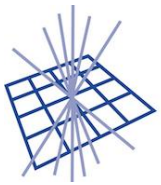
Science & Technology Facilities Council

Ceph @ RAL

Tom Byrne

George Vasilakakos, Bruno Canning, Alastair Dewhurst, Ian Johnson, Alison Packer



GridPP

UK Computing for Particle Physics

Outline

- Recap on Ceph at RAL
- Ongoing developments for the LHC Tier-1 storage system: “Echo”
  - Production status
  - Gateway overview

About me

- Tom Byrne
  - tom.byrne@stfc.ac.uk
- STFC - RAL
  - Scientific Computing Department
  - Data Services
  - Facilities/Tier-1
- Working mainly on Ceph

Science & Technology Facilities Council

Previous talks

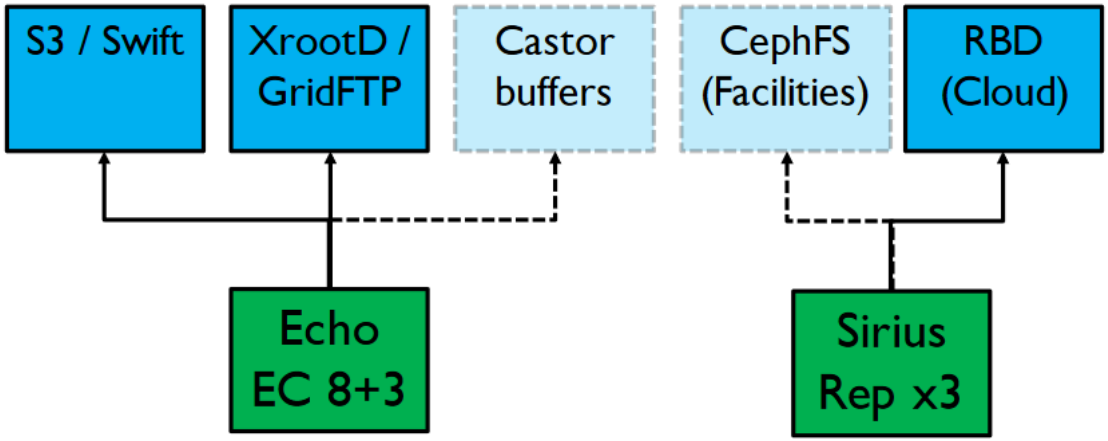
- HEPiX Fall 2016 – Experience of Development and Deployment of a Large-Scale Ceph-Based Data Storage System at RAL
  - https://indico.cern.ch/event/531810/contributions/2298934/
- CHEP 2016 – The deployment of a large scale object store at the RAL Tier 1
  - https://indico.cern.ch/event/505613/contributions/2230932/
- HEPiX Spring 2016 – Why so Sirius? Ceph backed storage at the RAL Tier-1
  - https://indico.cern.ch/event/466991/contributions/2136880/
- HEPiX Fall 2015 – Ceph object storage at RAL
  - https://indico.cern.ch/event/384358/contributions/909231/
- HEPiX Spring 2015 - Ceph storage at RAL
  - https://indico.cern.ch/event/346931/contributions/817835/
- Regular presentations at HEP Ceph meetings:
  - https://indico.cern.ch/category/4866/

Recap!

Science & Technology
Facilities Council

Ceph at RAL is used for two main projects:

- Sirius
  - Provide low-latency block storage to STFC cloud
    - thin storage nodes + replication
  - ~600TB raw disk
- Echo
  - Disk only storage service to replace Castor for LHC VOs
  - Emphasis on TB/£ rather than IOPS/£ (fat SNs + EC)
  - 9.9PB usable space, largest Ceph cluster using EC in production (of which we are aware)



The Echo Project – An Update

Erasure Coded - Ceph - High throughput - Object store

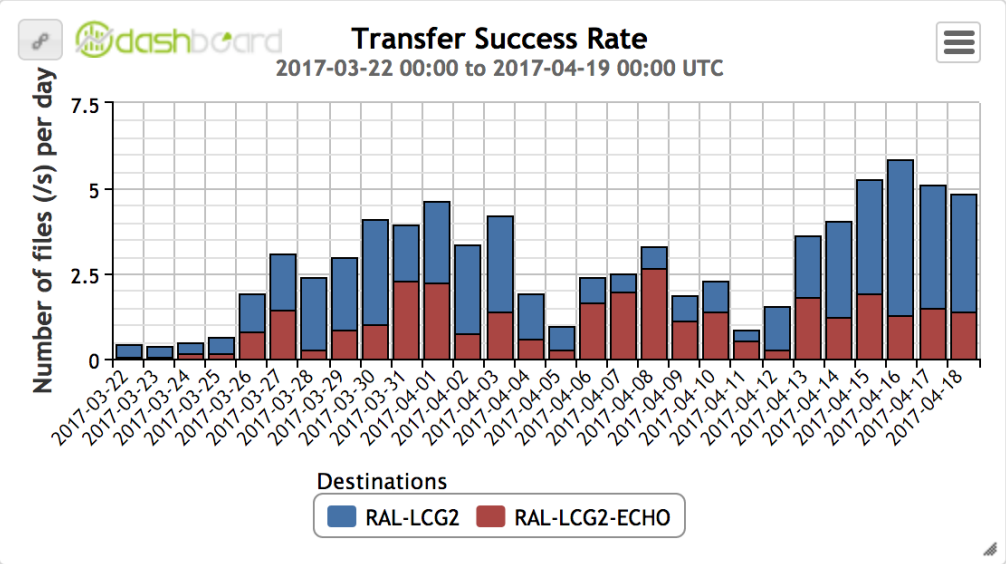
echo

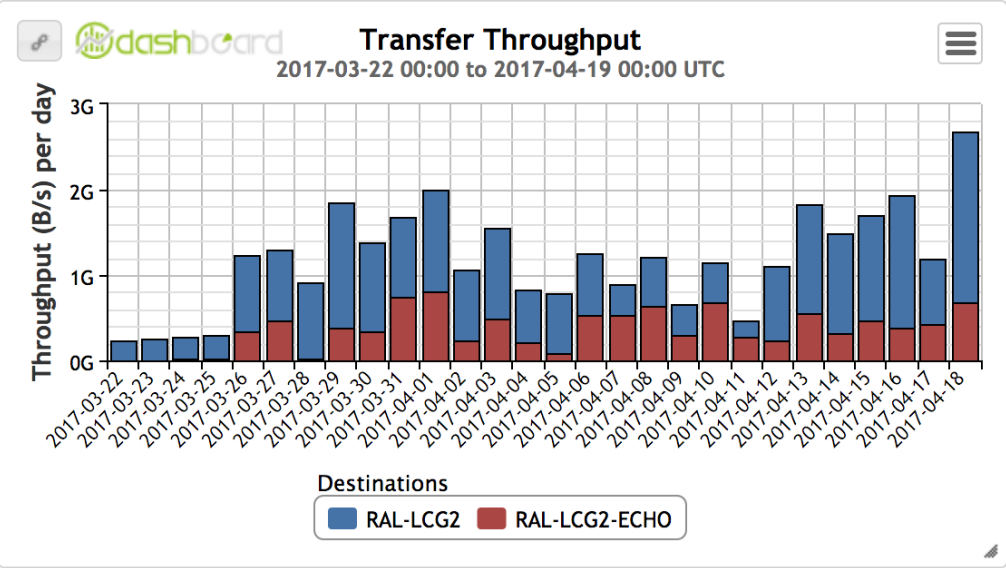
Science & Technology Facilities Council

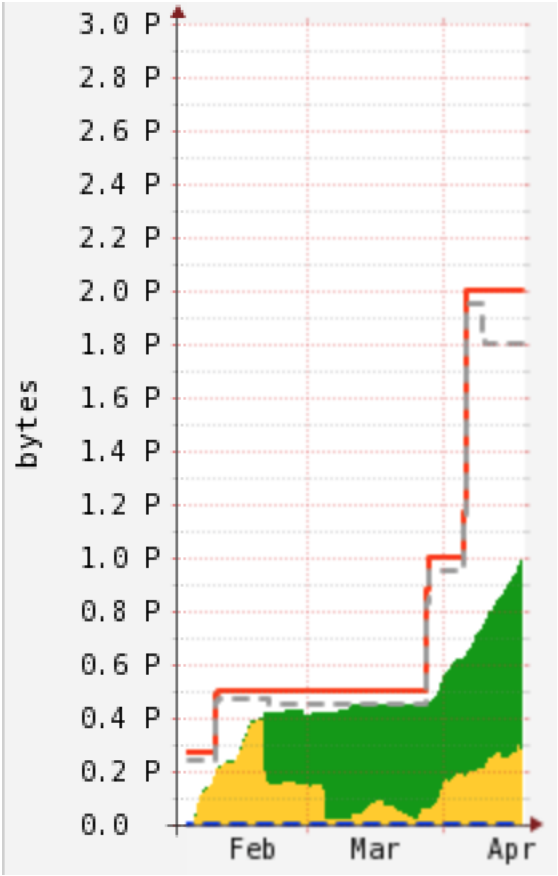
Production

- Echo is now accepting production data from LHC VOs
  - Currently GridFTP and XRootD supported as production I/O protocols.
  - VO pools can be accessed via either protocol.
- 7.1 PB of WLCG pledge provided by Echo this year.
  - Split between ATLAS, CMS and LHCb
- Already storing over a PB of data for ATLAS
- Lots of hard work: testing, development, on call procedures etc
  - The gateway development has been a large part of the work

ATLAS in Production







echo

Science & Technology Facilities Council

Operational issues: Stuck PG

- While rebooting a storage node for patching in February, a placement group in the atlas pool became stuck in a peering state
  - I/O hung for any object in this PG

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

- Typical remedies for fixing peering problems were not effective (restarting/removing primary OSD, restarting set etc.)

```
pg 1.323 is remapped+peering, acting
[2147483647,1391,240,127,937,362,267,320,7,634,716]
```

- Seemed to be a communication issue between two OSDs in the set.

Science & Technology Facilities Council

Stuck PG

- To restore service availability, it was decided we would manually recreate the PG
  - accepting loss of all 2300 files/160GB data in that PG
- Again, typical methods (force_create) failed due to PG failing to peer
- Manually purging the PG from the set was revealing
  - On the OSD that caused the issue, an error was seen
  - A Ceph developer suggested this was a LevelDB corruption on that OSD
  - Reformatting that OSD and manually marking the PG complete caused the PG to become active+clean, and cluster was back to a healthy state

Stuck PG: Conclusions

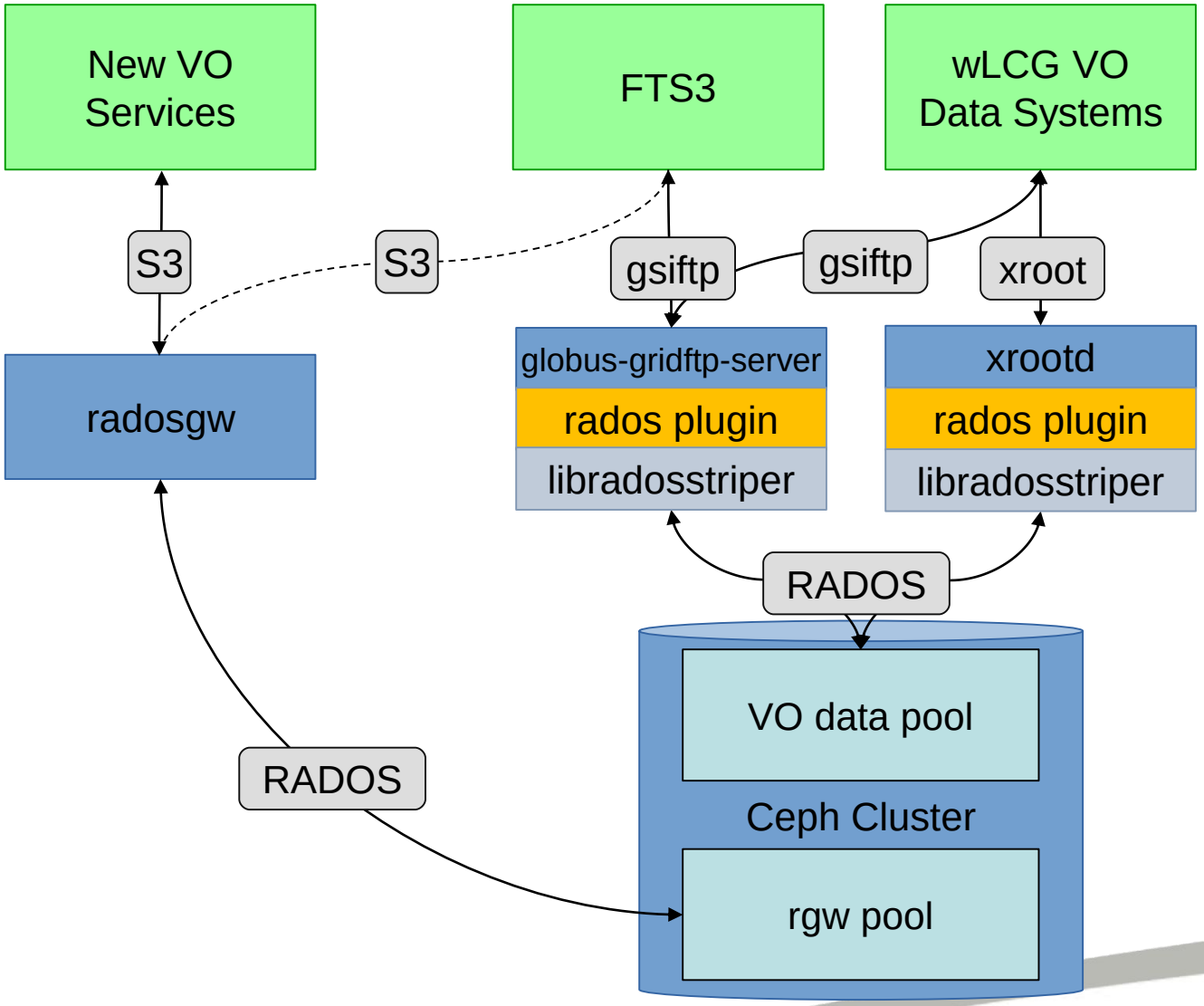
- A major concern with using Ceph for storage has always been recovering from these types of events
  - This showed we had the knowledge and support network to handle events like this
- The data loss occurred due to late discovery of correct remedy
  - We would have been able to recover without data loss if we had identified the problem (and problem OSD) before we manually removed the PG from the set

http://tracker.ceph.com/issues/18960

Echo gateway development

Gateway recap

- The aim was always to have a simple, lightweight solution
  - No CephFS or similar solution (scaling/maturity issues)
  - As close to a simple object store as possible
- Not possible get VOs to use radosgw (S3) from the start
  - Traditional HEP protocols needed
  - Same data needs to be accessible via all protocols
- Need to have control as to how the data ends up in Ceph
  - Future proofing and ability to support other access methods


New VO Services
FTS3
wLCG VO Data Systems
S3
S3
gsiftp
gsiftp
xroot
radosgw
globus-gridftp-server
rados plugin
libradosstriper
xrootd
rados plugin
libradosstriper
RADOS
RADOS
VO data pool
Ceph Cluster
rgw pool

XRootD Plugin

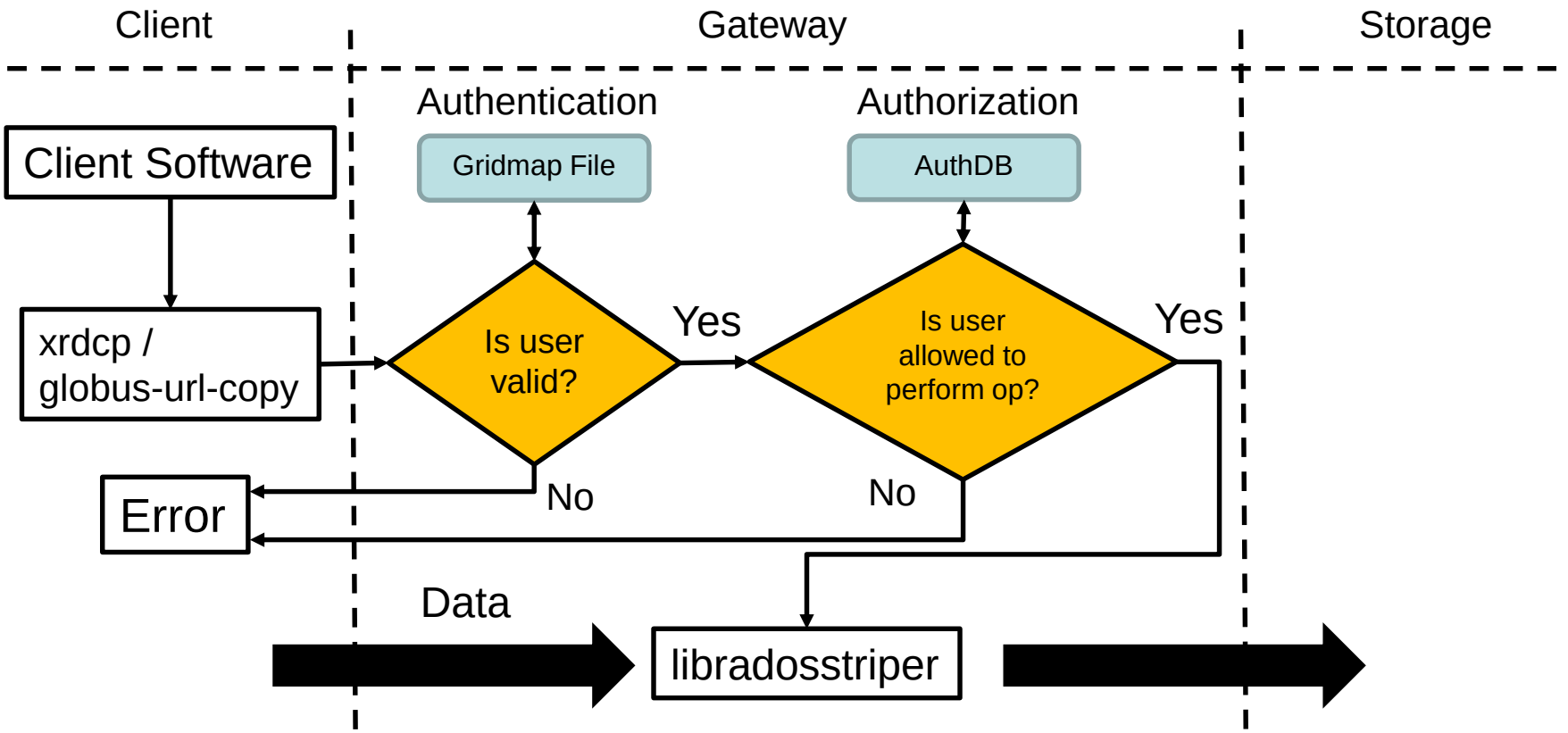
- XRootD plugin was developed by CERN.
  - Plugin is part of XRootD server software
  - They have a very restricted use case.
  - They switched from EC back to using replication.
- We have a more complex setup.
  - Direct I/O on EC pools was predictably slow.
    - Investigating XRootD memory caches on the gateways.
- We have encountered many bugs that have been fixed.
  - Check-summing, Redirection and Caching proxy

GridFTP Plugin

- GridFTP plugin was completed at start of October 2016
  - Development has been done by Ian Johnson (STFC)
  - Work started by Sébastien Ponce (CERN) in 2015
- ATLAS are using GridFTP for most transfers currently
  - XRootD just used for reading files for analysis jobs
  - CMS Debug traffic and load tests also use GridFTP
- Recently improvements have been made to:
  - Transfer timeouts
  - Check-summing (to align with XRootD)
  - Multi-streamed transfers into EC pools

https://github.com/stfc/gridFTPCephPlugin

XRootD/GridFTP AuthZ/AuthN

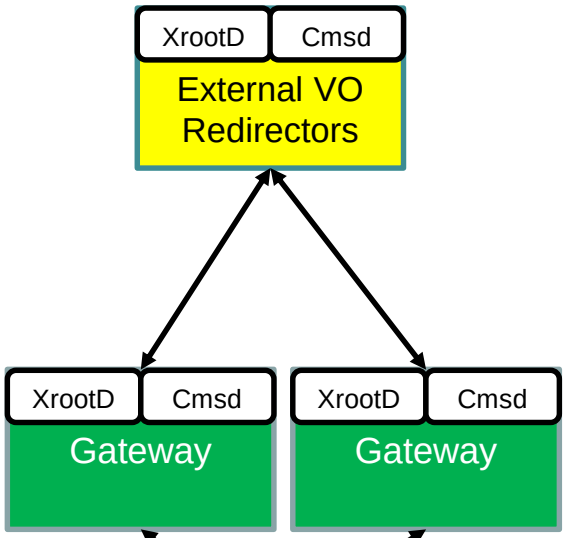


- Gridmap file used for authentication
- Authorisation is done via XRootD's authDB
  - Ian Johnson added support for this in the GridFTP plugin

Removing Gateway bottleneck for WNs

- Having all worker nodes talk to Ceph through the gateways seems like a bad idea
- Just need a few configuration files + keyring and any machine can talk directly to Ceph.
  - Make WN gateways!
- Testing on a small number of WNs running an XRootD gateway in a containers.
- Container work led by Andrew Lahiff

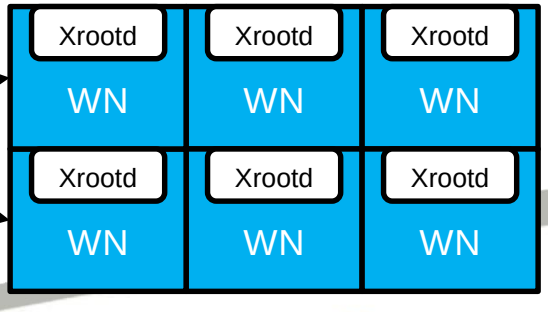
XRootD Architecture



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

Future Direction

S3 / Swift

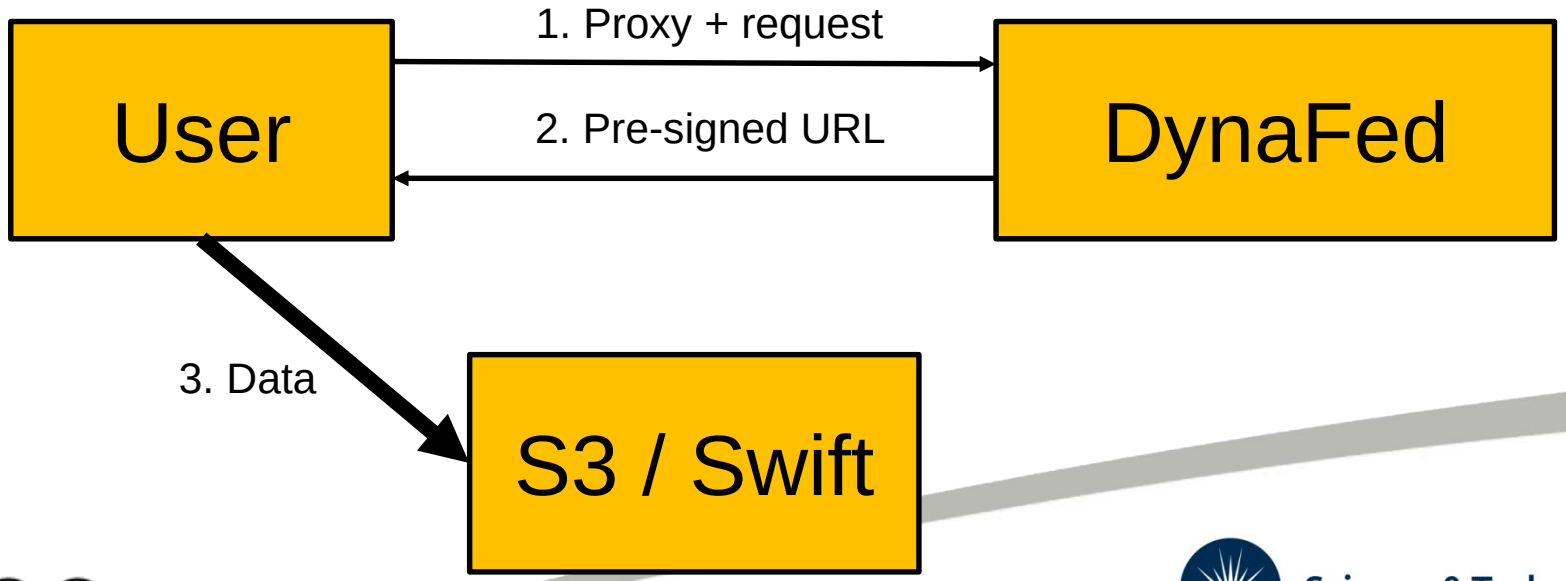
- We believe S3 / Swift are the industry standard protocols we should be supporting.

> S3 / Swift API access to Echo will be the only service offered to new users wanting disk only storage at RAL.

- If users want to build their own software directly on top of S3 / Swift, that's fine:
  - Need to sign agreement to ensure credentials are looked after properly.
- We expect most new users will want help:
  - Currently developing basic storage service product that can quickly be used to work with data.

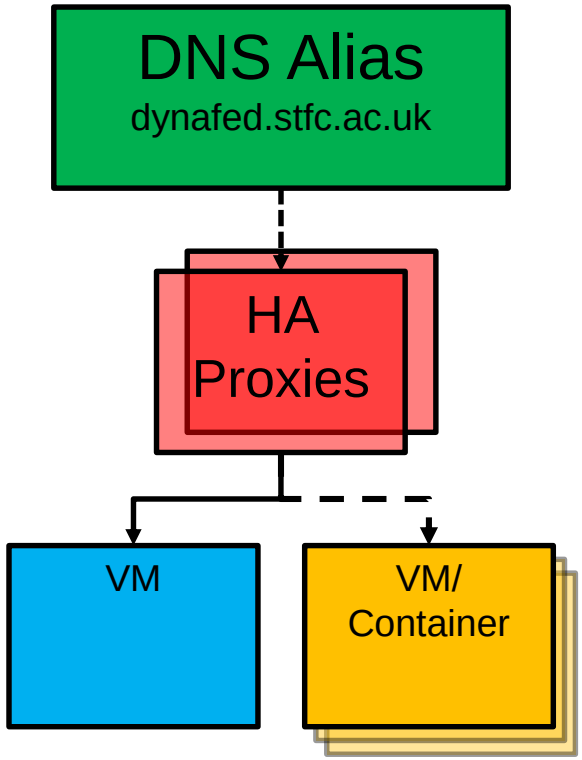
DynaFed

- We believe DynaFed is best tool to allow small VOs secure access.
  - S3/Swift credentials stored on DynaFed Box.
  - Users use certificate/proxy.
- Provides file system like structure.
- Support for transfers to existing Grid storage.



RAL Dynafed Setup

- Service is behind HA proxy.
  - Currently just one VM but easily scalable.
- Will be in production in the next 6 months.
  - Ian J will be spending 50% of his time working on Dynafed.
- Anyone with an atlas or dteam certificate can try it out:
  - https://dynafed.stfc.ac.uk/gridpp



CLI

```
# voms-proxy-init
# davix-ls -P grid davs://dynafed.stfc.ac.uk/gridpp/dteam/disk/
# davix-put -P grid testfile davs://dynafed.stfc.ac.uk/gridpp/dteam/disk/testfile
Or
# gfal-ls davs://dynafed.stfc.ac.uk/gridpp/echo/
# gfal-copy file:///home/tier1/dewhurst/testfile davs://dynafed.stfc.ac.uk/gridpp/dteam/disk/testfile2
```

Conclusion

- Echo is in production!
- There has been a massive amount of work in getting to where we are
  - Support for GridFTP and XRootD on a Ceph object store are mature
  - Thanks to Andy Hanushevsky, Sébastien Ponce, Dan Van Der Ster and Brian Bockelman for all their help, advice and hard work.
- Looking forward: industry standard protocols are all we want to support
  - Tools are there to provide a stepping stones for VOs

Science & Technology Facilities Council

Thank you

Science & Technology
Facilities Council

Extra slides

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-dowload) (1)	137.518	0.0253456	6255.06	1975.19
CASTOR (with lazy-dowload) (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