CloudStor: potential use as glue between ARDS components

Guido Aben

Guido Aben, Peter Elford, David Jericho
AARNet; email: firstname.lastname@aarnet.edu.au

CLOUDSTOR, THE BASICS
AARNet’s CloudStor platform has been in operation for a more than 4 years, having matured from a tentative proof of concept to a robust, supported service offering with over 25,000 users and 70TB stored.

CloudStor as currently deployed was designed to deliver a continental scale data movement and sync&share solution for the Australian research constituency. The interface functionality and client platform compatibility was intended to resemble mainstream functionality present in DropBox, while the backend was intended to be able to move science-size data volumes (Gigabytes to Terabytes, and directories with multiple thousands of files).

To date, CloudStor has operated in essentially a standalone mode; files are uploaded, downloaded, synchronised, shared and previewed all within the confines of the CloudStor system itself. This isn’t isolation by design; merely that up to this point, attention has been predominantly at ruggedizing the design and smoothing the user experience. The system as built however contains a number of potentially very useful APIs and connectors that could be used to tie together currently non-joined parts of the Australian eResearch landscape. Some of these connectors have already been tested in small scale proofs of concept and others are ready to be tested. This presentation will cover a number of compelling examples and illustrate that, with CloudStor used as a “uniform data pump”, many other entities in the landscape might work better together without themselves having to worry about large scale data movement issues.

APIS, ATTACHMENT POINTS, CONNECTORS AND INTEROPERABILITY

Where in the system can we readily plug into other workflows and/or interwork with other components and capabilities of the eResearch ecosystem? We will discuss these, first looking at the user interface and then the backend data movement components.

User Interface
To begin with the user interface, we chose to deploy the open source ownCloud platform and became a founding member in the TERENA/GEANT collective license procurement scheme to ensure our license burden would be sustainable for the foreseeable future.

As presently installed in the AARNet CloudStor system; which is to say, after customization to bring it in line with Australian eResearch practice, ownCloud delivers a number of features that align to the proposed Australian Research Data Systems (ARDS):

i) multi-platform client presence: synch clients exist for Windows, MacOS, Linux, Android and iOS. This will make born-digital ingest much easier, as data will trickle into the system as it is generated, instead of during lumpy, post-hoc upload actions that many operators loathe

ii) federated access: the web interface is accessed through the AAF and subsequently, personal identities and identities of share partners are based on AAF identities; by corollary, Identity Providers themselves can control how / with what identity a user presents within CloudStor. This may be helpful in the inherent identity matching challenges between cloudstor and legacy systems on campus.

iii) AARNet has added functionality to the federated login mechanism whereby we also look at a users’ realm (e.g., someuniversity.edu.au) and are able to use this information to change look & feel & livery, enable or disable certain add-on apps, and mount pre-emptively external storage volumes belonging to the users’ home institute.

iv) sharing functionality; given identities are dictated by the AAF, CloudStor can share to anybody who has identified itself to CloudStor via the AAF. As soon as ORCID identities become available, CloudStor can be
modified to also share to ORCID identities. An multi-vendor, open standards API is available (called “OpenCloudMesh”) and actively maintained that also allows machine-to-machine sharing, e.g., from CloudStor to an institutional repository, or a library curation system.

v) ability to read and write natively to a large number of external storage types and providers; files and directories present in these volumes can be presented as an integral, contiguous part of a user’s personal space. This will be instrumental in ensuring CloudStor can use be integrated with storage funding through the Research Data Service (RDS) program, campus-based storage or another cloud-storage environment such as Amazon S3 for its long-term large-volume archival, repository and safekeeping, so that CloudStor itself can restrict itself to only running as much working storage as it absolutely needs, and end user institutions retain control over the longer horizon data lifecycle. Some PoC trials have already been conducted in this area.

vi) definition of group accounts and mapping of individuals to groups, including finer-grained access control (not all members are necessarily entitled to read nor write all files). By harvesting metadata from systems such as CRAMS or ANDS’ Research Activity API, we could ensure these groups allocations are aware of entitlements held by this group on other capability plaforms (e.g., NeCTAR compute) and mount those proactively, according to the previous item (iv).

Data Movement Substrate

Next up is the underlying data movement system. The most poignant observation made during the exploratory phases of the early CloudStor project was that latency between client system (“the researcher”) and the sync&share server (whether built or bought; ownCloud or DropBox) is of paramount importance. Only if latency can be kept under ~20ms for any user that logs in can one reliably assume that seriously voluminous and high file count science volumes are handled speedily enough (>100Mbit/s as a minimum). The corollary was that we couldn’t build a monolithic, single-datacentre system. An additional corollary is that it’s unlikely that anybody can build such a system; not just AARNet, but other e-Infrastructures and Research Infrastructures alike. It follows that if somebody should succeed in building such a universal data pump, it might be very beneficial if it could be made available as a building block to other eScience efforts with nationwide distribution profiles.

Having tested and ruled out rather a few component subsystems, we settled on a stack that looks like this for every node:

```
[HAProxy]
  ^
  |   
  V
   / \
  [CERN EOS (storage)]  [MariaDB-Galera + MaxScale (database)]
```

Currently, an instance of such a node runs in Brisbane, Melbourne and Perth. Users (both browser as well as synch client users) are dynamically redirected (by use of anycast inside the AARNet network) to the node geographically closest to them. The system, specifically the EOS and MariaDB components, are connected to each other on a separate, dedicated AARNet 10G overlay network. For emergency redundancy reasons, the HAProxy nodes are able to redirect to a remote node should their local node stop responding to watchdog packets.

By virtue of fast networks and well-tuned software, the entire system is thus able to maintain state between both object store as well as database; any data ingested at one of the nodes is immediately replicated to the other nodes. As a science application can simply write into one node and assume safely that the same data will be available to collaborators elsewhere, who in turn only need to write into their local node. AARNet takes care of replicating across long-latency links (e.g., Perth to Brisbane). Essentially, what this enables is a National File System. We expect that this capability will assist in simplified deployment of digital science workflows across large geographies.

**ABOUT THE AUTHOR(S)**

GUIDO ABEN IS AARNET’S DIRECTOR OF ERESEARCH. HE HOLDS AN MSC IN PHYSICS FROM UTRECHT UNIVERSITY. HE’S DEFINITELY A GENERALIST MORE THAN A SPECIALIST, HIS CURRENT RESPONSIBILITY IS TO BUILD SERVICES TO RESEARCHERS’ DEMAND, AND GENERATING DEMAND FOR SAID SERVICES, WITH CLOUDSTOR AND SCIENCEDMZ PERHAPS THE MOST WIDELY KNOWN OF THOSE.