Optimising XRootD Access to Erasure Coded Object Stores

Andrew Lahiff, Alastair Dewhurst

Echo

Since the start of 2017, the RAL Tier-1's Echo object store has been providing disk storage to the LHC experiments. Echo provides access via both the GridFTP and XRootD protocols. GridFTP is primarily used for WAN transfers between sites while XRootD is used for data analysis.

Access to Echo happens via gateways. Echo has a small number of dedicated gateway machines that provide external connectivity. Jobs running on the RAL batch farm can access Echo through an XRootD gateway running inside a container on every worker node.

Erasure Coding in Ceph

Object stores and those using erasure coding in particular are designed to efficiently serve entire objects (which are normally a few MB in size).

When a file is written to Echo it is broken into objects up to 64MB in size. They objects are slit into 8 x 8MB chunks and an additional 3 parity chunks are calculated.

The 11 chunks are stored on different disk servers which provides excellent data resilience but means an entire 64MB object needs reassembling if a single byte is requested. Some VO jobs using XRootD direct I/O to access their data ran very inefficiently.

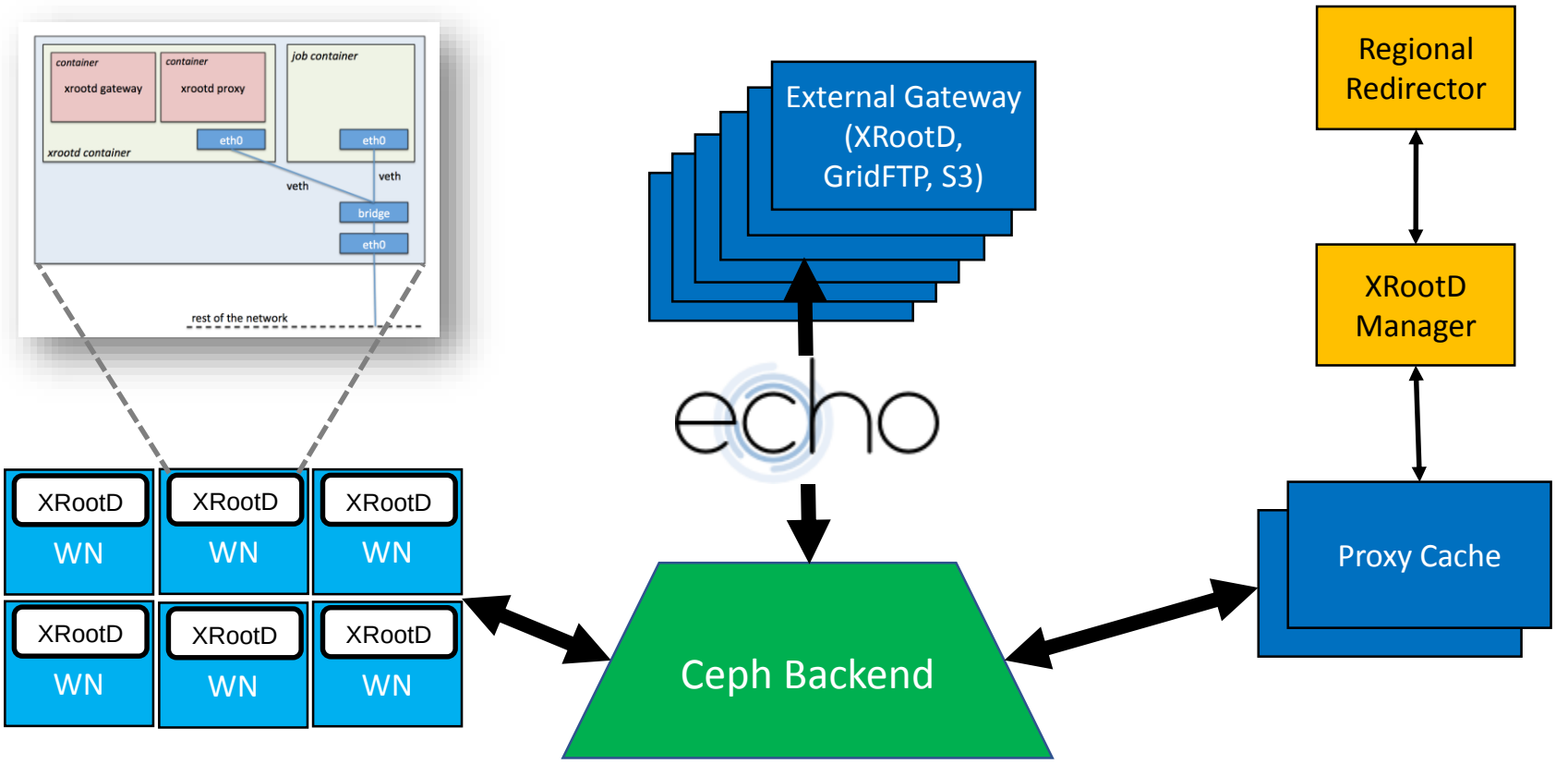
CMS AAA

For CMS, a dedicated XCache service has been provided to allow remote jobs to access data directly from Echo. This is has been built from two old disk servers acting as disk proxy caches and a VM to act as the manager.

A separate service was required because:

1) There is significant CMS specific configuration to enable it to join their AAA service.
2) There are very few tools available to throttle the throughput and protect the service from being overloaded.

These caches pull entire files from Echo.



Gateways on WN

All jobs that run on the RAL batch farm are in containers. Jobs use the same alias to access Echo, however an entry in the '/etc/hosts' file directs transfers to this cache. On every WN there are also containers running an XRootD gateway and proxy cache.

If the file is not cached, the proxy asks the gateway to retrieve the file from Ceph. These two daemons authenticate between each other via a Simple Shared Secret.

Strict cgroups are enabled on the XrootD containers to prevent resource contention with running jobs.

Performance

Two tests were run.

1) To simulate jobs that copy the data completely to the local WN scratch disk 4 GB files were repeatedly copied.
2) CMS jobs from their 'PhaseIIFall16GS82' reprocessing workflow, which is know to depend heavily on I/O, were run.

Both disk and memory caches were tested. A variety of 'Max2Cache' and 'Pagesize' parameters were tested for the memory cache but had little affect.

Time/s	No Cache	Disk (miss)	Disk (hit)	Mem (miss)	Mem (hit)
xrdcp	16.9	32.7	13.0	65.4	45.4
CMS job	536	148	138	187	N/A

Conclusions

Without caching jobs using XRootD direct I/O could run 3 times slower than expected. The addition of caches to every WN has fixed this problem and demonstrates that Object Stores can be work effectively for LHC workflows. Disk caches were more performant than memory caches and used less memory, which is a precious resource on WNs.

Acknowledgements

The authors would like to thank, Andy Hanushevsky, Wei Yang and Brian Bockelman for all the helpful advice and code updates they provided.

Science & Technology Facilities Council

GridPP UK Computing for Particle Physics