XRootD and Object Store:   
A new paradigm

Katy Ellis, Chris Brew, George Patargias, Tim Adye, Rob Appleyard, Alastair Dewhurst, Ian Johnson

Presented on 4th November at CHEP 19 Adelaide, in Track 4

Abstract:

The XRootD software framework is essential for data access at WLCG sites. The WLCG community is exploring and expanding XRootD functionality. This presents a particular challenge at the RAL Tier-1 as the Echo storage service is a Ceph based Erasure Coded object store. External access to Echo uses gateway machines which run GridFTP and XRootD servers. This paper will describe how third party copy, WebDav and additional authentication protocols have been added to these XRootD servers. This allows ALICE to use Echo as well as preparing for the eventual phase out of GridFTP.

Local jobs access Echo via XCaches on every worker node. Remote jobs are increasingly accessing data via XRootD on Echo. For CMS jobs this is via their AAA service. For ATLAS, who are consolidating their storage at fewer sites, jobs are increasingly accessing data remotely. This paper describes the coninuting work to optimise both types of data access by testing different caching methods, including remotely configured XCaches (using SLATE) running on the RAL OpenStack cloud infrastructure.

Introduction: RAL, Tier 1, Echo, Erasure coded Ceph Object Store

The Rutherford Appleton Laboratory (RAL) is the UK Tier 1 and supports all LHC experiments, as well as a growing number of others in HEP, Astronomy and Space Science. The RAL disk storage system, known as Echo, is based on an Erasure Coded Ceph Object Store. These terms are described in the next paragraph. Access to Echo is primarily via XRootD, an open source suite of fast and highly scalable data access tools commonly used by LHC experiments, but designed to use with file systems organised in directories [1].

Echo is not a file system in the traditional sense, and does not use a directory structure. Instead it contains ‘objects’, which are named so as to keep the directory name, e.g. /my/path/isActually/anObjectName.root. But in fact the entire ‘path’ is a string which defines the object name.

How Erasure coding works and why it is good

The Erasure Coding software divides every (significantly large) file into 64 MB ‘stripes’, and each stripe into 8 MB ‘shards’. A total of eleven shards are calculated and stored on different disks, and of these, any eight are required to reconstruct a stripe. Hence any three out of eleven disk servers can be

Although there are several advantages to the Object Store, there are also some disadvantages of bringing this new technology into existing experiment analysis models. RAL is unique among WLCG Tier 1s in incorporating this technology, although this is likely to change in the future. The challenge is to optimise access to Echo via XRootD for the different use cases. Several of these are described in this paper.

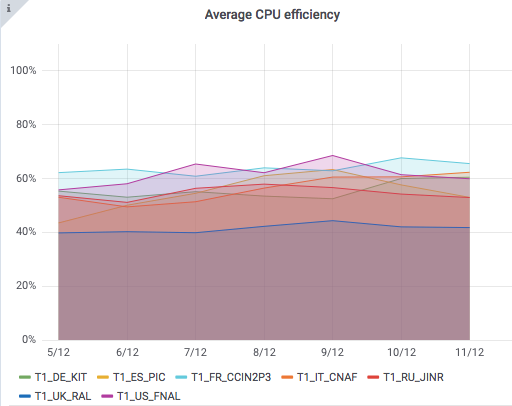
Setup of XRootD at RAL (WNs, GWs, CMS-AAA), cache usage

XRootD is used for both file transfers in and out of RAL, and for jobs running at RAL to access local disk storage. There is also a ‘gridftp’ plugin for transfers between sites, but the intention is to phase this out in the medium-term and replace the functionality with XRootD.

Access to Echo must be done via a ‘gateway’. These can be found on each individual worker node, as well as external facing machines for site-to-site transfers, known as ‘external gateways’. Supported specifically for the CMS experiment is the AAA service, which stands for ‘Any data, Any time, Any where’ and must also run a gateway.

Improving CMS jobs wrt I/O

The main focus of this work is the improvement of CMS job efficiency at RAL, under the assumption that I/O is a significant contributing factor. It is clear that RAL lags behind other Tier 1 sites in this regard, and to see monitoring data like the one in figure X, with RAL 10 – 20 percentage points below the others is typical.



CMS differs in data access methods from the other experiments that run jobs at RAL. Where other experiments will download an entire file before it is needed, CMS keeps an open connection and accesses only the parts of the file needed, when it is required. The advantage of this is that less data overall may be transferred to the worker node; however the nature of an Object Store can make this access pattern inefficient. The best use of an Object Store is to reconstruct entire files, or at least large portions thereof.

TPC, ALICE gateways

The ALICE experiment at the LHC required the development of a unique XRootD authentication plugin. This prevented ALICE moving onto Echo storage until October 2019. Although other experiments share the external Echo gateways, ALICE currently has bespoke access, with a special alias.

The Third Party Copy (TPC) functionality was always (??) present in XRootD, however it was not commonly used due to incompatibilities between storage systems. In 2019 a substantial cross-collaboration effort was made to put it into operation. A TPC transfer copies a file directly from the source to the destination without streaming it via the command issuing machine.

At RAL, some alterations were required to deal with the presence of multiple slashes in filenames. A file system interprets multiple slashes as a single slash, denoting the directory structure. However, an Object Store takes the full string literally, meaning that the two examples in figure X are different object names. In example 2, an additional slash has been added by XRootD.

1. root://xrootd.echo.stfc.ac.uk:1094/dteam:test1/test1
2. root://xrootd.echo.stfc.ac.uk:1094//dteam:test1/test1

Figure 2

The Trivial File Catalogue (TFC) file is normally only used by CMS to redirect files to alternate object names, but both it and the ‘authdb’ file were amended to accommodate surplus slashes for all experiments utilising the TPC functionality.

[1] The XRootD webpage: <https://xrootd.slac.stanford.edu>