ICR The Institute of Cancer Research

Adventures with AFM

(or how we migrated from v4.2 to 5.1)

Storage Scale User Group – 27/28 June 2023 – Robert.Horton@icr.ac.uk

Overview

- About US
- Our data migration using AFM
- Other AFM experience
- Things we learnt along the way (not just AFM)

What is the ICR?

- A world-leading research institute
- Close partnership with Royal Marsden Hospital
- A member institution of the University of London
- A charity
- Employs 1,100 scientific and professional staff
- Has 200 postgraduate research students
- Not to be confused with Cancer Research UK ;)

800 scientists across two sites in London

Chelsea (central London)

Chester Beatty Laboratories
Fulham Road

30 minutes from Heathrow airport

Central London

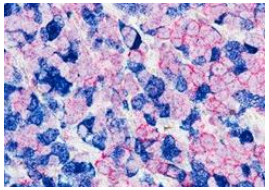
Sutton (south London)

Centre for Cancer Drug Discovery,
Opened in 2021

Centre for Cancer Imaging
Opened 2015

30 minutes from Gatwick airport

Research divisions

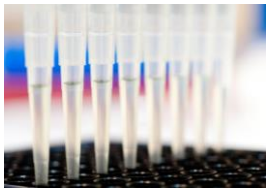


Breast Cancer Research

Cancer Biology

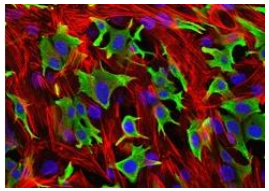
Clinical Studies

Cancer Therapeutics

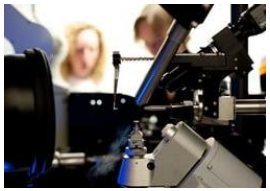


Genetics and Epidemiology

Molecular Pathology



Radiotherapy & Imaging



Structural Biology

Research Data Storage Platforms

History of multi-tier, multi-site data storage platforms

- “Pandorica” – SGI CXFS 2 Sites with 2 disk and 1 tape tier.
- “Gallifrey” – Quantum StorNext, 2 sites, 1 disk and 1 tape tier.
- “RDS” – GPFS stretched filesystem based on 2 DDN GS7Ks, plus DDN WOS tier, connected by DDN GS Bridge DMAPI handler.

In all cases:

- Complex to run, savings due to cheaper tiers didn’t really compensate.
- Having 2 copies on fast tier is expensive.
- Write performance in particular not great due to slow inter-site links
- Data migration a bit of an after thought!

RDS2

Went to tender in June 2021 and ended up with:

- IBM ESS-5000 SL5 (later expended with an additional SL2)
- 2 Power protocol nodes
- TS4500 with 12 LTO-9 drives
- Lenovo x86_64 Spectrum Protect Server + Client (and licenses)

Installed and handed over around April 2022.

What about the data on RDS1?

We needed to be able to able to switch to new hardware with minimal downtime and preserving access to data.

Adding NSDs to existing filesystem not really an option due to smaller sub-block option in 5.x, and in any case:

- We'd be going across 2 major versions (4.2 -> 5.1).
- Going from 2 replicas to just 1.
- New system had completely new network environment (IB/100GbE switches/IP ranges) which we wanted to keep separate.

What about rsync then?

The obvious “safe” method is to sequentially sync data from the old to new system then take old system offline for final sync.

- Rsync isn’t going to cut it due to being single threaded.
- Tools such as dsync from mpifileutils or fpsync would improve the transfer performance.
- Commercial options such as Myria also integrate with policy engine.
- Mmxcp wasn’t an option then but might be now.

Some issues with all these though…

Issues with rsync (et al)

- Old system was very full, plus concerns over hardware health so a desire to new data written to the new system as quickly as possible.
- Unclear how long initial copy might take, given WOS performance. Similarly unclear how long to allow for the final sync.
- All data needed to be backed up anyway, potentially the rate limiting step.

So how about AFM?

Why AFM?

- New data only written to new storage and each existing file only gets read once, reducing stress on old storage.
- Existing files migrated as read; remaining data to be migrated in the background; if this is a bit slow not a major problem.
- mmbackup skips non-migrated files out of the box, so the first backup doesn’t cause a mass recall. (Although doing a restore of a partially migrated fileset would be messy)
- Recognised migration route: https://www.ibm.com/docs/en/storage-scale/5.1.8?topic=afm-data-migration-by-using-migration-enhancements

We tried AFM before…

Aim was to make transferring data between HPC and RDS easier.

- Using then-new HPC scratch (Lenovo DSS-G) as cache for RDS
- 2 AFM gateway nodes remote mounted the RDS (via some complicated routing)
- Created IndependentWriter cache filesets on DSS filesystem for a few teams to test.

We tried AFM before…

It seemed to work at first but…

- Started getting evictions in the home cluster. Sometimes the gateway nodes, but sometimes other things. Possibly due to the funky networking, in retrospect nfs would have been a safer choice.
- Slow file writes on cache – improved by setting afmFastCreate.
- Queue flushing getting stuck, regular restarts needed
- Reads of small files slow even when cached, cache still needs to contact home to check they haven't changed.
- Kernel soft lockups on HPC nodes, later confirmed to be an AFM bug, fixed in 5.0.5-6
- Unfortunate interaction between AFM, DMAPI handler and WOS when moving files on cache.

We pulled the plug at this point…

So why use it for the migration?

- Whilst the trial was ultimately unsuccessful, we learnt a lot from it.
- Improvements to AFM in the meantime.
- Much less to go wrong in local-update compared to independent-writer cache.
- Lack of alternatives given concern over existing kit and desire to get new ESS online as soon as possible.

Migration procedure

Refer to IBM's documentation over this!

- NFS mounted the home filesystem on 2 AFM gateways.
  - We used kernel NFS3, a back-channel (mmafmconfig enable on home) provides support for sparse files, acls etc.
  - We used the 2 protocol nodes as AFM gateways, probably not really advisable but we got away with it.
  - We previously had a mix of "posix" and nfs4 acls but did a mmchfs –k nfs4 prior to migration to match new filesystem.
- Create new filesets with:

mmcrfileset <fs> <fileset> -p afmMode=LU,afmEnableAutoEviction=no,afmRefreshOnce=yes,afmReaddirOnce=yes,afmTarget=nfs://<path>

Migration procedure (2)

- Switch you exports / mounts etc to the new system then go home and relax ;)
- Force metadata migration with:

Mmafmctl fs prefetch fileset --metadata-only

- Force data migration with:

Mmafmctl fs prefetch fileset

- Check for unmigrated data with

Mmafmctl fs checkUncached fileset

Migration procedure (3)

Fix any uncached files by

- Running the prefetch again
- Turning off afmRefreshOnce then running the migration again
- If all else fails – just copy the file manually ☹ (Only came to this a few times luckily!).

How did it go?

- We made the switch during a quiet week and allowed 48 hours of downtime (it actually needed far less).
- Fairly quickly most reads were cache hits so the bulk migration could proceed in the background.
- Issue with semi-migrated(?) metadata causing very slow policy scans; resolved by doing a metadata-only migration.
- Some issues with reading (particularly migrated) files but minimal user disruption.
- Slow nfs home sometimes caused snapshots to fail / very long waiters / occasionally evictions – now fixed though.
- Discovered it was better to rehydrate all files on home before migrating.
- Whole migration took around 9 months, delay was mostly due to needing to expand ESS

Conclusion

- We achieved a switchover with minimal disruption so would consider it a success!
- It worked well for our circumstances but not the most straightforward migration method – other circumstances may favour a more conventional approach.
- Luckily we didn’t need to do any (significant) restores during the migration.

Things we learnt (1)

NFS isn’t that bad ;)

- I’d previously considered a native to be better in all circumstances, but
- No evictions with NFS
- Nicer client behaviour if the server stops responding (briefly)
- Clients don’t need the keys to the kingdom
- Time to move to NFSv4 though

Things we learnt (2)

Cluster members all need to be able to contact each other on their daemon address

- Sounds so simple!
- Including remote mounts
- Beware: complicated routing, private (NAT) networks, stretch clusters, mmchconfig networks etc.
- A node being able to connect to the cluster & mount a filesystem isn’t proof everything is set up correctly.

Things we learnt (3)

NFSv4 ACLs…

- Previously had posix and nfs4 acls enabled, primarily used posix permissions (with create umask in Samba).
- Protocols config requires –k nfs4 only. How complicated can it be??
- Didn’t really understand FileInherit/DirInherit until we needed them – ideally we’d have “fixed” all this on the old filesystem before migration.
- Still haven’t completely worked out how to manage these – maybe support for nfs4-acl-tools will help.

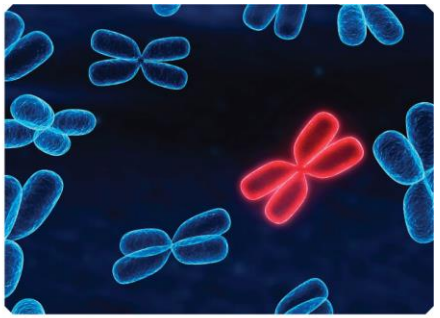
ICR The Institute of Cancer Research

Any Questions?

Making the discoveries that defeat cancer

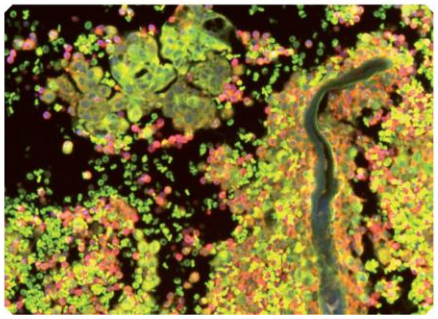
Unrivalled track record

ICR The Institute of Cancer Research



Making the discoveries that defeat cancer

ICR



One of the world's most influential cancer research institutes