greatly reduces the internal traffic accessing our XRootD servers (shown in Figure 1, where internal reads (yellow area) are eliminated, starting at about 10:00 on 2023-03-13).

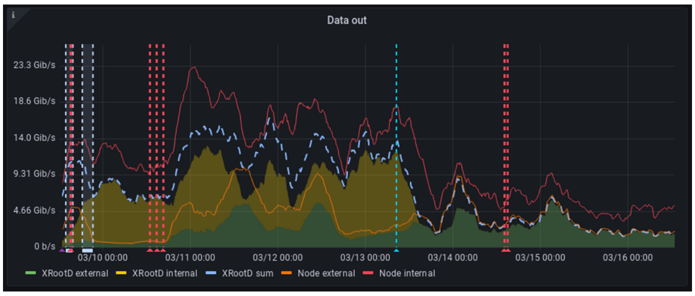

Figure 1. Rucio symlinking eliminating almost all internal reads via XRootD

For future development, native XRootD "redirect to file location" functionality would allow more efficient access for VOs that do not use Rucio.

2.3 XRootD Server Hardware Requirements

The experience gained from our diverse physical nodes (some purchased for the role, others repurposed) suggests that the most important hardware consideration for a high-performance XRootD installation is RAM. Our most performant (least loaded) servers are repurposed ZFS servers with 192GB of memory. Conversely, the redirector service is incredibly lightweight,

[2]James Walder (STFC)

causing very little load in comparison to the data services. We have near-term plans to move the redirector onto our institution's resilient virtualisation platform.

3 Data Storage – Ceph

We chose Ceph[3] as the system to provide our bulk data storage. Ceph is capable of scaling easily without down time. It is designed to run on commodity hardware and mitigates the risk of single point of failure providing a fault-tolerant system. As well as being an object store, Ceph also provides alternative methods for accessing stored data. We have chosen a POSIX-like interface (CephFS) and an S3/AWS interface. CephFS makes integration with existing/new software relatively straight-forward.

3.1 Installation and Configuration

There are several ways to install Ceph. The recommended method is to use the Ceph cluster management tool `cephadm` which is used to install, administrate and orchestrate the cluster. The versions of Ceph[4] that were being maintained were Nautilus, Octopus and Pacific. We chose to go with the newest version, Pacific. Configuration of all the nodes was straight-forward with the use of a few YAML files, and completed without any issues.

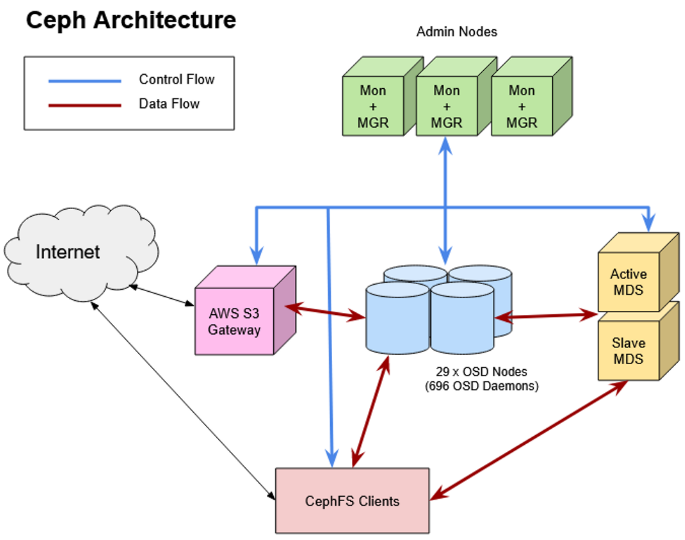

Figure 2. Overview of Ceph cluster

Figure 2 shows our installation, which follows a typical Ceph architecture whilst Figure 3 describes each OSD node in detail.. We are physically limited to four cabinets for the Ceph cluster, which is not ideal. In the event of a rack-level failure, the cluster will switch to a "read-only" state. The admin nodes are in separate cabinets to avoid losing more than one admin node in the event of a cabinet-level failure. An OSD node in each cabinet has been designated as possible a manager+monitor node. In the event of an admin node failure, the orchestration will automatically select an available OSD node and install the monitor and manager services, thus restoring the minimum required number of monitor+manager nodes.

Once the admin node has been restored, the OSD node can be returned to being just an OSD node. We chose to use the minimum of 3 admin nodes to maximise the available storage using the available budget.

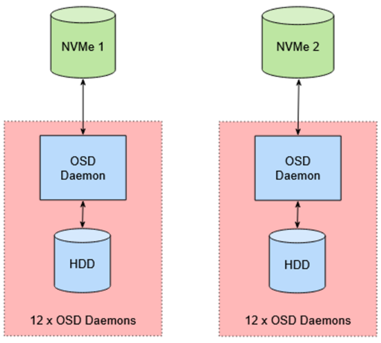

Figure 3. OSD Node Configuration. Each OSD node has 24 HDDs for object storage and 2 NVMe drives for the OSD daemon database (DB) and write-ahead log (WAL). Each NVMe stores the DB/WAL for 12 of the HDDs. The advantage of this approach over mirroring the NVMe drives is a 75% reduction in the data written to each NVMe, significantly reducing NVMe wear. The disadvantage is that a failed NVMe drive will cause the failure of 12 OSD daemons.

3.2 CephFS

The existing middleware (XRootD, ARC-CE, etc) is designed to interact with a POSIX file system. Ceph provides a POSIX-compliant interface via CephFS, which has proved simple to configure and use on the client machines.

CephFS offers two clients, one via FUSE, and the other in the kernel. We have chosen the kernel client as it provides better performance. We were running CentOS 7 on the client machines which only provides Ceph Octopus and lower-version clients. Fortunately, Ceph is designed to allow certain older clients to work with a cluster.

3.3 Administration

Since going live with the Ceph Cluster, we have had a few hard drive failures. We have also needed to apply security updates and restart all the machines in the cluster. Both of these actions were performed with the cluster remaining up and caused no noticeable impact from the users' perspective.

4 Monitoring

The installation is monitored using a Prometheus[5] time-series DB and a Loki[6] log aggregator, populated using various passive and active monitoring agents co-located with storage and gateway components. These include the Prometheus Node Exporter, Loki PromTail, Ceph's in-built exporter, and several custom scripts[3] to work with XRootD's f-streams and summary reporting, and with Ceph's presentation of SMART[4] disc health metrics. Grafana[7] provides the visualization of live and historical data to the operator, while AlertManager manages live delivery of Prometheus-generated alerts via email and potentially other notification systems. These components' interactions are shown in Figure 4.

Challenges were addressed mostly by custom monitoring agents:

[3]https://github.com/lancs-gridpp/gridmon/

[4]"Self-Monitoring, Analysis, and Reporting Technology", intrinsic to most modern storage drives, provides metrics indicating the current device state and enables the predicting of hardware failures.

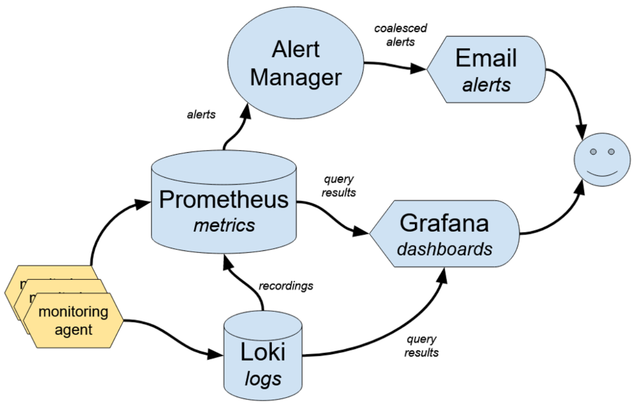

Figure 4. Monitoring and Alerting

- Custom scripts allow acquisition and translation of additional metrics, including interface reachability, some diagnostic Ceph status presently not available through its exported metrics, SMART data (through the Ceph API), and XRootD summaries and detailed events (which gateways push, rather than being scraped/pulled).
- Some scripts push metrics into Prometheus using its remote-write endpoint (which Loki also uses for recordings). These include metrics which themselves are pushed to the script (XRootD) and those which are time-consuming to collect in full (SMART).
- Static expectations are expressed as (rarely changing) metadata metrics, and allow distinction between accidental and intentional absence of metrics. For example, an alert can include the instance name and host of an XRootD instance that has appeared to stop reporting, even though that information normally only appears in the metrics it no longer produces.

Meeting these challenges in turn allows us to detect impending resource exhaustion on XRootD nodes, to detect growing defect lists and write errors of Ceph OSD discs, and to correlate disc failures to physical location, which is useful both for replacement, and for identification of environmental impact (e.g., heat or vibration).

4.1 Detection of File Descriptor Exhaustion

Monitoring helped to spot an irregularly recurring issue with XRootD, whereby it would appear to reach a cap of around 65k file descriptors attached for polling. In the first case, this correlated to a GGUS ticket on many connection timeouts. Restarting the problematic instance mitigated the problem. Monitoring identified which instances manifested the problem, and when, so only faulty instances needed a restart. It also showed that the redirector was unaffected, and that instances running on newer OSes (Rocky8 vs CentOS7) were also immune.

4.2 Monitoring Console

System health is observable through Grafana dashboards. Aspects monitored include overall Ceph health; lists of unused discs/counts of failed OSDs per host; unresponsive hosts, XRootD instances and monitoring agents; host reboots and service restarts (which appear as annotations on graphic visualizations); used and available storage capacity; recent annotations and current alerts; and links to other dashboards and external monitoring. The dashboards simplify identifying problems. For example, Figure 5 shows two OSD hosts, indicating that one of them lacks an OSD due to an absent disc (in the process of being replaced).

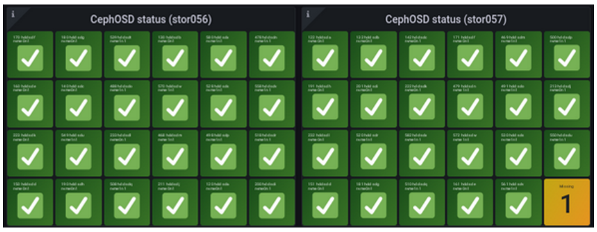

Figure 5. Monitoring showing two Ceph nodes' OSDs, indicating that one is missing

5 Architecture

The installation architecture is shown in Figure 2. A Ceph cluster provides 10PB of object store, presented through CephFS as a POSIX file system to worker nodes and gateways. A cluster of XRootD gateways present remote authenticated access to this file system, and through these Rucio and FTS populate it with transfers from other sites. This serves as preparation for jobs on the worker nodes, which consult Rucio to locate local copies of files. Rucio's structural awareness of the site allows it to point jobs at the local CephFS mount, and have them directly read from the file.

The XRootD gateways also perform administration tasks, such as producing lists of stored files or the Storage Resource Reporting JSON. Jobs with bulk output write back via the local gateways to enforce the more sophisticated access control that XRootD can enforce for writes, and this output can then be delivered under the control of Rucio/FTS to other sites. CephFS also provides a separate small pool, mounted read-write, as a job data scratch area, replacing NFS in our cluster. Gateways and storage are monitored continuously with Prometheus and Loki.

6 Conclusion

Our chosen configuration has provided several benefits for reliability, integrity and efficiency. We have several petabytes of storage tolerant to quickly identified disc failures, with file loss avoided in the short term by redundancy across devices, and in the longer term by on-line hardware replacement. Internal read access is efficient, and inter-site transfers can exploit the available network capacity, while maintaining authorization integrity both internally and externally. Use of CephFS has allowed XRootD gateways and Compute Nodes to continue using the same software. Other problems like disconnections and resource exhaustion can also be detected promptly and examined.

Despite a few hard-drive failures on the OSD nodes and system updates, the cluster has continued to work without needing downtime. Ceph is a complex system and, as a result, it takes extra effort to understand what it is doing and how to configure it correctly. After running the Ceph cluster, it was apparent we could have used slightly different hardware specifications for the Ceph MON/MGR and MDS nodes. The MON/MGR nodes could run with

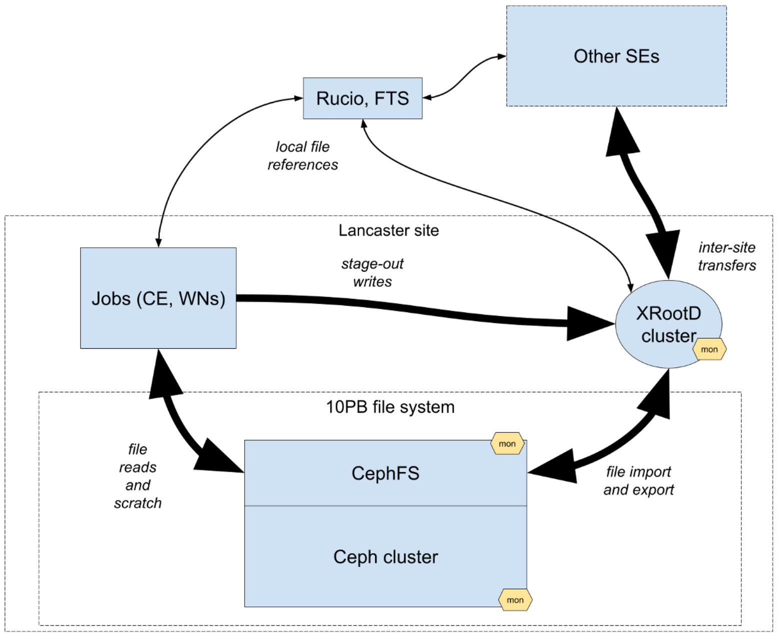

Figure 6. Lancaster Architecture

less processing power and the MDS nodes would have benefited from more RAM. XRootD has provided a scalable and configurable interface for populating and maintaining our data. Monitoring is a crucial aspect for running a healthy Ceph system, and has enabled us to stay ahead of any issues.

After two years of successful running, we can conclude that the architecture fits our purpose well, and provides a solid foundation for our storage operations.

References

[1] (SLAC/CERN, 2023) The XRootD Project. Available at: https://xrootd.slac.stanford.edu (Accessed 18 December 2023)

[2] Doidge M., Walder J. and Rand D (2022) 'Lessons learned from an ad hoc ATLAS data challenge to the Lancaster WLCG Tier2 site.' 3 November. Available at: https://indico.cern.ch/event/1200682/contributions/5106989/attachments/2540334/4372988/AdHocAtlasDataChallange-HEPiX22-MattDoidge.pdf (Accessed: 18 December 2023)

[3] (Ceph Foundation, 2023) Ceph, The future of storage. Available at: https://ceph.io/en/discover/ (Accessed https://ceph.io/en/discover 2023)

[4] Ceph Releases. Available at: https://docs.ceph.com/en/latest/releases/ (Accessed 18 December 2023)

[5] Prometheus Documentation https://prometheus.io/docs/introduction/overview/ (Accessed: 18 December 2023)

[6] Grafana Labs Loki Documentation https://grafana.com/docs/loki/latest/ (Accessed: 18 December 2023)

[7] Grafana Labs Grafana Documentation https://grafana.com/docs/grafana/latest/ (Accessed: 18 December 2023)