XRootD Introduction

Sean Murray

1iThemba LABS

XRootD Intro for SAGrid, 2013
This is no way supposed to be viewed as authoritative on xrootd. This is in no way endorsed by the members of any experiment/root/xrootd. Some of this information might well be entirely false, fabricated, or true.
The first time you have xrootd explained to you, you think wow its nice and simple. When looking deeply, you realise wow its complicated. Thankfully most of the stuff can be ignored, someone with much more experience will handle it. Storage is a non trivial task. XRootD however is very easy to get setup in a simple configuration. As soon as you want to do anything fancy or have a complex hierarchy things get interesting.
Outline

1. Overview
2. features
3. Parts
Why another storage system

Provide a high performance data serving system for hep experiments. High end storage networks (SAN) are not cost effective.
Overview

History

1997  Objectivity, first attempt to use commercial DB for physics data.

2001  BaBar decides to use root framework vs objectivity. Padova/SLAC collaboration

- Design and develop high performance data access system
- Work based on what was learned from Objectivity.

2003  First deployment of XRootD at SLAC.

2013  Wide deployment with several implementations

- ALICE, ATLAS, CMS, EXO, Fermi, LSST.
- Protocol is available in dCACHE, DPM, EOS.
How it works

- request a file from xrootd.tlabs.ac.za (the redirector)
- it responds with, your file resides on xrootd1.tlabs.ac.za
- you then connect to xrootd1.tlabs.ac.za retrieving your file.
- uniform name space, various caching mechanisms, no databases.
- can be geographically distributed across planet as for alice.
- if user requests a file not residing on a node (xrootd1.tlabs.ac.za) it will be sourced and place on there.
Features - server

- A plugin loader, with a default set of plugins that do:
  - Storage aggregation (disks/server/sites) – A single names and entry point.
  - high performance data access through specialised client
  - Smart design, modern protocols, timeouts, near infinite scalability, fault tolerance, . . .
  - no databases, the file systems already know enough about the content.
- Fully plugin based.
- Alone it does basic things, power comes from configurability and adaptability to HEP and HPC requirements.
plugins Architecture

- Protocol Driver
  - XRD
- Protocol (1 of N)
- authorised
  - default, alice
- File system
  - of, sfs, alice
- Clustering
  - cmsd
- Storage Sys
  - oss, dsm, srm
- Authentication
  - gsi, krb, x509
Features - clients

- highly optimized for the data types being stored/copied
- prefetching
- parallel copy streams from parallel sites.
- eg 15 copy streams from russia, germany, italy.
Name space across a site/company/country/federation

Mount points are local, fast, but fragmented, local mount points are hidden and a unique name space is presented. For optimal performance extended file attributes must be used.
Agregate mount points

Mount points can be aggregated on a server as well (oss.cache), called cache file system, but not a cache.

- The true filename is put as a symlink into the LOCALROOT
- The data file (slightly renamed) is put into the appropriate data partition
- The link points to the real data file
Clustering

- cmsd daemons clusterise into a tree shaped network.
- xrootd daemons talk to their cmsd counterparts.
- Redirector: Manager, supervisor, meta-manager
- Data server machine, where the important stuff sits.
- Redirector will cache requests, subsequent requests will not query all back end machines.
- This is a massive subject and requires longer.
Different sites can be clustered and participate to a global name space
They subscribe to a Global Redirector
Each site thinks that it has a tape-like backend
Instead it is not tape, it’s the possibility of fetching files from an external xrootd cluster
This cluster can be external or can be made by the same sites clusterized together
New in 3.3.0

- new client coming, in beta.
- userfriend cli
- third party copy
- integrated checksums at redirector level
- plugin version checking

V4 to be ipv6.
Simple Setup

all.export /tlabsexp
if xrootd.tlabs.ac.za
all.role manager
else
all.role server
fi
all.manager xrootd.tlabs.ac.za 3121
cms.allow *.tlabs.ac.za
xrootd.fslib /opt/xrootd/prod/lib/libXrdOfs.so
xrootd.port 1094
Parts of xrootd

Most commands take if statements at their end to do things based on dns entries or ip.

- xrd Extended Request Daemon
- acc Access control (i.e., authorization)
- cms Cluster Management Services
- ofs Open File System
- oss Open storage system (i.e., file system implementation)
- sec Security authentication
- xrootd The xrootd protocol implementation.
XRD Extended Request Daemon

- xrd.protocol xproofd:1093 /opt/xproofd/lib/libXProof.so
- xrd.protocol wan xrootd *
- Trace execution :: xrd.report myhost:1234 every 15m all -poll
ACC Access control (i.e., authorization)

- xrootd.fslib Load the shared library implementing the ofs and acc components.
- xrootd.seclib Load the shared library implementing the sec (authentication) component.
- ofs.authlib Load the shared library implementing a special acc component.
- ofs.authorize Enables access control, acc component.
CMS Cluster Management Services

- `cms.allow host x.y.z` :: allow x.y.z to connect.
- `all.role [server | supervisor| manager] if dns.of.server`
OFS Open File System
OSS Open Storage system

- oss.localroot configures the LFN<->PFN translation
- oss.cache specify the mountpoints to aggregate via linking.
SEC Security Authentication

Various mechanisms are available for authentication.

- kerberos
- SSO
- ldap
- pam
- GSSI