Ceph and XrootD at Glasgow

(and a bit at RAL)

Sam Skipsey

GridPP45 Oct 2020

Structure

- Context
  - RAL/ECHO
  - Development work
- Configuration at Glasgow
- Issues
  - Resolved
  - Ongoing
- Future work

Context

Over the past year, Glasgow has been transitioning its storage from DPM to a lightweight Xrootd layer on top of Ceph.

Some complications have arisen (some due to 2020, some technologically related).

As of this talk, our production storage for ATLAS is xrootd+gridftp / ceph.

We'll cover how we're set up, how we differ from RAL's ECHO, and what we learned so far. (And future developments.)

Xrootd @ RAL “ECHO”

Since 2017, RAL have been transition(ing) from CASTOR to their Xrootd/Ceph based infrastructure: ECHO.

[There’s still some gridftp in their infrastructure too.]

Glasgow transition was planned in 2019 to follow RAL’s lead. Some of our architectural decisions were based on following RAL, but with a T2 bias.

(We erasure code, but with shorter stripes / less parity for lower overheads, for example.)

This year, RAL has also begin some concerted dev effort to support the xroot/ceph oss plugin that underpins both sites’ provisioning.

RAL / xrootd-ceph dev [thanks Ian Johnson]

Enhancing XRootD support for Ceph - 1

- Functional Improvements - completeness
  - XRootD Ceph plugin now supports overwriting of existing files. XRootD hadn't been used for file transfer until ALICE VO started using ECHO (previously, only file access and Direct IO). The RAL bug fix for this is now in latest XRootD v4.12.5
  - RAL and Glasgow are mapping out appropriate support in XRootD Ceph plugin for GFAL2 and other libraries/applications which expect to issue POSIX-type file requests (e.g. FTS). The plugin needs to either reject or pretend to honour some requests, such as mkdir, rename, etc, which aren't supported by Ceph RADOS pools
  - HTTP TPC Support – testing has shown requests are not aligned on RADOS pool stripe width (multiple of N KiB), causing second write in a transfer to fail. STFC looking at a fix for this.

RAL / xrootd-ceph dev [thanks Ian Johnson]

Enhancing XRootD support for Ceph - 2

- Non-functional Improvements - performance
  - Handling Vector Read requests can cause requests to XRootD caches to time-out. STFC are working on handling vector reads directly in the Ceph plugin, which will reduce the load at the cache layer.

Xrootd and xrootd-ceph builds

We build xrootd (including xrootd-ceph) locally to Glasgow.

Historically, because xrootd-ceph needs to be built against the correct ceph release (and originally, we used a different ceph release to RAL)

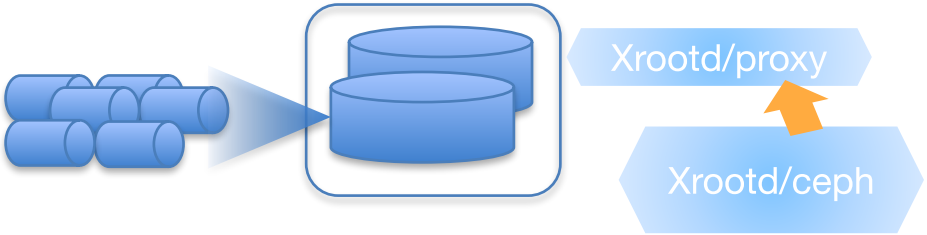
Currently: xrootd 4.12.4 in production

Devel: xrootd 5.0.2 + STFC fork of xrootd-ceph

This isn’t too much work, although xrootd needs newer gcc’s than are in standard CC7, and a lot of support libraries.

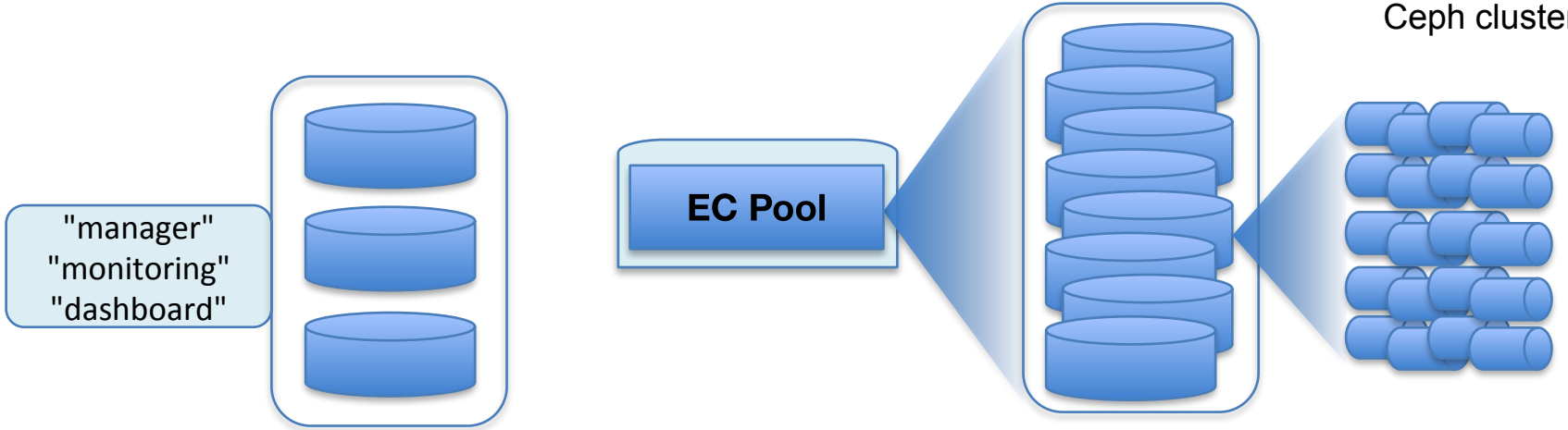
We deploy from tarfiles unpacked into localised locations for efficiency.

Glasgow Ceph/Xrootd Structure



Xrootd server

Ceph cluster



Issues with original model

- Xrootd Proxy Caches are not “writeback”
  - Need two entries in AGIS for ATLAS
    - “Read-only” cache for job input
    - Direct entry for *server* for job output
- Xrootd clustering/redirection problems
  - Xrootd redirectors seem unable to “see” files in an xrootd-ceph instance
  - Redirection always fails as a result.

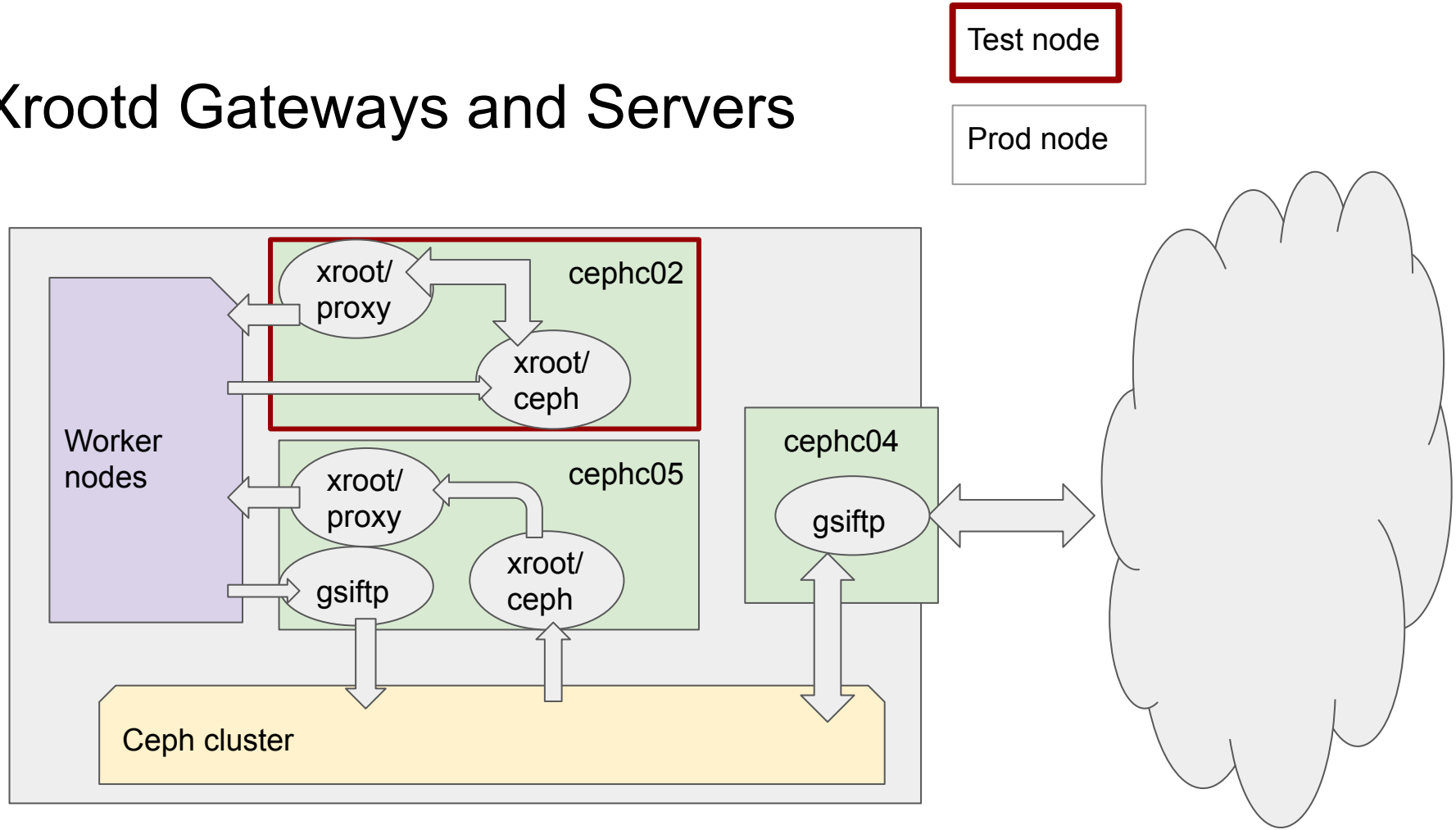
Issues with original model

- Xrootd servers are easily ?overloaded? by job output
  - This isn’t a limitation of interface bandwidth or in the ceph layer
- Issues with provision of full set of functionality required by pilots
  - gfal2 seems to want support for renaming, other posix-only functionality

Resolution / current config

- Caches installed with gridftp as a “write-back” service
- Works perfectly, whilst xrootd cache can still buffer reads.
- I currently suspect that issues are actually related to xrootd-ceph plugin, currently being addressed with dev work.
  - Misaligned writes, feature support, etc

Xrootd Gateways and Servers



Xrootd Proxy Cache “on-disk” config

cephc02
On disk cache is a software RAID of 6 SSDs (in RAID0 config for io performance)
Xrootd Cache sees a single filesystem on a single device.

Xrootd config snippet:
oss.localroot /cache/

Filesystem:
/dev/md127 on /cache type ext4

cephc05
On disk cache is 6 separate filesystems + “redirection” directory (on a RAID1 set).
Xrootd Cache sees all 6+1, and handles management internally (“xrootd spaces”)

Xrootd config snippet:
oss.localroot /redirect
oss.space public /storage*
pfc.spaces public public

Filesystems:
/dev/md127 on /redirect type ext4
/dev/sdc2 on /storage0 type ext4
/dev/sdd2 on /storage1 type ext4
/dev/sde2 on /storage2 type ext4
/dev/sdf2 on /storage3 type ext4
/dev/sdg2 on /storage4 type ext4
/dev/sdh2 on /storage5 type ext4

Xrootd Spaces

Redirection filesystem acts as a global namespace

```
/redirect/
    atlas:datadisk
        [hierarchy of usual Rucio hashes]
            filename          → symlink to file in one of the storage fs
            filename.cinfo    → symlink to metadata in same storage fs
```

storage filesystems are themselves arranged as

/storage0/public/hash/hashed_object_id

spacename

Redirect dir acts as a namespace db for the entire system, so IOPs important on it

Differences to RAL

	Glasgow	RAL	
Cache Topology	Central large proxy caches, on-disk (SSD) ; act as traffic gateways for all access via grid. Each co-lo with xrootd-ceph server	Distributed (1 per WN) small caches, in-memory; each with on-node xrootd-ceph server	Glasgow: Cache efficiency higher (RAL cache mostly sanitises requests) ; but vuln to loss of cache server.
Auth'n/z	LCMAPS plugin mapping VOMS roles to Xrootd "authdb" roles [additional call out to central ARGUS for global banning] (local authn by non-grid mechanisms, to remove need for host certs)	Gridmapfiles mapping proxy DNs to Xrootd "authdb" roles	Glasgow: Mapping VOMSroles->authdb roles more natural; setup more involved (dependency chain bigger) Relies on OSG work for LCMAPs plugins. Not clear how to map to macaroons.

So far...

Ingress gateway easily sustains 10Gbit/s writes, maxes out our share of network.

Internal configuration has scaled, so far, to support as many ATLAS jobs as we've reached so far.

- No sign of load-related issues, as yet.

Ceph itself has been remarkably resilient [and has survived teething problems that would have lost data in our previous system].

Further Work

- Xrootd HTTP-TPC
  - Also testing at RAL
  - Needs xrootd5 [installed on test cache]
- Vector Reads support in Xrootd-Ceph
  - Supporting streaming reads from xrootd service.
  - See work at RAL.
- Fixes to xrootd-ceph plugin to make writes work in xrootd?

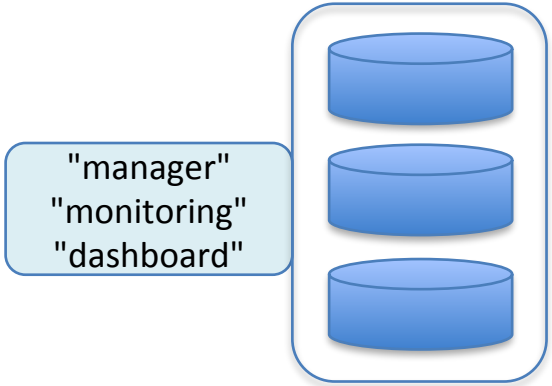
Ceph (Nautilus)

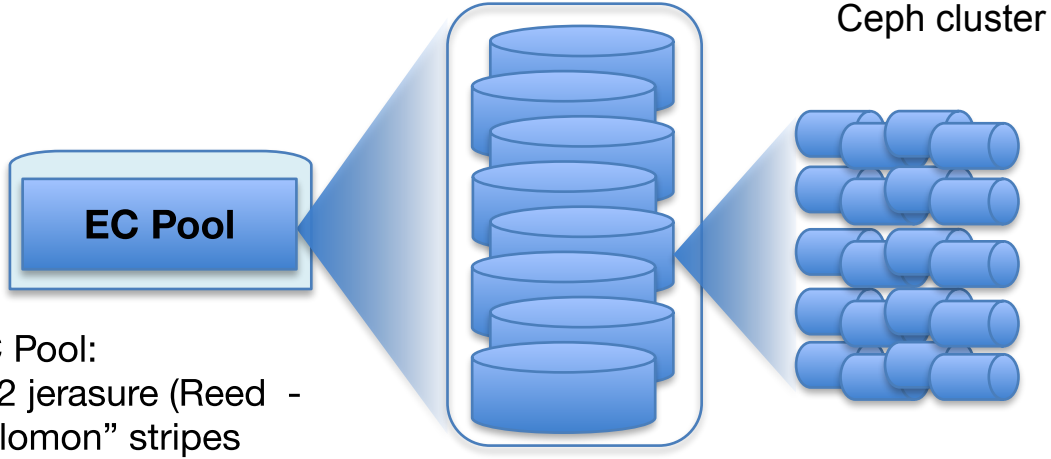
- Moved Luminous -> Nautilus to follow RAL
  - (RAL skipped Mimic, which we'd downgraded from to match them)
- Required new object store for ATLAS to get new features
  - Copy from old to new took ~week whilst ongoing work

Ceph config

cephs01 - 03: management servers
Each runs:
MON (metadata server, PAXOS consensus across all 3)
MGR (rebalancing, metrics, dashboard, “high level” stuff; warm failover across mgrs)

cephd01 - 16 : disk servers.
20 x 10TBdisks per server, each with a separate OSD process adding them to cluster (3.2PB raw)
10Gb/s interfaces





EC Pool:
8+2 jerasure (Reed - Solomon” stripes
Host resilient placement

Ceph (good things)

Highly resilient: zero lost data, even with many OSDs out with on-going transfers.

Highly available.

Lots of monitoring, including prometheus integration.

High bandwidth.

Ansible provisioning tools provided are v good out of box (just slight snagging with upgrading Luminous→Nautilus)

Ceph (Nautilus) snagging

- Irregular OSD dropouts
  - Apparently not associated with load
  - *possibly* associated with “uptime”
  - Several memory management / soft desync issues in issues
    - (None are fatal, but can cause loss of OSD comms)
- Currently solved with automated scripts on OSD nodes to restart “dropped out” OSDs

Ceph future

- OSD nodes currently on single 10GigE network.
- Planned to add second link (channel bonded)
  - Delayed by COVID-19
- We expect this will increase scalability of cluster to writes
- Might also fix/reduce OSD dropouts if timing issue.

- Upgrade to newest Nautilus (end of year). Hopefully also fix OSD dropouts