THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

High performance S3 access with IBM Storage Scale, ECE and IBM Storage Fusion

Jake Carroll, Chief Technology Officer, Research Computing Centre, The University of Queensland, Australia.

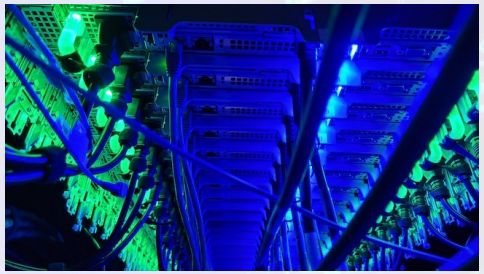
jake.carroll@uq.edu.au

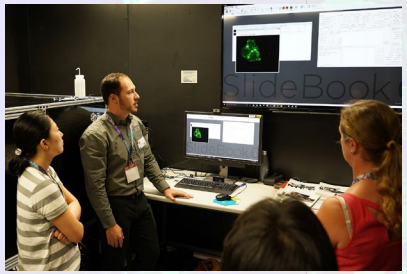
Context

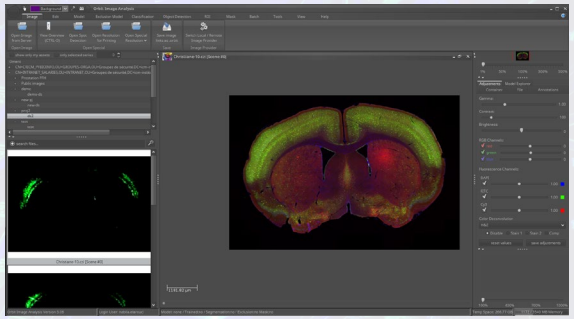
UQ is:

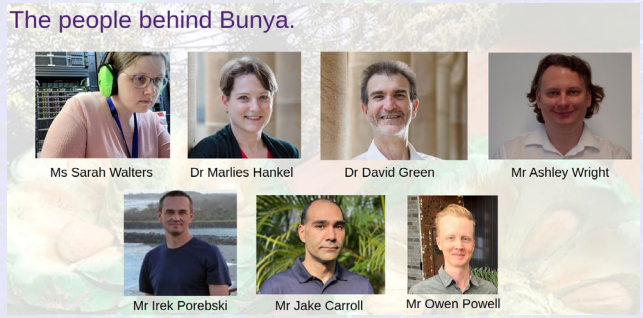
- Six major faculties
- Eight institutes
- Fifteen sites
- ~55,400 students (2022)
- ~7,410 full time staff (2022)
  - 4000 researchers
- ~25,000 endpoints
- Tier2 supercomputer: more than 10,000 CPU cores, cutting edge GPUs.
- ~100 PB of research data storage under management.
- A whole bunch of ESS systems. GH14's, ESS3000's, ESS5000's, ESS3500's….*and an ESS 6000 (!)…*

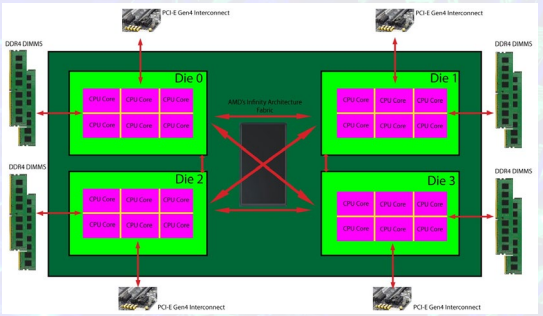
UQ Research Computing Centre











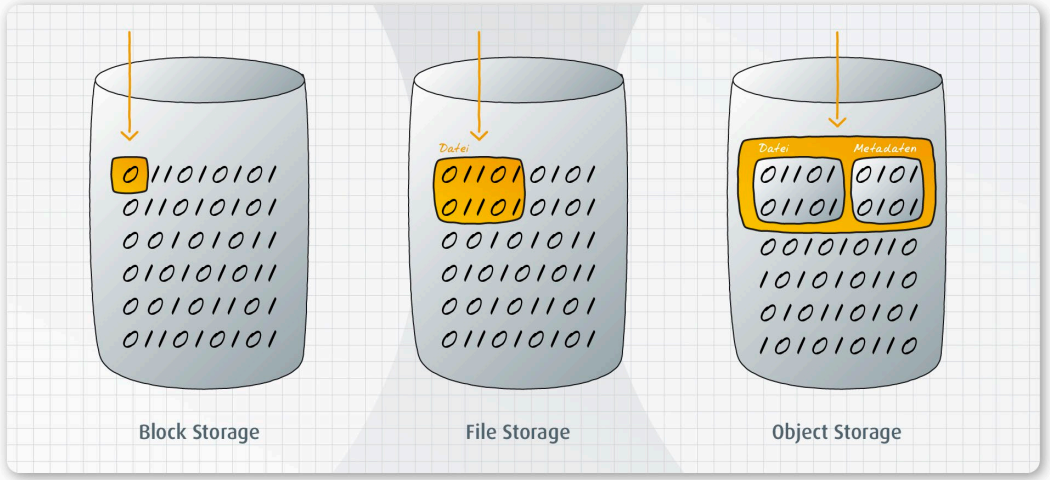
Story time...

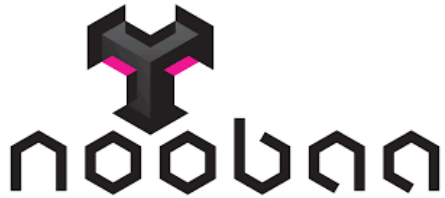
- A few months ago, we caught up with Ulf in Germany at ISC 2023.
- We learned lots about *Sauerkraut*, but we also talked about protocols and what might come next in ESS.
- S3 has had an odd and bumpy life in IBM Storage Scale land. It seems to have changed shape as often as IBM marketing people change the name of their products.
- Andrew Beattie and I went back to the lab to see where we might take this.

The Protocols Talk

Qualifying the S3/object use case

- The use of object storage is *still* contentious in the research and advanced computing communities.
- The worthwhile nature of object storage for scientific, supercomputing and research computing communities has been limited due to application, performance and flexibility limitations.
- Persistently, we see people wanting to treat their S3 object targets like filesystems. *“I want to browse it with Cyberduck like windows explorer/Mac Finder over SMB”*
- A use case emerged a couple of months ago where the only feasible way to ingress and egress data over long distances was `https`.





“Can we have S3 please, Jake?”

The University of Queensland, Brisbane, Australia

Problem domain: Comparatively high latency, comparatively low bandwidth. Limited native filesystem connectivity options. Needs to be convenient access for “stashing” and “hauling” data.

The Australian National University, Canberra, Australia

Getting data to and from a supercomputer in another state of Australia: 1,188km away (738 miles)

The ***Bunya*** Supercomputer.

Storage Scale – ESS3500: AFM Cache

The ***Gadi*** Supercomputer.

No AFM target, or IBM Storage Scale targets here. All just Lustre scratch.

The Anatomy of our S3 + Storage Scale setup, so far.

UQ, St Lucia Campus

ESS 5000/3000 running containers with Noobaa/NSFS.

S3 https object access to AFM Filesets (cache)

AFM cache to home relationship (NSD, 100Gbps), 30km

AFM Home

ESS 3500+5000 Storage Scale Filesystems

```
/UQ00
/UQ01
/UQ0..
/UQ07
```

CES Ganesha NFS Exports

Polaris Data Centre

Light weight events via Ganesha interface processed by DMF7 to events_filter for DMAPI processing

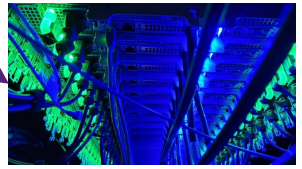
dmf7-gpfs-dm1
dmf7-gpfs-dm2

200G HDR fabric

Cassandra Object Scale db, all nvme HDR200 connected event processing and reflection table generation cluster

CRICOS code 00025B

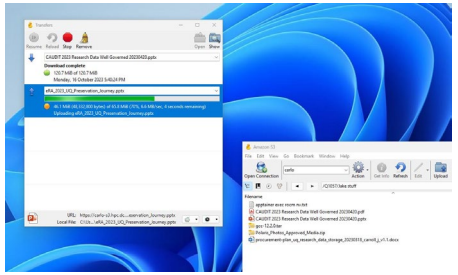
DMF7 DMAPI managed GPFS filesystems tier into tepid and cold storage layers



HPC ***Bunya***

noobaa

Storage Scale CES-S3 Service

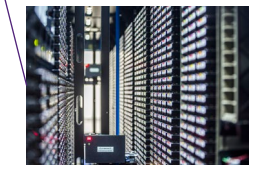


Cyberduck Client or whatever…

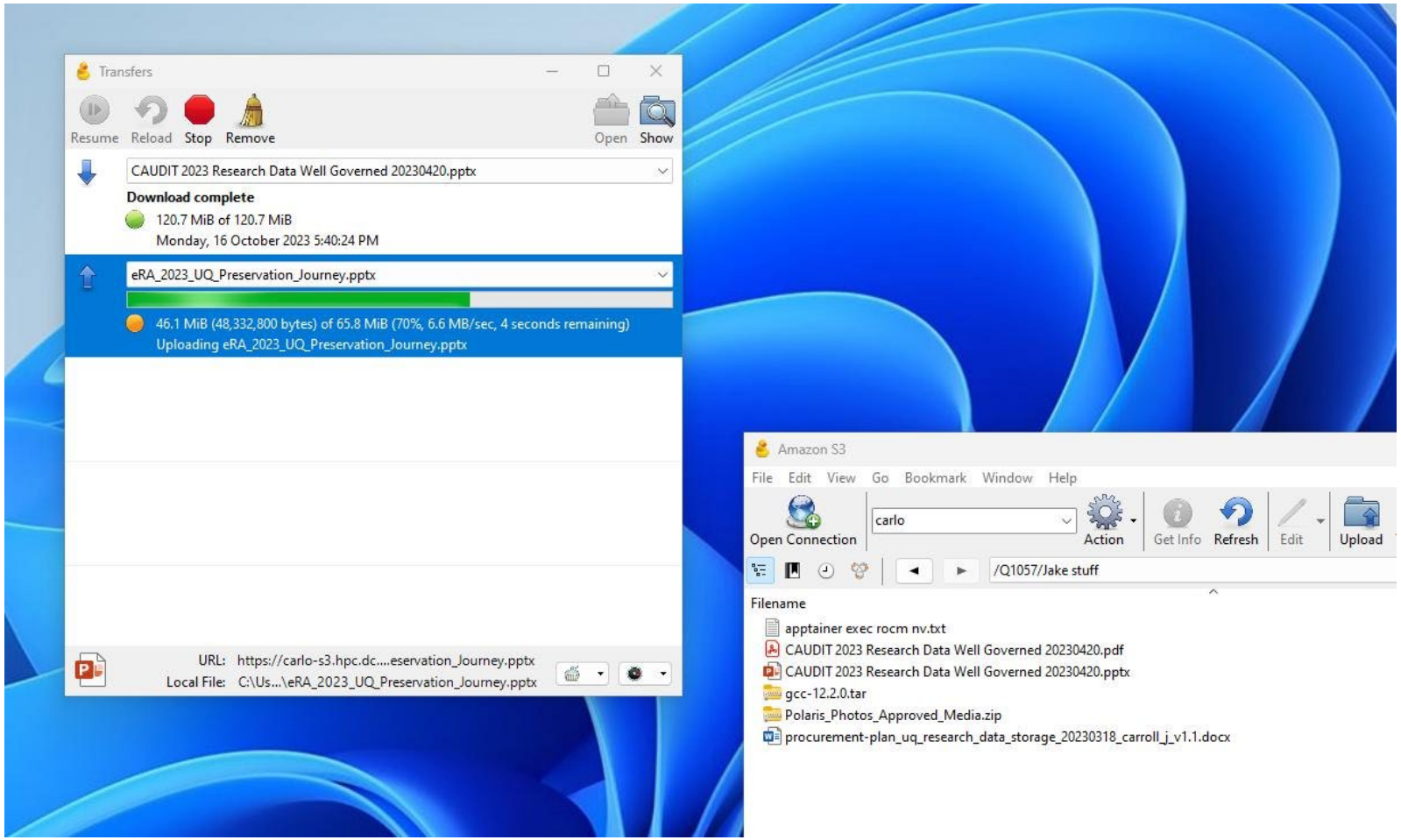
Fusion Multi Cloud Gateway
Fusion HCI AFM → S3 Service

Fusion

AFM filesets to S3 front end atop Scale ECE or back to S3 object store on Carlo



What does it look like, to a user?



TL;DR – Browse your AFM fileset, just like you would an SMB or NFS share out of a regular CES protocol node stack.

How does this knit together?

- ***Three*** scenarios to express. This is the Noobaa Service:
  - running inside Fusion HCI. It is acting as an AFM front end (like a scale AFM fileset) and points back to an AFM home (carlo) over https, to our CES-S3 service.
  - running inside Fusion HCI and pointing at the internal ECE filesystem with a persistent claim for an S3 object store backend.
  - running inside Carlo's Power9 nodes to provide CES-S3, exposing AFM filesets *as an S3 representation* for S3 GET's and PUT's
- Delivering S3 from either the Fusion HCI Scale filesystem (S3 to persistent volume claim) or…
- S3 from External object store mapped into the Fusion cluster as an internal S3 data service

Why is that significant?

- We can use the Fusion MCG (Multi Cloud Gateway) to write to multiple different external object stores simultaneously..

Benchmarking

COSBENCH - Controller Web Console

Beta Release
version: 0.4.1

Controller Overview

Name: *not configured* **URL:** *not configured*

Driver	Name	URL	IsAlive	Link
1	release-name-cosbench-driver-0.release-name-cosbench-driver	http://release-name-cosbench-driver-0.release-name-cosbench-driver:18088/driver	●	view details
2	release-name-cosbench-driver-1.release-name-cosbench-driver	http://release-name-cosbench-driver-1.release-name-cosbench-driver:18088/driver	●	view details
3	release-name-cosbench-driver-2.release-name-cosbench-driver	http://release-name-cosbench-driver-2.release-name-cosbench-driver:18088/driver	●	view details
4	release-name-cosbench-driver-3.release-name-cosbench-driver	http://release-name-cosbench-driver-3.release-name-cosbench-driver:18088/driver	●	view details

submit new workloads

config workloads

advanced config for workloads

Active Workloads 0

☐	ID	Name	Submitted-At	State	Order	Link

Cancel

Historical Workloads 11

view performance matrix

☐	ID	Name	Duration	Op-Info	State	Link
☐	w1	s3-carlo	Oct 27, 2023 2:08:20 AM - 2:24:20 AM	init, prepare	finished	view details
☐	w2	s3-carlo	Oct 27, 2023 2:38:39 AM - 2:38:47 AM	init, prepare	terminated	view details
☐	w3	s3-carlo	Oct 27, 2023 2:39:25 AM - 2:39:40 AM	init, prepare	cancelled	view details
☐	w4	s3-carlo	Oct 27, 2023 2:41:51 AM - 2:43:45 AM	init, prepare	cancelled	view details
☐	w5	s3-carlo	Oct 27, 2023 2:43:56 AM - 2:48:46 AM	init, prepare	finished	view details
☐	w6	s3-carlo	Oct 27, 2023 3:03:51 AM - 3:08:32 AM	init, prepare	failed	view details
☐	w7	s3-carlo	Oct 27, 2023 3:09:15 AM - 3:10:18 AM	init, prepare, write, cleanup, dispose	finished	view details
☐	w8	s3-carlo	Oct 27, 2023 3:11:18 AM - 3:11:26 AM	init, prepare, write, read, cleanup, dispose	terminated	view details
☐	w9	s3-carlo	Oct 27, 2023 3:11:50 AM - 3:13:13 AM	init, prepare, write, read, cleanup, dispose	finished	view details
☐	w10	write1GBfull	Oct 27, 2023 3:13:33 AM - 3:17:22 AM	init, prepare, write	cancelled	view details
☐	w11	s3-carlo	Oct 27, 2023 3:20:56 AM - 3:26:02 AM	init, prepare	failed	view details

Archived Workloads 0

view performance matrix

load archived workloads

resubmit

Workload

Basic Info

ID: w13 **Name:** write1GBfull **Current State:** processing **Current Stage:** 1GB1

Submitted At: Oct 27, 2023 3:32:23 AM **Started At:** Oct 27, 2023 3:32:23 AM **Stopped At:** N/A

more info

Snapshot

General Report

Op-Type	Op-Count	Byte-Count	Avg-ResTime	Avg-ProcTime	Throughput	Bandwidth	Succ-Ratio

The snapshot was taken at 3:32:56 AM with version 2.

Stages

Current Stage	Stages completed	Stages remaining	Start Time	End Time	Time Remaining
1GB1	0	7	3:32:23 AM		

ID	Name	Works	Workers	Op-Info	State	Link
w13-s1-1GB1	1GB1	1 wks	1 wkrs	write	running	view details
w13-s2-1GB8	1GB8	1 wks	8 wkrs	write	waiting	view details
w13-s3-1GB32	1GB32	1 wks	32 wkrs	write	waiting	view details
w13-s4-1GB64	1GB64	1 wks	64 wkrs	write	waiting	view details
w13-s5-1GB128	1GB128	1 wks	128 wkrs	write	waiting	view details
w13-s6-1GB256	1GB256	1 wks	256 wkrs	write	waiting	view details
w13-s7-1GB512	1GB512	1 wks	512 wkrs	write	waiting	view details

There are 7 stages in this workload.

Actions

cancel-workload

go back to index

How is it looking, in real life?

```
[user123@gadi-login-04 scratch]$ s3cmd get s3://Q3967/TestDir/xx.csv.gz ./

download: 's3://Q3967/TestDir/xx.csv.gz' -> './xx.csv.gz'  [1 of 1]

3502926275 of 3502926275   100% in   56s    58.65 MB/s  done

[user123@gadi-login-04 scratch]$ s3cmd put xx.csv.gz s3://Q3967/TestDir/xx2,csv,gz --
disable-multipart

upload: 'xx.csv.gz' -> 's3://Q3967/TestDir/xx2,csv,gz'  [1 of 1]

3502926275 of 3502926275   100% in  127s    26.16 MB/s  done
```

I get around 60sec for 3GB of put/get , but I am pretty sure that could be optimised with a better .s3cfg configuration.

After some practice, I can now do the same in about 30sec!

Where to from here?

There is still some work to do to make this a production grade in our opinion.

1. It isn't HA (yet). Single point of failure in CES-S3 protocol stack pinned into a node.
2. It isn't scalable from an auth-setup, key handout and secret sharing process.
3. There is no automation yet to scale up our existing filesets over AFM to make them all S3 and object accessible "at the click of a button".

Where to from here – auth complexity to solve?

We have some “one to many” vs “many to one” auth challenges, too. UQ’s auth model will likely demand individual keys and secrets per user. Perhaps that goes without saying, but consider…

- That same user might want access to multiple buckets. *(Common use case!)*
- That has ACL issues, instantly – as buckets are mapped to a single GID.
- If you have multiple users with multiple UIDs/GIDs they can’t all have individual keys with out some sort of internal database to map or some kind of external authentication system to manage it

Basket weaving, at the moment.

There is a lot of retrofitting, scripting and glue to be put in here that doesn't exist yet. We think IBM might need to do a bit of work on this one to make it more robust from the auth perspective.

Irek, Andrew and Dale had to manually hand-craft our first S3 enabled AFM fileset parameters for our users whom asked for them. We aren't at "click a button and share" status yet.

Thank you.

UQ:

- Irek Porebski

IBM

- Andrew Beattie, Dale McCurdy