Deploying a new realtime XRootD-v5 based monitoring framework for GridPP

Robert Currie[1,*] and *Wenlong* Yuan[1,**]

[1]School of Physics and Astronomy University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Road, Edinburgh, EH9 3FD

Abstract. To optimise the performance of distributed compute, smaller lightweight storage caches are needed which integrate with existing grid computing workflows. A good solution to provide lightweight storage caches is to use an XRootD-proxy cache. To support distributed lightweight XRootD proxy services across GridPP we have developed a centralised monitoring framework. With the v5 release of XRootD it is possible to build a monitoring framework which collects distributed caching metadata broadcast from multiple sites. To provide the best support for these distributed caches we have built a centralised monitoring service for XRootD storage instances within GridPP. This monitoring solution is built upon experiences presented by CMS in setting up a similar service as part of their AAA system. This new framework is designed to provide remote monitoring of the behaviour, performance, and reliability of distributed XRootD services across the UK. Effort has been made to simplify ease of deployment by remote site administrators.

The result of this work is an interactive dashboard system which enables administrators to access real-time metrics on the performance of their lightweight storage systems. This monitoring framework is intended to supplement existing functionality and availability testing metrics by providing detailed information and logging from a site perspective.

1 Introduction

To optimise the amount of compute available to GridPP [1], [2] smaller sites are focussing on maximising their processing power contributions. Providing access to these distributed computing facilities requires these sites to run a lightweight storage cache to provide high overall job efficiency. One of the best solutions for providing distributed lightweight storage services is to deploy an XRootD[3]-proxy cache at each site on top of a POSIX filesystem.

Previous work at Edinburgh [4] has focussed on developing a monitoring system targeted towards XRootD-proxy caches used within the UK at Edinburgh and Birmingham WLCG [5] Tier2 facilities. This system was built by adopting the monitoring approaches used within XCache [6] as part of the SLATE project [7].

With the release of v5 of the XRootD project it is now possible to integrate our monitoring more closely with XRootD improving the experience of sites looking to deploy such a service.

[*] e-mail: rob.currie@ed.ac.uk
[**] e-mail: wenlong.yuan@ed.ac.uk

2 Requirements for the GridPP XRootD monitoring framework

A summary of good requirements for a centralised monitoring framework for distributed lightweight storage solutions have been summarised below:

- **Be simple to deploy/maintain**
  Ideally a remote monitoring framework would require very little configuration/setup on behalf of the site administrator to be deployed at a Tier2 site. Additionally, the service needs to run with minimal interaction from a site admin and should not require additional time/effort to maintain.
- **Centrally monitor multiple instances**
  The end goal of this would be to track and monitor multiple remote services distributed across the UK. This framework has to provide a common monitoring interface for multiple XRootD storage elements.
- **Track the use of storage by XRootD**
  Reporting the usage and free space of the storage behind the XRootD service is an essential part of understanding how well the resources are being used.
- **Track the network use by XRootD**
  As XRootD is used for both hosting and buffering access to files across the network, it's essential to track the network use of the service itself to determine if it's running correctly.
- **Monitor the efficiency of an XRootD cache (if one is present)**
  If the XRootD service is buffering access to remote data via cached storage it would be useful to know how much of an improvement that using this cache is offering.

Previous attempts at collecting enough monitoring data to satisfy the requirements above involved site administrators installing and running additional Python-based services on these XRootD hosts. An effort was also made to provide access to a Docker [8] based solution, however this was not appropriate for all use cases. This main drawback of this approach was that it required additional effort for site administrators and placed additional load on the site storage. Unfortunately this monitoring system has proven to be difficult to scale.

A better solution for monitoring the storage services was found by integrating with the monitoring metrics collected by XRootD itself.

3 Integration with v5 XRootD

Integrating monitoring collection of the lightweight storage endpoints directly with the XRootD service has several advantages. In particular this allows access to the internal state of the XRootD service without having to perform additional inspection of the running storage.

One of the new features of v5 XRootD is the "g-stream" monitoring stream. This allows an XRootD service to report additional monitoring and caching metrics. The advantage of this monitoring is that it is now be possible to use the XRootD service itself to report the metrics which are of interest. One of the big advantages of integrating with XRootD directly is all that is required is to add a few lines to the site configuration as demonstrated in Figure 1.

As the monitoring now integrates with XRootD it allows for XRootD service to be monitored in a generic way. By taking this approach it is also possible to integrate with the XRootD component of the DPM [9] storage systems at sites. This allows for this service to be used as a distributed monitoring solution for many XRootD deployments within GridPP.

```
all.sitename Edinburgh
if exec xrootd:
  xrd.report tatties.ph.ed.ac.uk:9931 every 1m all
  xrootd.monitor all fstat 60s lfn ops tatties.ph.ed.ac.uk:9930
  # g-stream configuration for v5.1
  # xrootd.mongstream all use send json fullhdr tatties.ph.ed.ac.uk:9931
fi
```

Figure 1. XRootD configuration changes which had to be introduced for running services at Edinburgh to report to the new GridPP XRootD monitoring collectors hosted on tatties.ph.ed.ac.uk. This does not represent the entire XRootD-proxy service configuration, only the additional lines which had to be added to collect metrics from the running services.

4 Ingesting monitoring data

The XRootD monitoring streams broadcast data over UDP to a remote server from the storage host. To process this data into a graphical monitoring format it first must be ingested by a collector. A custom collector has been developed which is based on the OSG tool [10] already in production and used by CMS as part of the AAA framework [11]. The new features we have integrated into this collector allow for the collection and parsing of new information exposed by XRootD as part of the "g-stream" metrics exposed in XRootD v5.

Once the monitoring data has been collected it has to be processed and stored before it can be displayed. Building on experiences shared by CMS at building infrastructure like this for production use, we have built and deployed a clustered monitoring system using Docker Compose [12].

An overview of this monitoring system is shown below in Figure 2. The full deployment of this service has been designed around redundancy and scalability. In particular, as the data is broadcast in realtime some effort has been made at the collector and monitoring level to allow data to be buffered to reduce data loss caused by maintenance of individual components. This has proven useful during the development of this cluster as it has allowed individual services to be re-configured and re-deployed as the production service evolves.

Managing various components through Docker also means that the only services which need to be publicly exposed are the metric collectors as well as the Kibana|[13] presentation dashboards.

5 Results

Making use of the monitoring framework which has been deployed for GridPP, it's possible to view live statistics of various sites throughout the UK in much the same way as AAA is able to view live XRootD activities for CMS.

Figures 3 and 4 show data which has been collected in realtime against both the Edinburgh and Birmingham Tier2 storage elements. Figure 3, shows access patterns of files from these different storage servers. This information is similar to metrics made available to CMS through the AAA service. Figure 4, shows how the ratio of file access via normal vs vector reads varies compared to the size of the file. This is not meant to be representative of a single HEP workflow but is the total aggregate of many file accesses against a site. Information such as this is easiest to extract from the XRootD service itself and will be useful in determining how best to optimise these lightweight storage services.

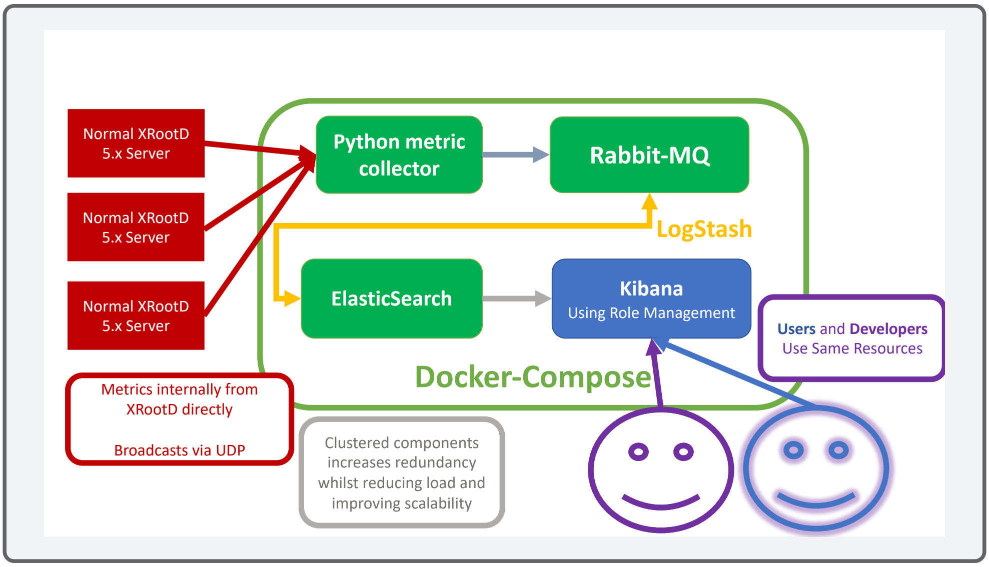


Figure 2. Overview of the service within Docker Compose as setup for monitoring small XRootD storage endpoints. Data from remote sites is first collected via redundant metric collectors which are externally listening on the host. Importantly after the data has been collected it is managed internally through various Docker controlled services until it's presented to the end user via a publicly accessible Kibana instance. This approach reduces the number of services which are externally accessible whilst still keeping all parts of the system resilient to individual component outages.

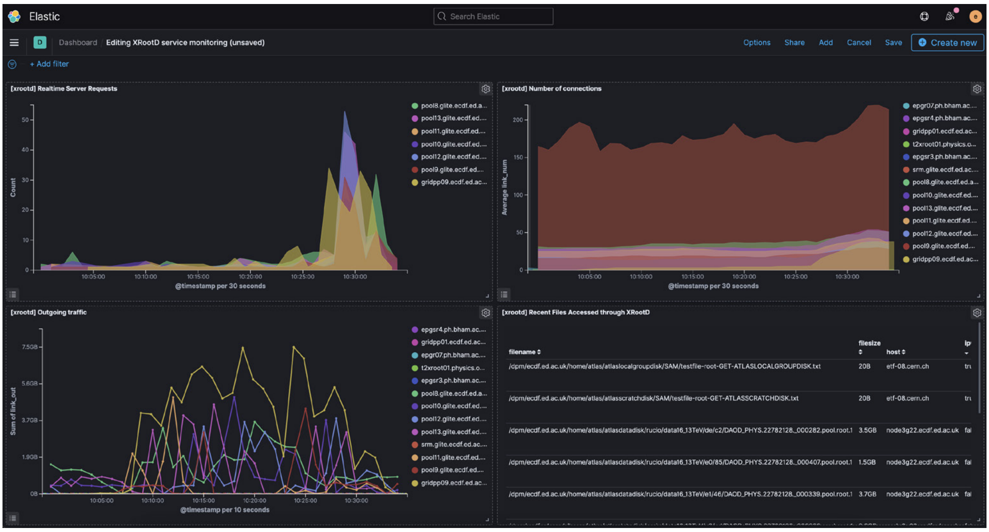
The time between broadcast and presentation of the monitoring data is less than 5 minutes. This allows for this data to be useful in order to support realtime debugging and monitoring of remote lightweight storage services.

6 Summary

We have shown we are able to provide a reliable service for realtime monitoring of distributed XRootD storage. This will allow us to better support many lightweight storage caches distributed across the UK. This new framework has been designed in a way as to allow the various components to scale and be maintained. In addition, we have also designed this solution to be simple to setup for remote site administrators, whilst requiring no additional maintenance overhead.

As well as providing an additional cross-check of the availability and reliability of services running at sites, this monitoring system tracks and logs detailed metrics from the site perspective which is useful in understanding performance of a production service.

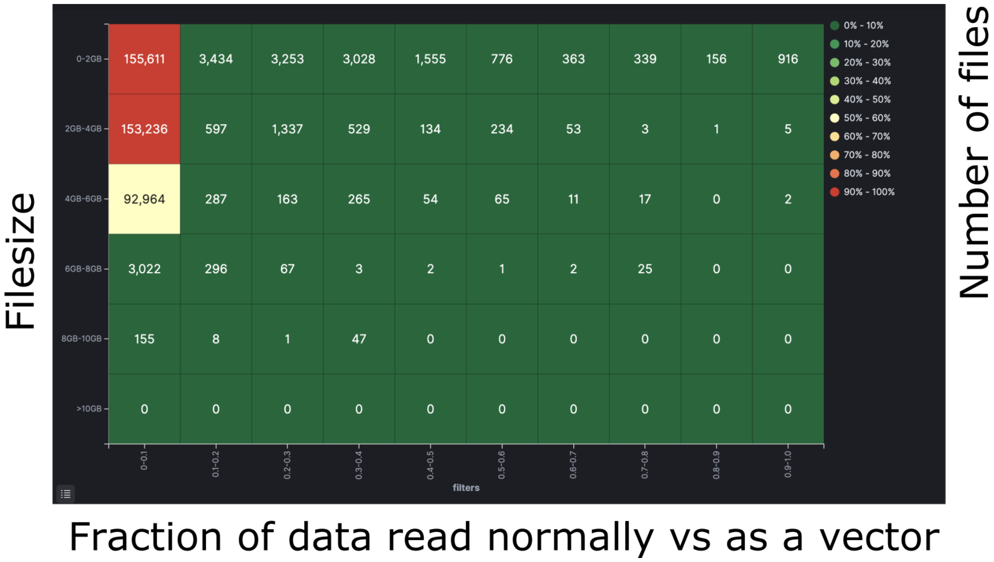
We have also shown that we are able to collect both summary and realtime metrics of the performance of sites which have subscribed to this service. This monitoring data will allow for the configuration of lightweight storage systems to be understood and optimised thereby improving overall site performance.



[xrootd] Recent Files Accessed through XRootD

filename ⇕	filesize ⇕	host ⇕	ip
/dpm/ecdf.ed.ac.uk/home/atlas/atlaslocalgroupdisk/SAM/testfile-root-GET-ATLASLOCALGROUPDISK.txt	20B	etf-08.cern.ch	tru
/dpm/ecdf.ed.ac.uk/home/atlas/atlasscratchdisk/SAM/testfile-root-GET-ATLASSCRATCHDISK.txt	20B	etf-08.cern.ch	tru
/dpm/ecdf.ed.ac.uk/home/atlas/atlasdatadisk/rucio/data16_13TeV/de/c2/DAOD_PHYS.22782128._000282.pool.root.1	3.5GB	node3g22.ecdf.ed.ac.uk	fal
/dpm/ecdf.ed.ac.uk/home/atlas/atlasdatadisk/rucio/data16_13TeV/e0/85/DAOD_PHYS.22782128._000407.pool.root.1	1.5GB	node3g22.ecdf.ed.ac.uk	fal
/dpm/ecdf.ed.ac.uk/home/atlas/atlasdatadisk/rucio/data16_13TeV/e1/46/DAOD_PHYS.22782128._000339.pool.root.1	3.7GB	node3g22.ecdf.ed.ac.uk	fal

Figure 3. This figure shows realtime information on individual transfers whilst providing monitoring of the XRootD servers themselves. These metrics include the number of connections over time for the various services, the amount of data transferred over the network as well as which files were accessed and when.



	0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
0-2GB	155,611	3,434	3,253	3,028	1,555	776	363	339	156	916
2GB-4GB	153,236	597	1,337	529	134	234	53	3	1	5
4GB-6GB	92,964	287	163	265	54	65	11	17	0	2
6GB-8GB	3,022	296	67	3	2	1	2	25	0	0
8GB-10GB	155	8	1	47	0	0	0	0	0	0
>10GB	0	0	0	0	0	0	0	0	0	0

Figure 4. This figure shows the distribution of individual file accesses at the Edinburgh site over a roughly 2 month period. This heatmap shows the distribution of files according to both filesize (GB) and the fraction of reads which were performed using normal read operations or vector reads (0.0-1.0).

References

[1] The GridPP Collaboration, GridPP: development of the UK computing Grid for particle physics, J. Phys. G **32** N1-N20 (2006)

[2] Britton D., et al., GridPP: the UK grid for particle physics, Phil. Trans. R. Soc. A **367** 2447-2457 (2009)

[3] *XRootD*, http://www.xrootd.org (2021), accessed: 2021-02-01

[4] Currie R., Li, T., Washbrook A., EPJ Web of Conferences **214**, 04047 (2019)

[5] *Worldwide LHC Computing Grid (WLCG)*, http://wlcg.web.cern.ch (2021), accessed: 2021-02-01

[6] Hanushevsk A., et al, EPJ Web of Conferences, **214**, 04008 (2019)

[7] Breen, J., et al, Building the SLATE Platform, PEARC '18: Proceedings of the Practice and Experience on Advanced Research Computing, , 1-7 (2018)

[8] *Docker*, https://www.docker.com (2021), accessed: 2021-02-01

[9] Alvarez A., et al, DPM: Future Proof Storage, CHEP 2012, J. Phys.: Conf. Ser. **396** 032015 (2012)

[10] *XRootD monitoring collector*, https://github.com/opensciencegrid/xrootd-monitoring-collector (2021), accessed: 2021-02-01

[11] CMS Xrootd Architecture, https://twiki.cern.ch/twiki/bin/view/Main/CmsXrootdArchitecture (2021), accessed: 2021-02-01

[12] *Docker Compose*, https://docs.docker.com/compose (2021), accessed: 2021-02-01

[13] *Kibana*, https://www.elastic.co/kibana (2021), accessed: 2021-02-01