Big Internet Pipe and Cloud Saved My Storage in Crisis

Dan Sun

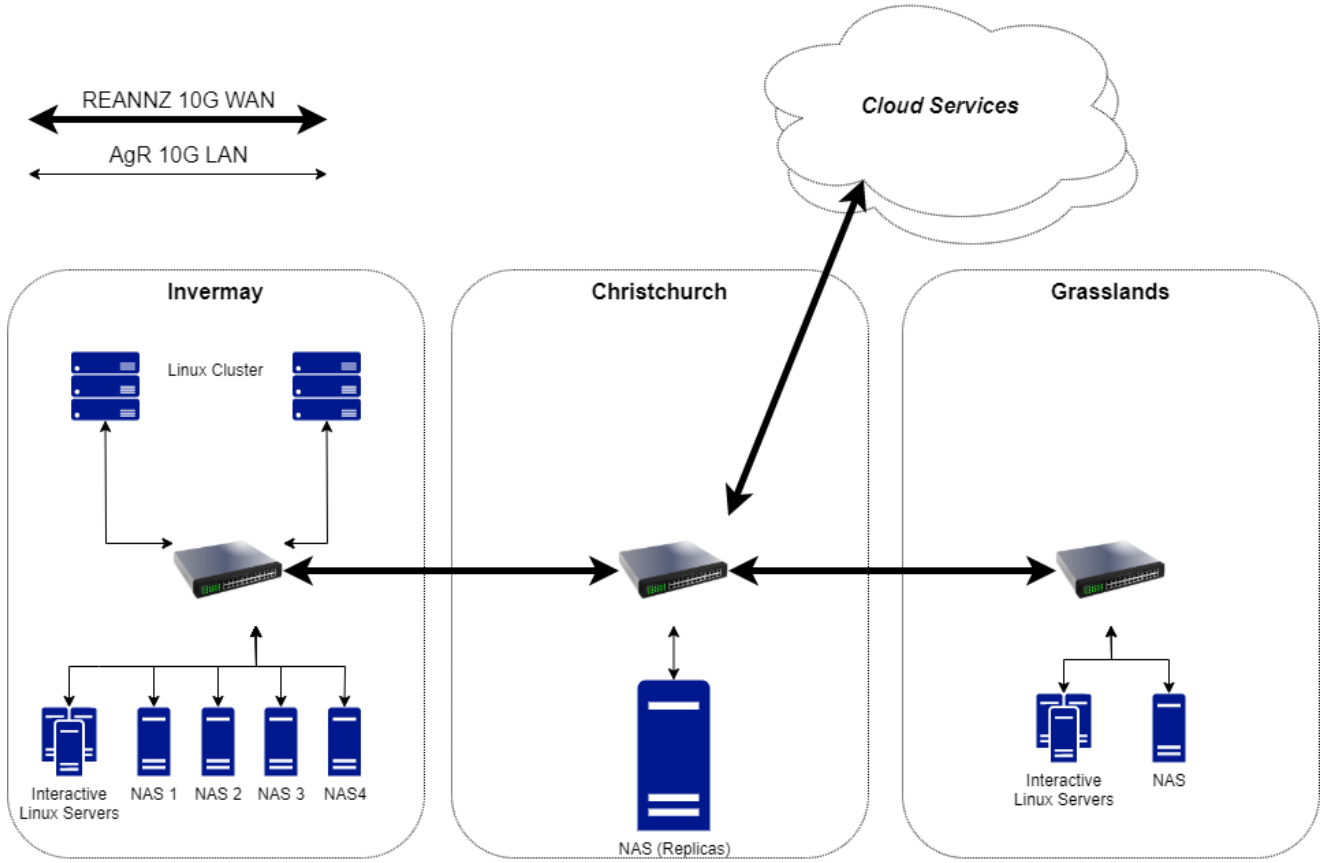
dan.sun@agresearch.co.nz

agresearch

āta mātai, mātai whetū

Current Linux Infrastructure in AgResearch

- Resources are distributed across three sites
- Storage is provided by individual Network Attached Storages (NAS)
- Files are organised in *Datasets* and compute resources access them via *NFS*
- Files are protected by snapshots and replications



When storage utilisation went above 85%...

- Performance degraded rapidly
- Users were complaining about their jobs failing unexpectedly
- Filesystem was essentially unusable

Our Simple Recovery Plan

- Copy some data somewhere else quickly and delete it on the NAS that is over-utilised

- ***However***
  - Our *snapshot* policy keeps deleted files in snapshots of their parent datasets for up to 6 months and removing individual files in a snapshot is not possible
    - *Solution*: move entire datasets

  - There is no other storage in AgResearch that has spare capacity and can be utilised immediately
    - *Solution*: utilise the Cloud

More Considerations

We wish to maintain the same data protection level throughout the remediation process

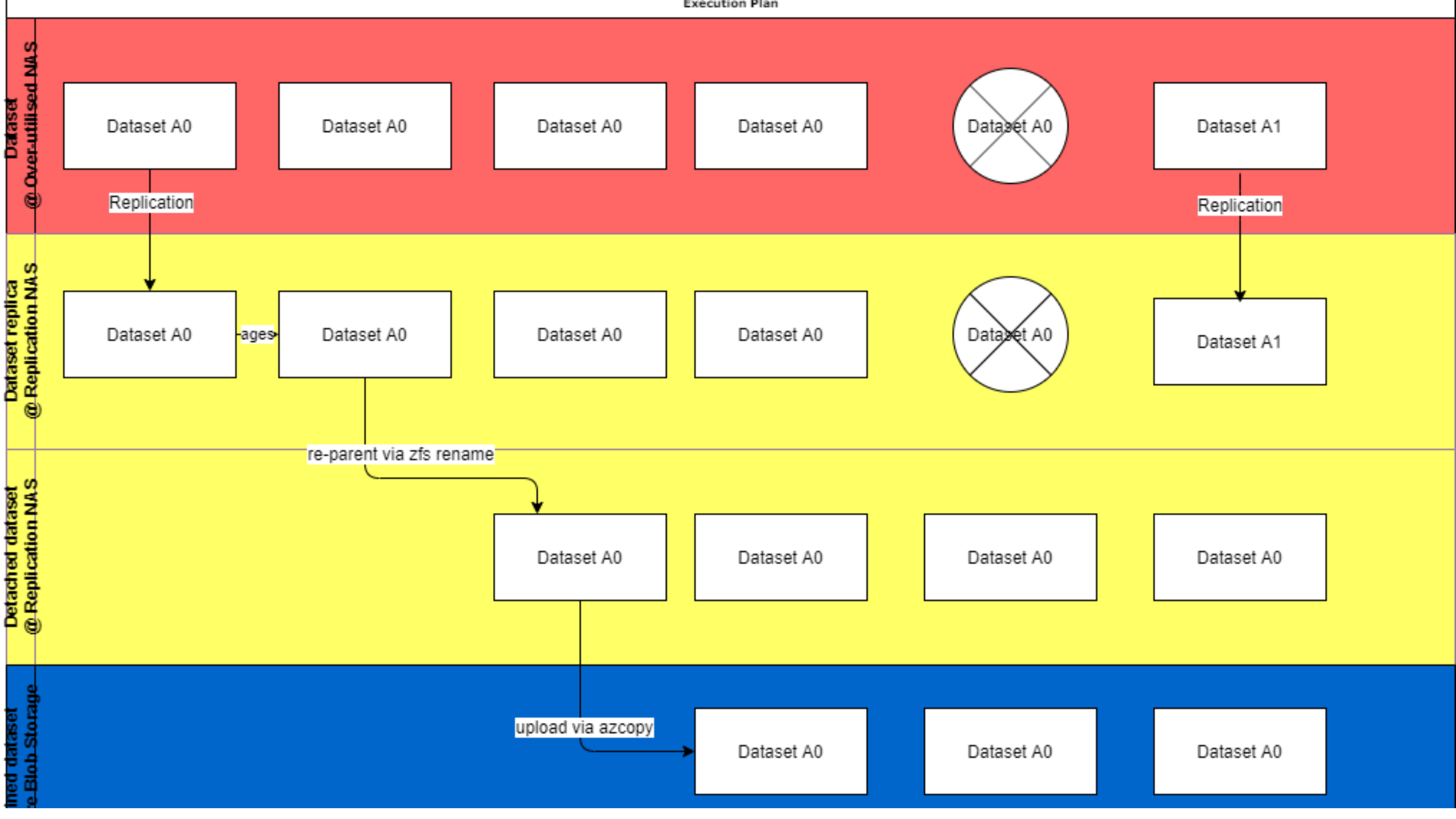
– i.e. two copies of the same data in two different sites

The NAS which stores all replications is closer to AgResearch's link to the cloud and is not utilised during business hours

– i.e. it is a better data source for uploading data to the Cloud

Storage in the Cloud may be expensive over a long period of time

– i.e. Archival object store is the most fit for purpose


Execution Plan
Dataset
@ Over-utilised NAS
Dataset A0
Dataset A0
Dataset A0
Dataset A0
Dataset A0
Dataset A1
Replication
Replication
Dataset replica
@ Replication NAS
Dataset A0
ages
Dataset A0
Dataset A0
Dataset A0
Dataset A0
Dataset A1
re-parent via zfs rename
Detached dataset
@ Replication NAS
Dataset A0
Dataset A0
Dataset A0
Dataset A0
upload via azcopy
Dataset A0
Dataset A0
Dataset A0

Some interesting technical observations

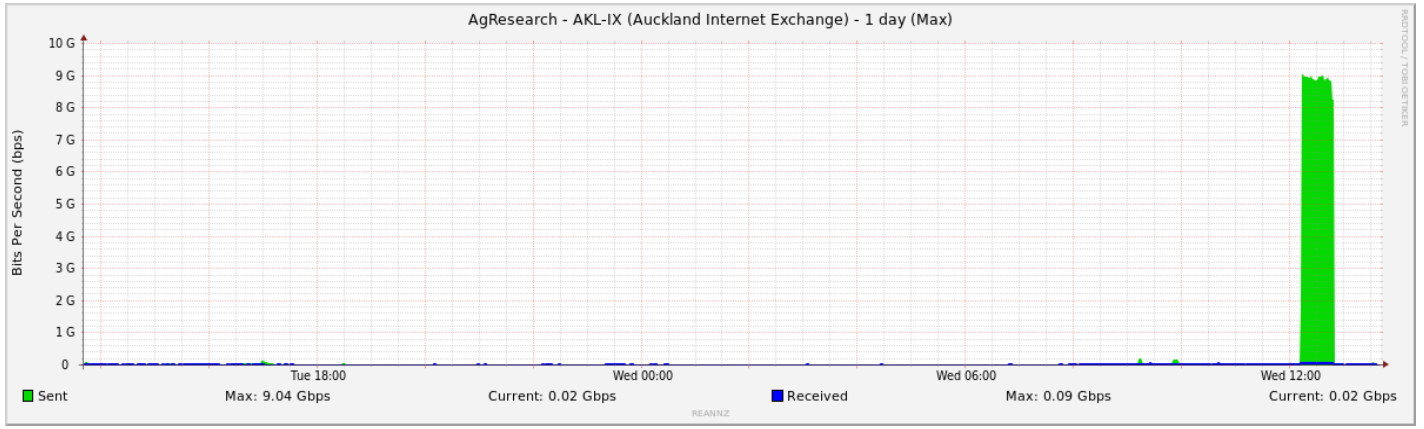
Performance of OS file level operations are poor when working with many files. We used lower level filesystem tools to move all files in a dataset and to remove all files in a dataset.

Both ***AzCopy*** and ***Blobfuse*** can be used to upload files to MS Azure's Blob storage, but *AzCopy* uploads data significantly faster. However *Blobfuse* allows you to treat an Azure Blob container/bucket as a local filesystem, which is user-friendly if a user ever needs to navigate an archived filesystem and retrieve a small number of files.

You can only upload data to the Cool tier of MS Azure's Blob store; instead of moving uploaded data to the Cold (archive) tier manually, we used MS Azure's lifecycle management tool to automate this process.

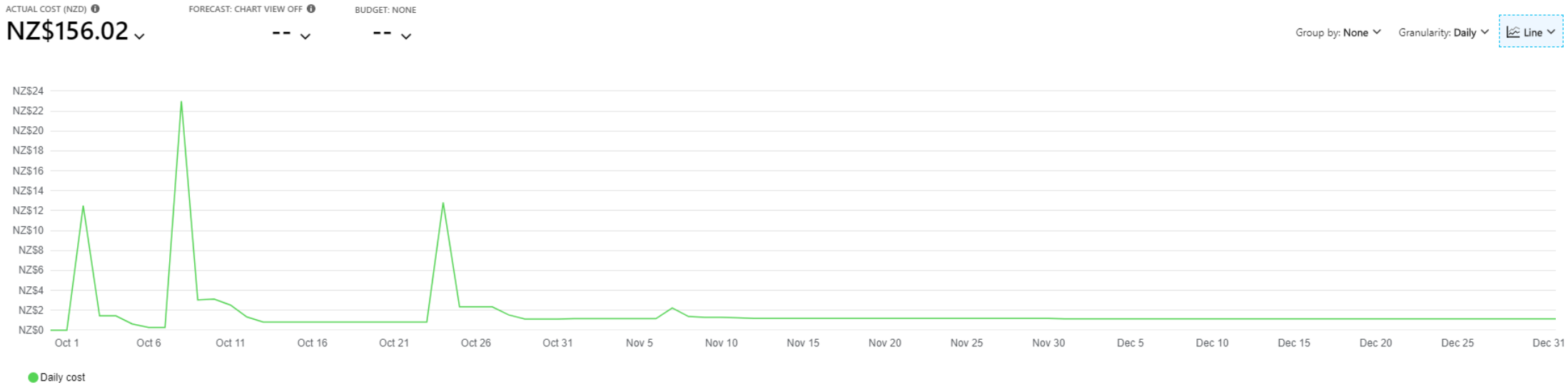
Some notable observations

INFO: Scanning...
INFO: Using OAuth token for authentication.
Job cdc79604-8997-da45-4805-6d33e83c970b has started
Log file is located at: /home/sund/.azcopy/cdc79604-8997-da45-4805-6d33e83c970b.log
5459 Done, 0 Failed, 4 Pending, 0 Skipped, 5463 Total, 2-sec Throughput (Mb/s): 158.3198 (Disk may be limiting speed)
Job cdc79604-8997-da45-4805-6d33e83c970b summary
Elapsed Time (Minutes): 37.0855
Total Number Of Transfers: 5463
Number of Transfers Completed: 5463
Number of Transfers Failed: 0
Number of Transfers Skipped: 0
TotalBytesTransferred: 2244166874726
Final Job Status: Completed



We transferred approximately 2TB in 37 minutes
(i.e. 7.5Gbps in average over a 10Gbps link)

Costing in Q4 2019

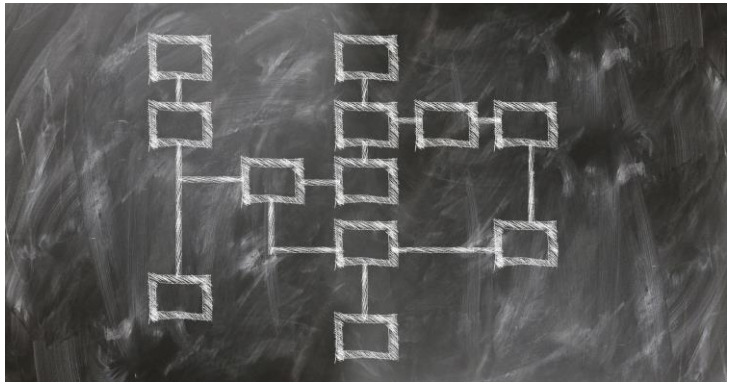


Estimated Cost for Strong 8TB of data per Month			
Files	Blob (Hot tier)	Blob (Cold tier)	Blob (Archive tier)
$540	$164	$122	$23

Moving data to archive tier is making a big difference!!!

It’s a tactical success, but it’s not a silver bullet

- Offloading some data to the Cloud indeed gave us quite some breathing space and ended the crisis
- Unfortunately, we cannot use the same approach to further extend the useful life of the current storage



The Root Cause of the Storage Crisis

The Current Data Management Process

Some meta data is captured for each dataset

Each dataset is stored in three sub-filesystems

Each sub-filesystem has a different data protection level and performance characteristics

There is no Information Life Cycle defined for datasets

Users are responsible for using the “right” sub-filesystem when storing data in a dataset

Quota is not applied on datasets

The Human Element

It is inconvenient to move data between sub-filesystems in a dataset and I am busy

I ***MAY*** need the intermediate data later, so I am keeping it in the *scratch* filesystem

I am too busy to submit a request for a new dataset now. I will just store the new data in an existing dataset

The project has no funding to tidy up its data after its outcome is published

Storing data should be cheap because I can buy an 1TB external USB HDD for less than $200 in a local computer store

Very Stretched Technology

- NAS / Fileserver centric architecture has had its day
- A filesystem only storage platform is no longer fit for purpose
- Manually moving data between different tiers of storage is not sustainable
- No native Cloud tiering capability is limiting our use of the Cloud

What we learned from this incident

Asking people, particularly users, to manage a large amount of data and many datasets manually is not a sustainable approach

Technologies have limited life and they need to be refreshed regularly to acquire new capabilities to support new challenges

The Cloud should be an extension of on-premise infrastructure by design

Different Cloud resources support different use cases; select the fit for purpose resource to be cost effective

REANNZ’s network is highly capable and they do not mind heavy usage

Questions, Comments and Discussion