Science & Technology Facilities Council

The Echo Project

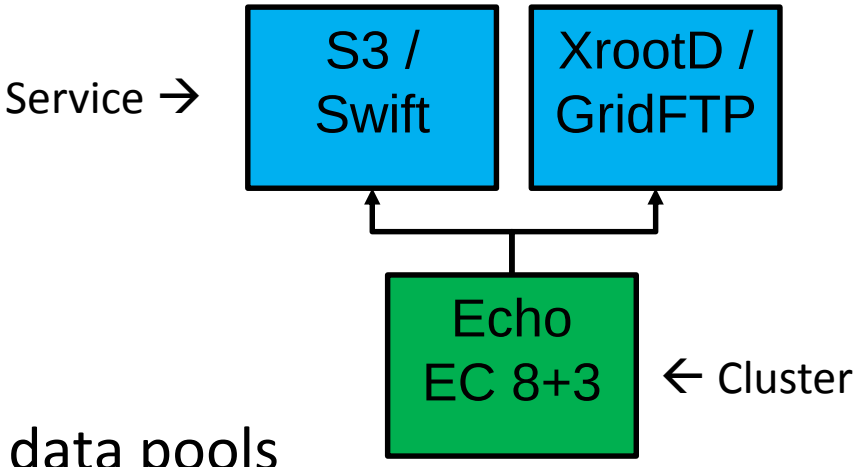
Ian Collier, on behalf of

Tom Byrne, Alastair Dewhurst, Ian Johnson, Alison Packer, George Vasilakakos

HEPiX Fall, KEK, Japan 18th October 2017

echo

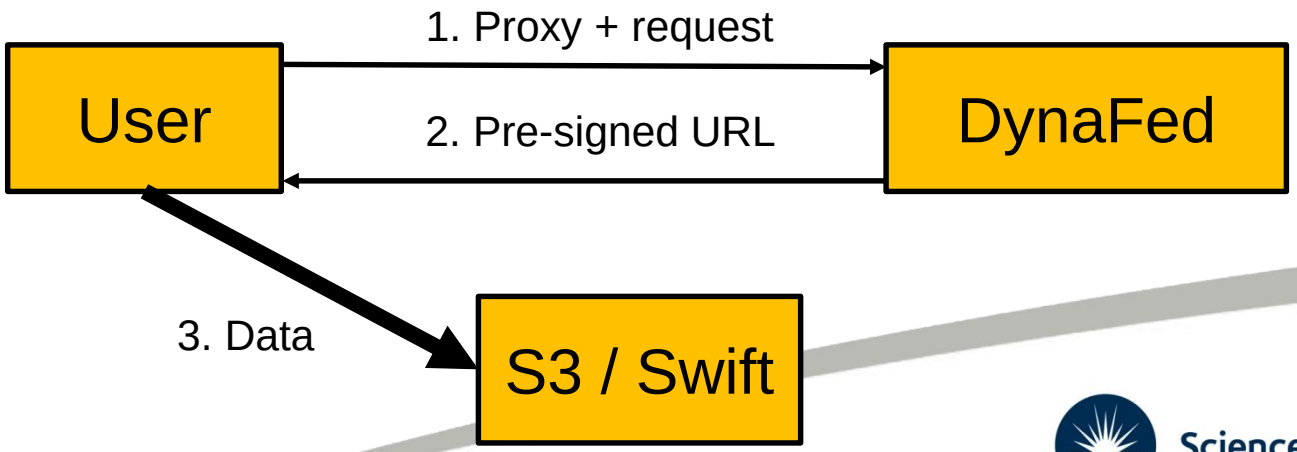
What is Echo?



- Ceph cluster with Erasure Coded data pools
- Disk only storage to replace CASTOR disk for LHC VOs
  - 60 36 disk nodes (5TB disks)
    - initially only 40 nodes used
  - 10PB+ usable space, largest Ceph cluster using EC in production (of which we are aware)
- GridFTP and XRootD have now been in production for ~6 months
- S3 / DynaFed to enter production before the end of the year

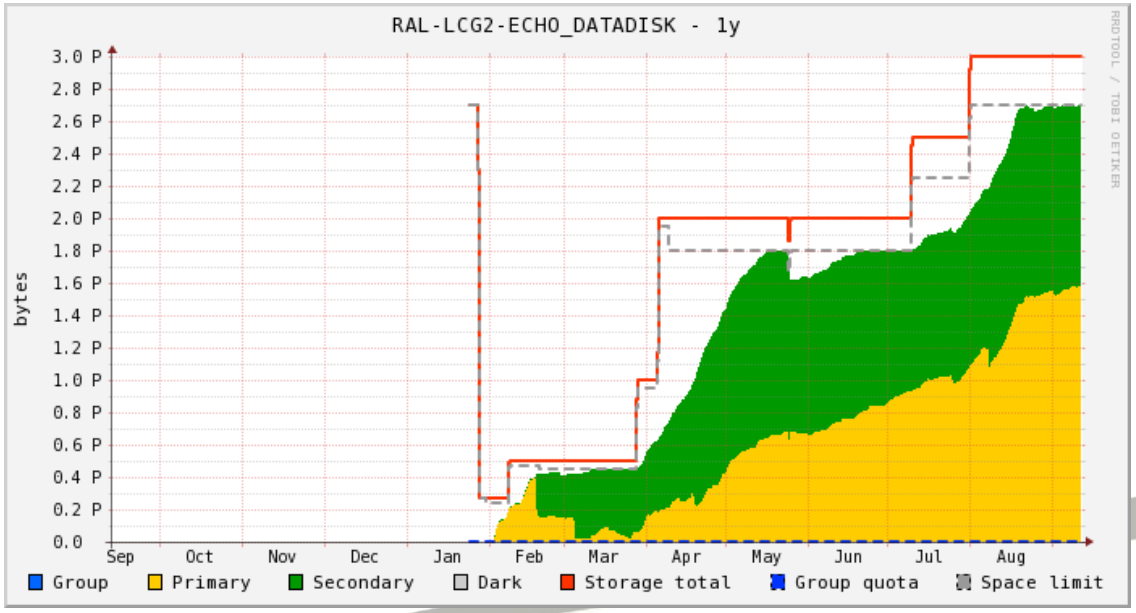
S3 / DynaFed

- S3 API to Ceph is provided by the widely used RADOS Gateway
- DynaFed provides secure access to Echo S3
  - S3/Swift credentials stored on DynaFed Box (VO never needs to see it).
  - Authentication to Dynafed can be anything Apache understands.
    - We are currently using certificate/proxy.
- Provides file system like structure.
- Because DynaFed is developed by CERN, it supports transfers to existing Grid storage.



ATLAS

- Since April, ATLAS have been using Echo in production
  - At the end of July, we reached the intended pledged amount for the year
- It has been a reliable and performant service
  - All the major incidents (discussed later) have affected ATLAS



WAN – GridFTP via FTS
Production jobs – GridFTP
Analysis jobs – XRDCP
with GridFTP writes
(GridFTP failover for reads)

echo

Science & Technology Facilities Council

CMS

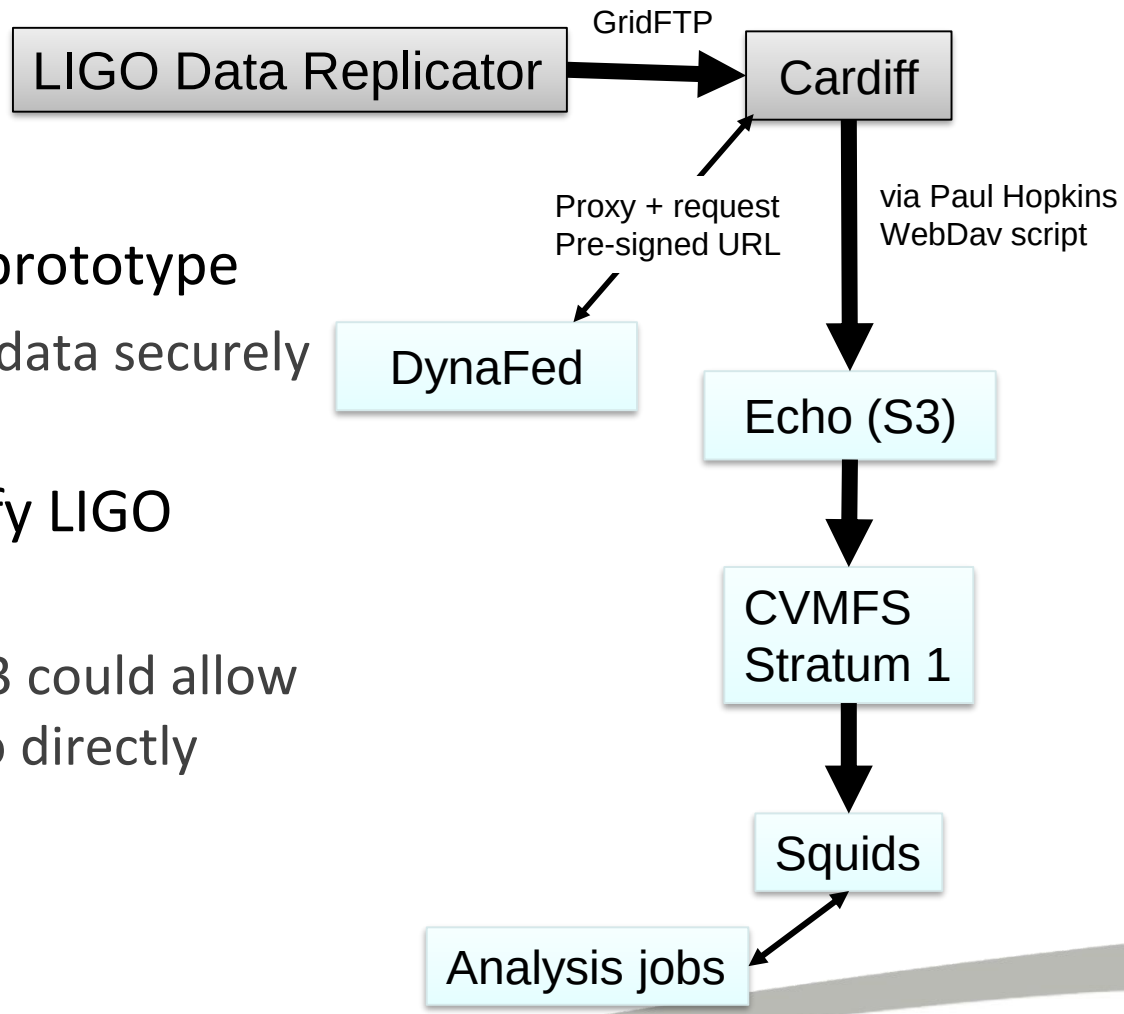
- Extensive testing of all CMS activities has been performed
  - CMS workflow utilizes direct I/O more than Atlas
    - Caching and prefetching needed, this is being accomplished at the XRootD gateway layer
- We have a clear migration plan
  - It was decided to commission Echo as a separate PhEDEx site and this is now complete
  - Separate XrootD proxy cache cluster for AAA (using old CASTOR hardware) is being commissioned

LHCb & Alice

- LHCb have been able to transfer data into Echo via FTS
  - Demonstrated redirection between Echo and CASTOR works
  - Chris Haen is developing gfal plugin for DIRAC to allow LHCb jobs to access Echo
- Work with ALICE started in September
  - Discussions between RAL staff and ALICE storage experts at CERN
  - ALICE use case similar to CMS but with their own XRootD authentication layer

LIGO

- We have a functioning prototype
  - Allows LIGO to access data securely via CVMFS
- Work ongoing to simplify LIGO getting data into Echo
  - New GridFTP DSI to S3 could allow Ligo Data Replicator to directly upload data to Echo



XRootD & GridFTP plugin bugs

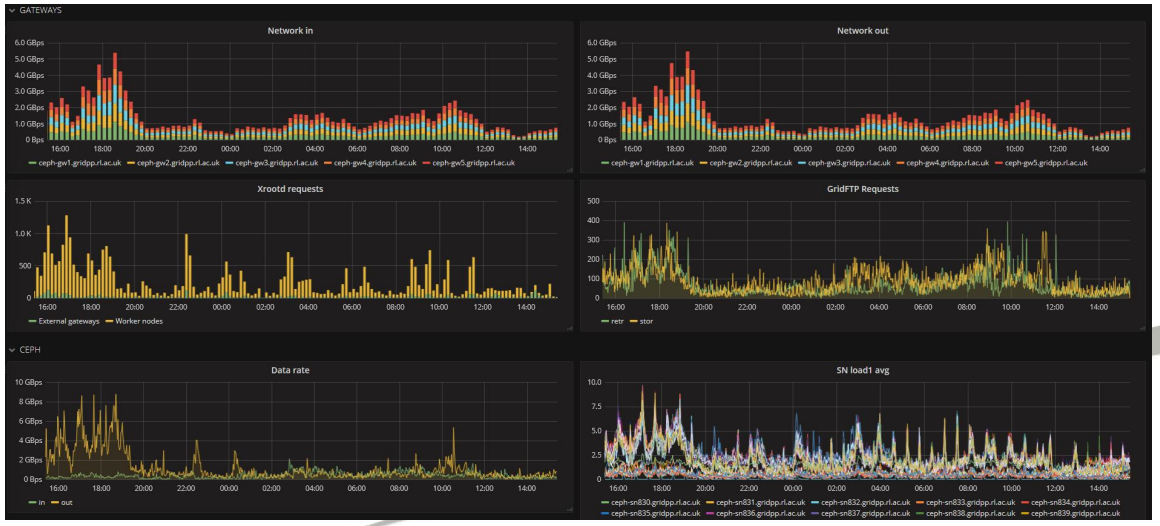
- There have been a few bugs
  - Check-summing didn't work (Fixed).
  - Redirection didn't work (Fixed).
  - Caching proxy didn't work (Fixed).
  - Name-to-Name (N2N) component didn't work (Needed for CMS tests, now fixed).
  - File overwrite doesn't work.
  - Memory usage in both plugins being investigated

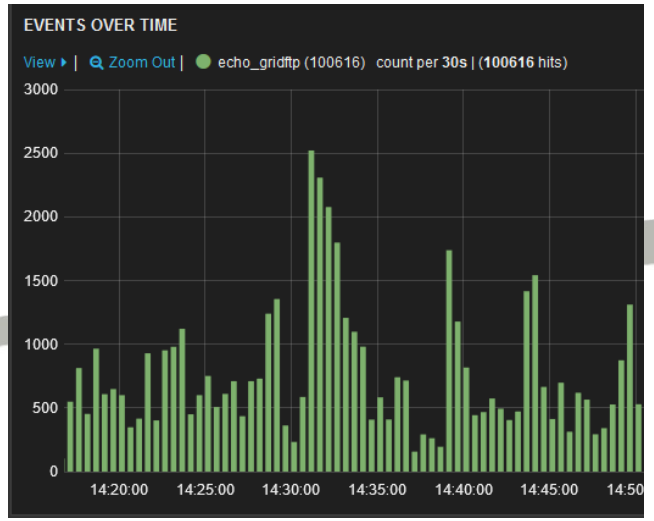
Operational perspective

- Day to day running has been fairly smooth
- We're getting about one callout a week
  - Mainly things that Ceph has handled, but intervention required to manually resolve (inconsistencies found during scrubbing etc.)
  - A lot of work can be done to automate these things, but taking it slow and safe for now
    - Disk removal/replacement is a bit of a time sink currently
- Cluster operations have been mostly transparent to VOs
  - kernel patching
  - placement group increases
  - storage node removal

Issue 1 – Gateway memory usage

- In June the gateway machines started swapping and becoming unresponsive
  - the I/O usage on Echo was high, but not unusual, ATLAS were running ‘normal’ workflows at RAL
  - Determining the specific cause of the extra memory usage was not trivial
    - More specific gateway and transfer metrics collection was put into place





Issue 1 – Gateway memory usage

- XRootD (and GridFTP) Ceph plugin memory usage during reads is proportional to the size of the file being transferred
  - A large number of jobs that were requesting large (~10GB) files started shortly before the GWs started swapping
  - Problem specific to Ceph plugin and the buffer size used
    - Writes are serial and use two buffers max, but reads reassemble all stripes in parallel so the entire object stripes are allocated buffers
    - Tests reducing the buffer size (64MB → 4MB) reduced the memory footprint ~15 times with a 10% reduction in single transfer speed
- Improved monitoring (Kibana and specific gateway Grafana dashboards) has been a huge help in diagnosing operational issues since this issue

Issue 2 – Data loss

Sequence:

- Added the remaining storage nodes from the 2015 generation into Echo in early August, number of OSDs increased to 2160
- An OSD was removed due to read errors
  - This was the primary OSD of a backfilling placement group
- The first 3 OSDs in the set started crashing, flapping and finally died
  - Removing the problem OSD(s) didn't help
- Ultimately, the PG was manually removed and recreated from all OSDs in the set, causing loss of ~23K ATLAS files
  - Only 4000 files were unique, lower than usual for an incident of this type at the T1, as we were able to briefly restore the PG and some high priority files were copied off.

An object storage daemon (OSD) is responsible for storing objects on a local file system, and providing access to them over the network

A placement group is a section of a logical object pool that is mapped onto a set of object storage daemons

Data loss – continued

- This turned out to be caused by a bug in the EC backfilling code
  - http://tracker.ceph.com/issues/18162
- During a backfill, if a read error occurs anywhere in the PG, the primary OSD will crash
  - This is then followed by the next OSD to assume the primary duties crashing in the same manner
  - Eventually enough OSDs crash to render the PG incomplete
- It is likely that further complications caused the data loss to be unavoidable
- There was a separate bug that meant we were seeing an incorrect/unhelpful error message for the backfill bug crash
  - The first day of working on this problem was spent upgrading Ceph to the latest Kraken version, which fixed this bug

Data loss – conclusion

- We have seen ~4% of the new disks start reporting errors
- Currently the gateways' Ceph plugins do not gracefully handle inactive PGs
  - Connections hang and continue to consume resources on gateways
- Subsequent occurrences of the backfilling bug have been resolved without data loss

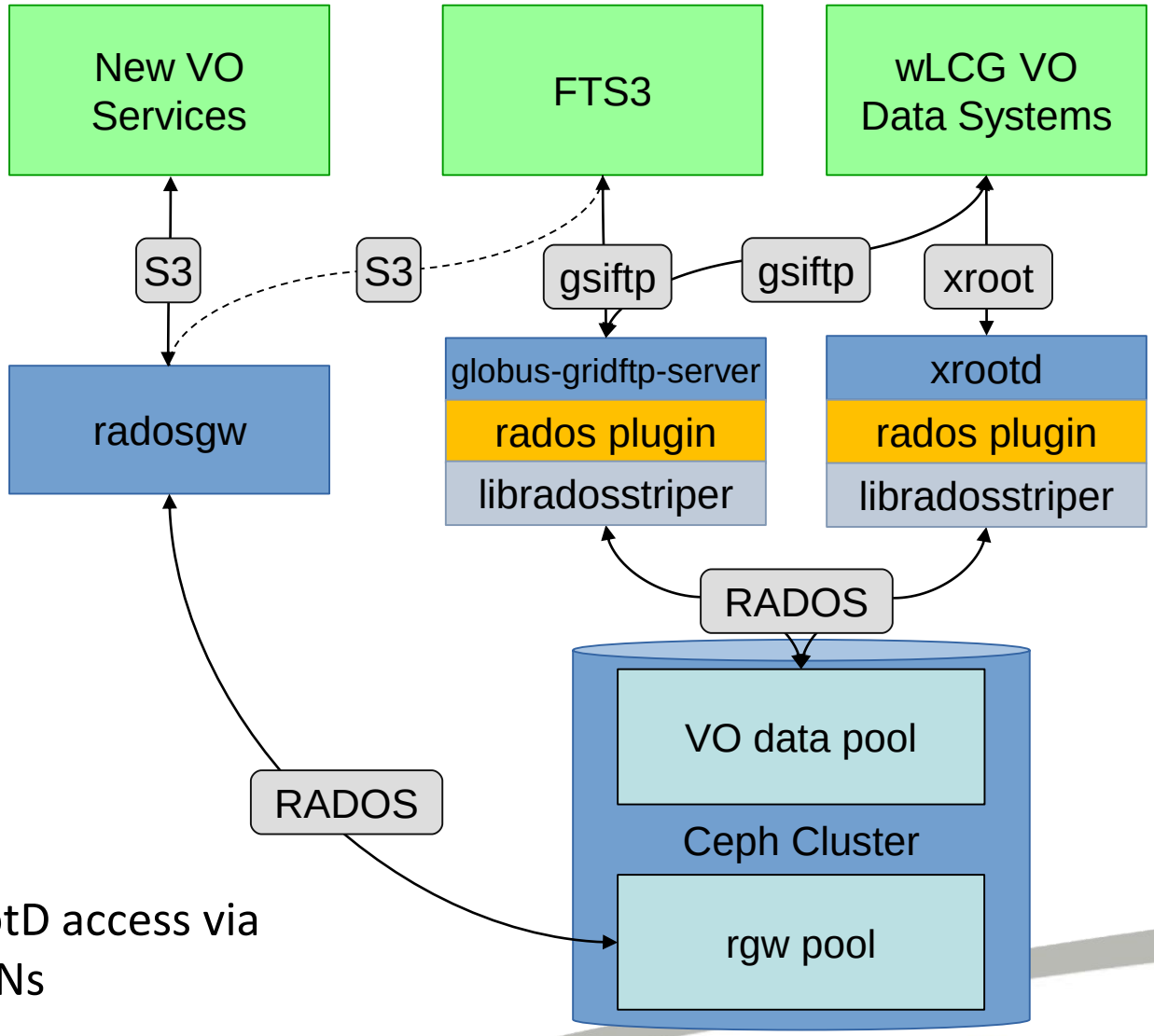
Future plans

- The S3 and DynaFed services will become production Echo offerings before the end of the year
- Aim to upgrade to Luminous in November
  - backfilling bug will be fixed in Luminous
- The 2016 generation of disk servers will be installed with BlueStore OSDs
  - BlueStore is a new OSD backend
    - OSD consumes data disk as a raw block device
    - Substantial performance improvement

Conclusion

- Busy getting VOs and services up and running, and actively looking for new users
- There have been some operational issues
  - One incident was serious, the ATLAS 23K file data loss
  - Every incident has been a learning experience
    - We haven't been bitten by the same thing twice
  - All issues that have occurred were identified in the risk review
- The first 6 months of Echo in production have been encouraging
  - Confidence in Echo is growing

Thank you

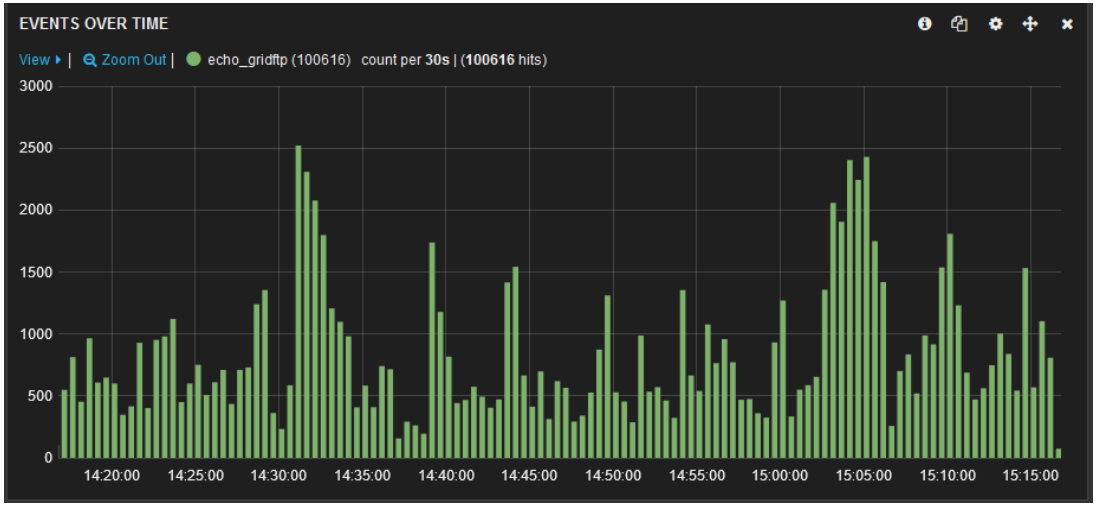


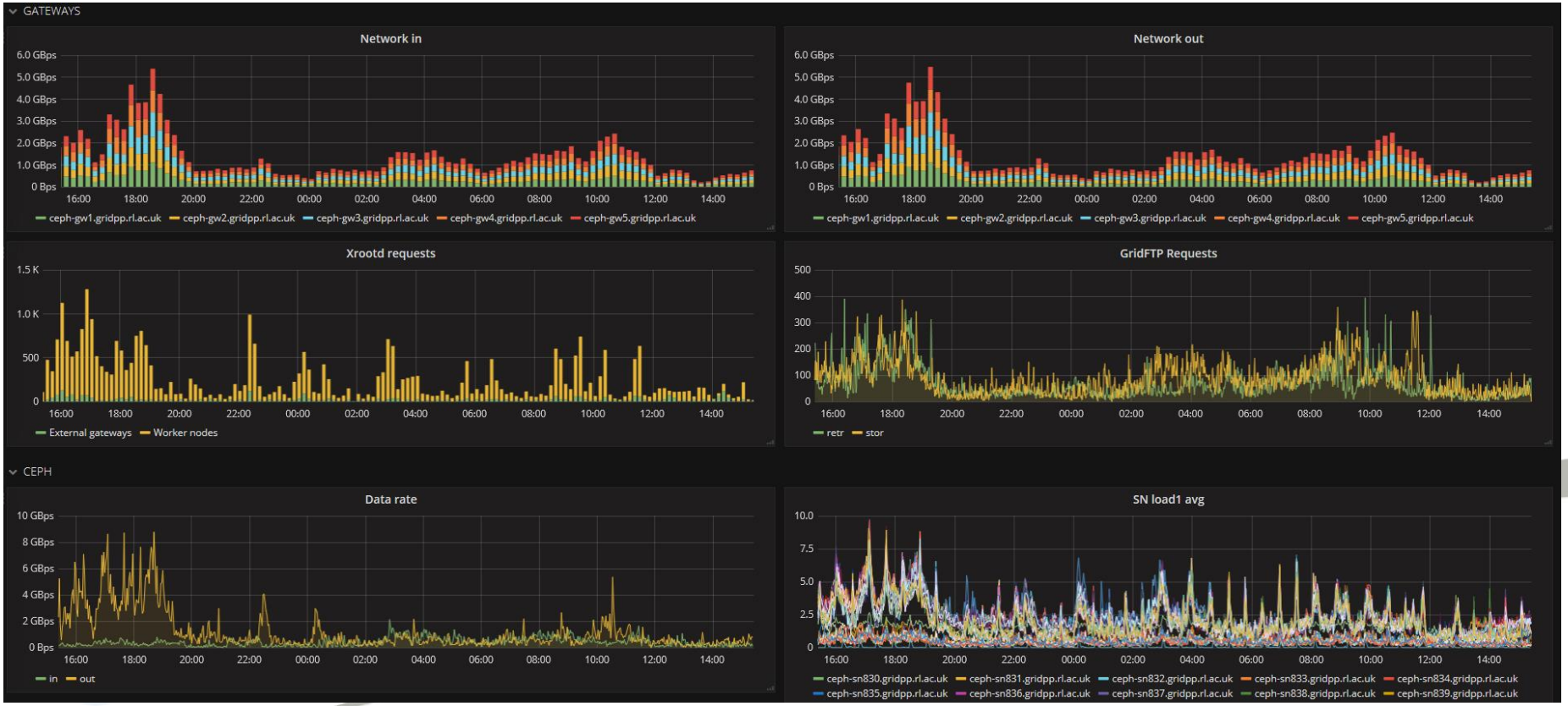
Internal XRootD access via proxies on WNs

Echo hardware

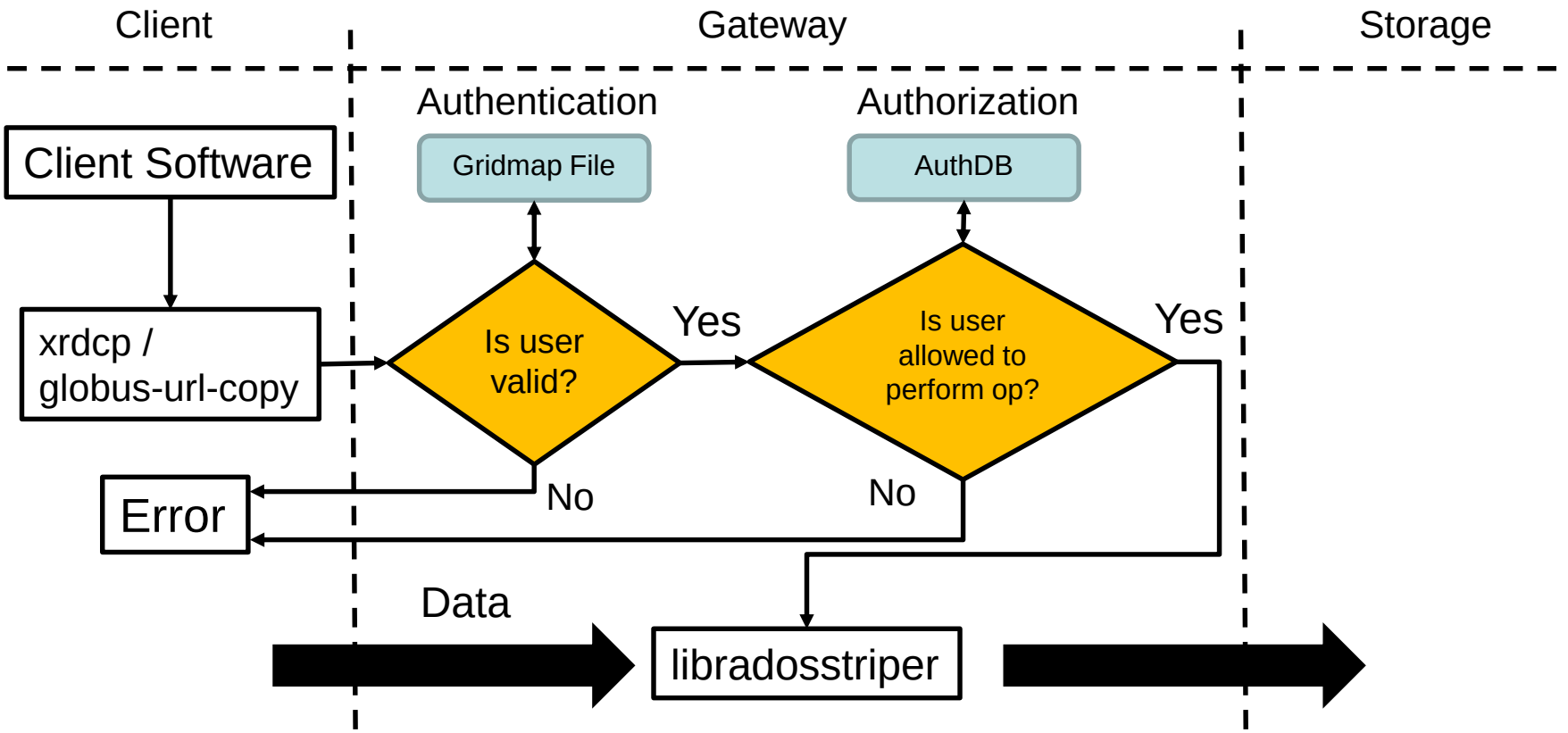
- 60 XMA storage nodes
  - 36 5TB data disks per node
  - 40 nodes used for the initial cluster, the remainder were added in August
- 5 monitor nodes
  - Large SSD's for cluster map storage
- 5 external gateway nodes
  - 128/192GB RAM
  - 4x10G Ethernet

Gateway Monitoring Examples



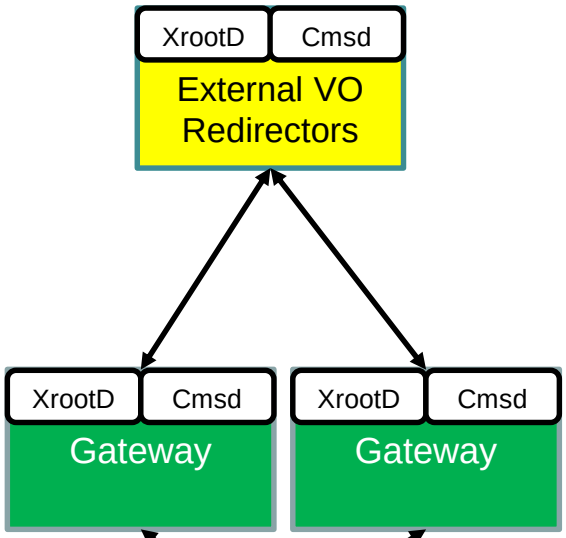


XRootD/GridFTP AuthZ/AuthN



- Gridmap file used for authentication
- Authorisation is done via XRootD's authDB
  - Ian Johnson added support for this in the GridFTP plugin

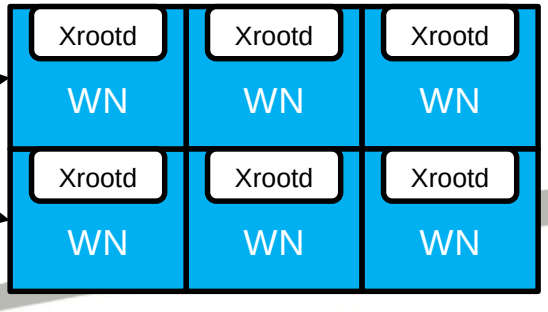
XRootD Architecture



All XRootD gateways will have a caching proxy:

- On External gateways it will be large as AAA from other sites doesn't use LD.
- On WN gateways it will be small and used to protect against pathological jobs.

WN will be installed with an XrootD Gateway.
This will allow direct connection to Echo.
80% of batch farm now using SL7 + containers.



echo

Science & Technology Facilities Council

XRootD Caches

- When even one byte of data is requested from an Erasure Coded object it will have to be completely reassembled.
  - This happens on the primary OSD for the PG the object is in.
- ATLAS Jobs configured to ”copy-to-scratch” whole files.
- CMS jobs need to access data directly from the storage.
  - Tested Lazy-download (which downloads 64MB objects at a time)
  - Can’t use Lazy-download with federated XRootD (AAA) access.
  - Lazy-download appear to add a significant overhead to certain types of jobs.
- Solution to add Caches to the gateways.
  - Currently testing memory cache (as opposed to disk cache).
  - Initial testing (by Andrew Lahiff) looks promising

	AvgEventTime	EventThroughput	TotalJobCPU	TotalJobTime
CASTOR	142.303	0.024478	6490.24	2043.1
CASTOR (with lazy-download) (1)	137.518	0.0253456	6255.06	1975.19
CASTOR (with lazy-download) (2)	133.403	0.0258622	6073.56	1933.73
Echo (remote gateway) (1)	531.954	0.0071882	8390.39	6956.31
Echo (remote gateway) (2)	477.632	0.0079287	7432.37	6306.67
Echo (remote gateway, with lazy-download) (1)	140.139	0.0249962	6044.94	2002.1
Echo (remote gateway, with lazy-download) (2)	134.204	0.0263754	5784.75	1898.26
Echo (local gateway) (1)	560.677	0.00678703	9031.4	7367.29
Echo (local gateway) (2)	482.265	0.00788256	6810.49	6343.49
Echo (local gateway + proxy A)	204.94	0.0175442	7315.44	2850.26
Echo (local gateway + proxy B) (1)	185.221	0.0194803	6463.38	2567.13
Echo (local gateway + proxy B) (2)	185.949	0.0194002	6482.62	2577.73
Echo (local gateway + proxy C) (1)	189.796	0.0188336	6798.95	2655.01
Echo (local gateway + proxy C) (2)	180.042	0.0198329	6238.77	2521.27
Echo (local gateway + proxy D) (1)	171.915	0.0208115	5751.84	2402.68
Echo (local gateway + proxy D) (2)	185.37	0.0193491	6571.65	2584.44
Echo (local gateway + proxy E) (1)	186.836	0.0196245	6541.54	2548.29
Echo (local gateway + proxy E) (2)	184.155	0.0196972	6408.48	2538.67
Echo (local gateway + proxy F) (1)	178.208	0.0200925	5990.55	2488.9
Echo (local gateway + proxy F) (2)	194.985	0.0189344	6938.31	2641.09
Echo (local gateway + proxy G) (1)	176.353	0.0205068	5860.05	2438.7
Echo (local gateway + proxy G) (2)	182.01	0.0197938	6168.35	2526.58
Echo (local gateway + proxy H) (1)	174.227	0.0204519	5696.38	2444.92
Echo (local gateway + proxy H) (2)	175.732	0.0202411	5989.54	2470.37

XRootD caching tests- CMS PhaseIIFall16GS82

Conclusions:

- A proxy gives a significant performance boost
- Proxy parameters such as max2cache & pagesize have a negligible effect
- Lazy-download improves performance more significantly than a proxy

For a single job:

Notes:

- lazy-download is not used unless explicitly specified
- lcg1652.gridpp.rl.ac.uk used for testing - Centos 6 container on SL7
- xrootd daemons running in containers with host networking
- proxy parameters (A): pss.cache debug 3 max2cache 4m pagesize 4m size 1g
- proxy parameters (B): pss.cache max2cache 32m pagesize 64m size 16g
- proxy parameters (C): pss.cache max2cache 32m pagesize 96m size 16g
- proxy parameters (D): pss.cache max2cache 16m pagesize 64m size 16g
- proxy parameters (E): pss.cache max2cache 8m pagesize 64m size 16g
- proxy parameters (F): pss.cache max2cache 32m pagesize 32m size 16g
- proxy parameters (G): pss.cache max2cache 32m pagesize 16m size 16g
- proxy parameters (H): pss.cache max2cache 32m pagesize 128m size 16g

Science & Technology Facilities Council