

XRootD and ObjectIDs

Alastair Dewhurst



Alastair Dewhurst, 13th February 2019



Motivation

- XRootD has since its conception expected files to be stored in something like a filesystem.
 - Expects files to start with a /
 - In some config files (e.g. authDB) it will treat paths not beginning with a / completely differently.
 - Multiple consecutive / will be collapsed down to a single /
- The main difference is that XRootD expects a // after the host.
 - In libXrdCeph the pool name is after the host and a // would mean the pool is called /atlas rather than just atlas.
- Andy H et al have put a significant effort into accommodating objectIDs.
 - We still keep uncovering edge cases
 - In order to get TPC to work, every site would need to add some extra lines of configuration.



libXrdCeph

Config file syntax

```
ofs . osslib libXrdCeph . so [[ user@ ] pool ]
```

Extended file syntax

```
[[ user@ ] pool : ] path
```

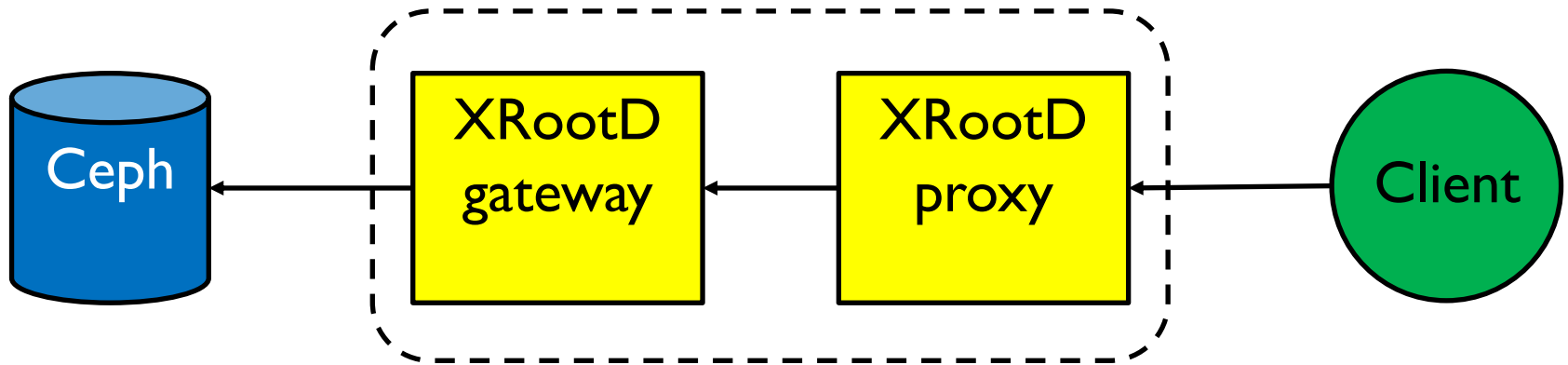
Examples

```
xrdcp root://myserver/mypool:myfile ...  
xrdcp root://myserver/myuser@mypool:myfile ...  
xrdcp root://myserver/:file_with_a:in_it ...
```

<https://indico.cern.ch/event/330212/contributions/1718786/attachments/642384/883834/CephPluginForXroot.pdf>



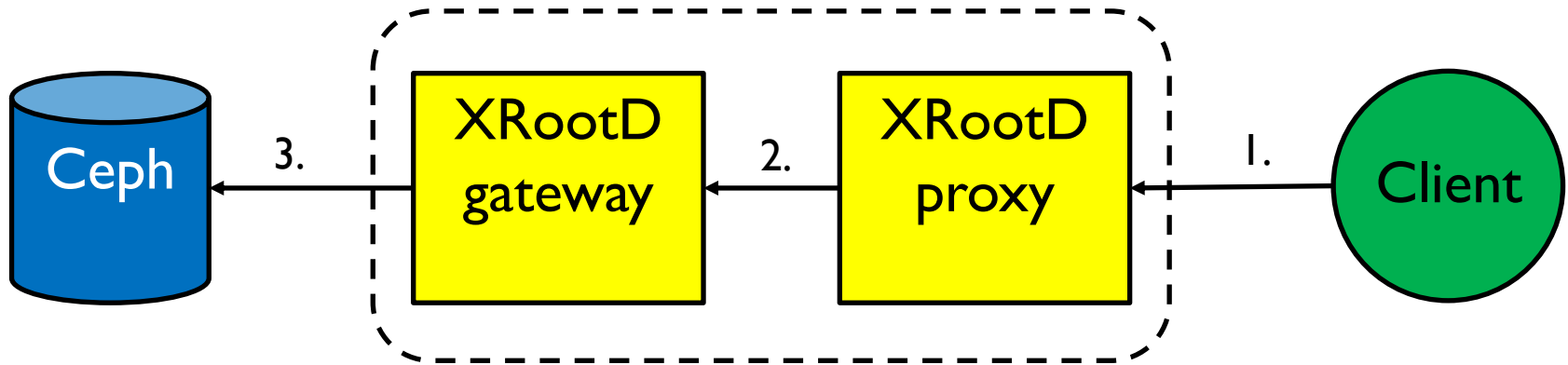
RAL Setup



- XRootD clients talk to Echo via a proxy cache.
 - This can be on the same machine or elsewhere.
- The proxy uses a grid-mapfile + authDB for authorization and authentication.
- The XrootD gateway has all the Ceph configuration in.



// Transfer request



1. `xrdcp root://xrootd.echo.stfc.ac.uk//atlas:scratchdisk/foo/bar /tmp/.`
2. The authDB must be configured to allow this request.
 - Currently it would give permission denied.
3. libXrdCeph must strip out the leading / so it can talk to the correct pool.
 - If there was only ever a single / the plugin must do what it does now.
 - CERN use libXrdCeph on Castor and probably don't want its functionality changed?



AuthDB

#VO = Atlas

u atlasprod \atlas:datadisk/ a \atlas:scratchdisk/ a \atlas:accounting/ r \dteam: a
u atlasuser \atlas:datadisk/ r \atlas:scratchdisk/ a \atlas:accounting/ r \dteam: a

OLD

#VO = CMS

u cmsprod \cms:/store/ a \dteam: a \store/ a
u cmsuser \cms:/store/temp/ a \store/temp/ a \cms:/store/ r \store/ r \dteam: a

#VO = LHCb

u lhcbprod \lhcb:prod/ a \lhcb:user/ a \lhcb:failover/ a \lhcb:buffer/ a \lhcb:accounting/ r \dteam: a
u lhcbuser \lhcb:prod/ r \lhcb:user/ a \lhcb:failover/ a \lhcb:buffer/ r \lhcb:accounting/ r \dteam: a

#VO = Atlas

u atlasprod /atlas:datadisk/ a /atlas:scratchdisk/ a /atlas:accounting/ r /dteam: a
u atlasuser /atlas:datadisk/ r /atlas:scratchdisk/ a /atlas:accounting/ r /dteam: a

NEW

#VO = CMS

u cmsprod /cms:/store/ a /dteam: a /store/ a
u cmsuser /cms:/store/temp/ a /store/temp/ a /cms:/store/ r /store/ r /dteam: a

#VO = LHCb

u lhcbprod /lhcb:prod/ a /lhcb:user/ a /lhcb:failover/ a /lhcb:buffer/ a /lhcb:accounting/ r /dteam: a
u lhcbuser /lhcb:prod/ r /lhcb:user/ a /lhcb:failover/ a /lhcb:buffer/ r /lhcb:accounting/ r /dteam: a



VO changes

- ATLAS can update an entry in AGIS to add an additional / to all their surls
- LHCb can update baseSE path in DIRAC to add an additional / to all their surls.
- CMS use LFN and rely on the Trivial File Catalogue (TFC) to generate the correct surl.



- Could we use the TFC functionality to add or subtract / as required without changing libXrdCeph?
- I don't know enough about how CMS works
- Would this have any advantages over editing the code?

```
<storage-mapping>
<!-- The following is always applied (we specify protocol=xrootd in the xrootd config file) -->
<lfn-to-pfn protocol="xrootd" chain="direct" path-match="(.*)" result="$1"/>
<!-- Below we define the mappings used for each VO as necessary, exiting on the first match -->
<!-- CMS mapping for AAA testing -->
<lfn-to-pfn protocol="direct" path-match="/+store/test/xrootd/TI_UK_RAL/+store/(.*)"
result="cms:/store/$1"/>
<!-- CMS mapping from CMS LFNs to ECHO object names -->
<lfn-to-pfn protocol="direct" path-match="/+store/(.*)" result="cms:/store/$1"/>
</storage-mapping>
```



Conclusions

- It looks like it is relatively straight forward to cope with a // after the host.
 - Pools would not be able to start with a /
- To avoid forking the code base, we should consult with the CERN Castor team.
- Will support for objectID need to persist if we can cope with the extra /?
 - i.e. should we try and remove all single / surls?
- Potentially we could do something with the TFC that could allow test transfers to work?!



Backup



GFAL problems

- There are problems accessing Echo if using XRootD through the gfal wrappers.
- Gfal adds a leading '/' to object IDs.

```
$ xrd fs ceph-gw5.gridpp.rl.ac.uk:1094 stat atlas:scratchdisk/test/rajafile |
Path: atlas:scratchdisk/test/rajafile |
...
$ gfal-stat root://ceph-gw5.gridpp.rl.ac.uk:1094/atlas:scratchdisk/test/rajafile |
gfal-stat error: 13 (Permission denied) - Failed to stat file (Permission denied)
```

- <https://cern.service-now.com/service-portal/view-request.do?n=RQF0789778>
- There is now a flag that will disable this 'relative' to 'absolute' path conversion.



Gateway configuration

12

```
all.export *?  
all.export /  
  
# The adminpath and pidpath variables indicate where the pid and various IPC files should be placed  
all.adminpath /var/spool/xrootd  
all.pidpath /var/run/xrootd  
  
xrootd.async segsize 67108864  
xrd.buffers maxbsz 67108864  
  
# Configure sss security  
xrootd.seclib /usr/lib64/libXrdSec.sosec.protocol sss -s /etc/grid-security/xrootd/sss.keytab.grp -c  
/etc/grid-security/xrootd/sss.keytab.grp  
sec.protbinding * only sss  
  
# Configure rados connection  
ofs.osslib +cksio /usr/lib64/libXrdCeph.so xrootd@,1,8388608,67108864  
ofs.xattrlib /usr/lib64/libXrdCephXAttr.so  
xrootd.chksum Adler32  
  
# Configure the port  
xrd.port 1095
```

