dCache, Sync-and-Share for Big Data

AP Millar\textsuperscript{1}, P Fuhrmann\textsuperscript{1}, T Mkrtchyan\textsuperscript{1}, G Behrmann\textsuperscript{2}, C Bernardt\textsuperscript{1}, Q Buchholz\textsuperscript{1}, V Guelzow\textsuperscript{1}, D Litvintsev\textsuperscript{3}, K Schwank\textsuperscript{1}, A Rossi\textsuperscript{3}, P van der Reest\textsuperscript{1}

\textsuperscript{1} IT Dept., DESY, Notkestrasse 85, Hamburg, Germany  
\textsuperscript{2} Gerd Behrmann, Copenhagen, Denmark  
\textsuperscript{3} Fermilab, Batavia, IL, USA  
E-mail: paul.millar@desy.de

Abstract. The availability of cheap, easy-to-use sync-and-share cloud services has split the scientific storage world into the traditional big data management systems and the very attractive sync-and-share services. With the former, the location of data is well understood while the latter is mostly operated in the Cloud, resulting in a rather complex legal situation.

Beside legal issues, those two worlds have little overlap in user authentication and access protocols. While traditional storage technologies, popular in HEP, are based on X.509, cloud services and sync-and-share software technologies are generally based on username/password authentication or mechanisms like SAML or Open ID Connect. Similarly, data access models offered by both are somewhat different, with sync-and-share services often using proprietary protocols.

As both approaches are very attractive, dCache.org developed a hybrid system, providing the best of both worlds. To avoid reinventing the wheel, dCache.org decided to embed another Open Source project: OwnCloud. This offers the required modern access capabilities but does not support the managed data functionality needed for large capacity data storage.

With this hybrid system, scientists can share files and synchronize their data with laptops or mobile devices as easy as with any other cloud storage service. On top of this, the same data can be accessed via established mechanisms, like GridFTP to serve the Globus Transfer Service or the WLCG FTS3 tool, or the data can be made available to worker nodes or HPC applications via a mounted filesystem. As dCache provides a flexible authentication module, the same user can access its storage via different authentication mechanisms; e.g., X.509 and SAML. Additionally, users can specify the desired quality of service or trigger media transitions as necessary, thus tuning data access latency to the planned access profile. Such features are a natural consequence of using dCache.

We will describe the design of the hybrid dCache/OwnCloud system, report on several months of operations experience running it at DESY, and elucidate the future road-map.

1. Introduction
The term “cloud storage” has come to have different meanings to different people; for example, a large file-system, a generic object-storage and a sync-and-share style service have all been deemed cloud storage. Perhaps the most common aspect is that a large (seemingly unlimited to the users) remote storage service is provided, often with a direct-charge business model that tries to ensure the sustainability of that service.

Storing large amounts of data is not a new concept to the high-energy particle physics (HEP) community, which has constantly strived against hardware limitations to store data. The result
is software, such as dCache[1], that combines the storage capacity of many computers, allowing sites to offer a scalable storage service both in terms of performance and capacity. This further allows sites to store both the HEP data produced by detectors and the result of physics analysis.

Despite this experience with large amounts of data, users found the services offered by sync-and-share cloud companies a useful addition to the services offered by sites. Synchronising files between desktop(s) and laptop(s), and easily sharing those files with colleagues provided a convenient and light-weight solution compared to more heavy-weight solutions that were officially supported by sites.

Allowing disconnect synchronisation (which does not require pairs of devices be available concurrently) means that files must be stored on some central service, which may be located in a different country from the user. Within a changing climate, it became desirable not to expose such files to external companies, especially those based in foreign countries.

For these reasons, DESY needed to provide a cloud sync-and-share service for its users—most prominently, for particle physicists and photon scientists—with the requirements of a similar level of service and easy replication between services, but with data stored on-site.

In this paper, we present this sync-and-share service, its design and implementation, the current status and planned future work.

2. Requirements
The most basic requirement of the DESY sync-and-share service derives from DESY users’ prior experience with external services. It must be “easy to use” or sufficiently similar to existing services that users can migrate to the new system with little or no support.

- It should support all the computing platforms DESY currently employs: Microsoft Windows-based desktops, Apple Macintosh- and Linux-based computers. There should also be support for mobile devices, such as those that have adopted the Android Operating System and iOS.
- All data must be stored at DESY, so users can be confident that their data has not been exported to third-parties for data-mining or other unauthorised activity.
- The service should integrate well with the existing DESY infrastructure; in particular, users should be able to authenticate with their DESY username and password.

Although not hard requirements, the following describes some additional use-cases that are anticipated. These are not required for the service to be useful, but are likely to be requested by users.

- The users may want some storage capacity that is not synchronised with their devices. Such files would be available for sharing with others without being synchronised on all devices. Files should be movable between synchronising and non-synchronising space.
- Providing different quality-of-service guarantees: archival storage for mission critical data, disk-only copy for bulk data, and low-latency storage using SSDs.
- Supporting the transfer of files to some remote service without the data travelling through the user’s computer.
- Direct access to storage files from computing facilities.

3. Design
The project evaluated different existing solutions and, while no single project satisfied all requirements, a combination of two projects satisfied the existing requirements and provided sufficient flexibility to satisfy the anticipated additional usage. These two projects are ownCloud[2] and dCache.
The ownCloud project is an open-source sync-and-share storage solution that is, at the time of writing, gaining popularity. In particular, it is the use of this service in other HEP laboratories that makes ownCloud particularly attractive. Although ownCloud provides users with a good front-end system, including web-based browsing and synchronisation clients for Microsoft Windows, Linux-based OSes, Mac OS X and mobile devices, it lacks support for managing data on a large scale. Instead, it assumes a site has an appropriate storage system mounted on the local filesystem.

The dCache project provides a powerful managed storage system, with good integration with scientific data life-cycle and automatic file migration between different media. It currently provides no sync-and-share facilities.

The DESY Cloud design combines both dCache and ownCloud. dCache’s support for standard protocols allows for easy integration with ownCloud; dCache is mounted via NFS v4.1[3] on the RHEL 7 servers hosting the ownCloud software. The ownCloud instances access files using the standard POSIX API.

In ownCloud, user authentication is based on username and password. The user-supplied credential is used to authenticate an LDAP query, where the LDAP server uses Kerberos interaction to verify the supplied credential (a “pass through” configuration). The LDAP query provides information about the user, which is used by ownCloud to establish whether or not the user is allowed to use the service.

At DESY, a proprietary system (called “the registry”) provides information about users. Various adapters (called “platform adapters”) allow custom integration of this information with the various services DESY provides. For the DESY Cloud service, there are two parts to the platform adapter: the registry platform adapter and the server-side platform adapter.

The registry platform adapter is located as part of the registry service. It provides information for the LDAP server and sends events to the server-side platform adapter when an account changes state. Such changes are part of that account’s life-cycle. Events include when an account is created, when it expires, when it is either enabled or disabled, and when an account should be archived.

The server-side platform adapter updates dCache in response to events from the registry platform adapter; for example, when it receives an account-created event, a suitable home directory is created for that user. Sufficient information is included in the event for the server-side platform adapter to operate correctly.

Several ownCloud instances are run concurrently on different machines. These form the individual services in a load-balanced pool, with client requests being directed to one machine. In addition to scaling the front-end service, this load-balancing allows routine maintenance of the ownCloud machines without service down-time.

4. Scientific Cloud Vision
The DESY Cloud service fits within the dCache “Scientific Cloud” vision. In this model, a single storage system provides storage capacity that may be used in different operational modes. For example, the same storage system might be used for fast ingest of data, a sync-and-share service, storage for HPC and HTC computing farms, and remote storage for moving data to remote sites. Figure 1 shows this diagrammatically.

Providing a single system brings many advantages. A single system reduces the cost of operating the system, because different services appear as different facets of the same underlying storage. The tight integration can provide a better end-user experience since there are no scripts copying files between systems. It can lead to better use of resources, as unnecessary redundancies can be removed.
5. Current status
The initial prototyping and deployment has been completed. A production infrastructure has been developed and is being rolled out within the DESY community.

At time of writing, the current set of users is small: only some 200 users have access to the service. They currently have around 2 million files stored on the service, consuming some 2.4 TiB of capacity.

The experience gained from this limited deployment has allowed the deployment team to discover problems both in ownCloud and dCache software. The team at DESY has been proactive in fixing these problems by writing patches and sending them to the corresponding project. These patches have been accepted in new releases of both dCache and ownCloud, in keeping with the desired goal of running unpatched software.

The experience gained from running the software and the DESY team’s alacrity in diagnosing and fixing problems as they are found, instill a high degree of confidence in the viability of this solution as we roll out the software to all members of DESY.

6. Development and future work
One future goal is to furnish direct access to the files a user has stored via DESY Cloud service. Such access would allow users to run analysis work based on analysis program files kept in sync with their work machines. It would also allow users to synchronise the output from their analysis work and to share those results with their colleagues.

To support this usage, the files written into dCache must have the user’s ownership. Normally, files written by an application have that application’s ownership; for example, all files written by the Apache daemon (that hosts the ownCloud service) are normally owned by user httpd, rather than the DESY user. It proved too difficult to fix this within ownCloud, so a simple work-around was introduced in dCache.

Another common use-case involves the bulk transfer of files between two sites, perhaps in different countries. Services exist[4][5] that support transferring scientific data reliably, optimising throughput and retrying transfers as necessary. By allowing access through the

---

Figure 1: The Scientific Cloud
protocols used in such services, the DESY Cloud service would allow users to migrate data as necessary.

dCache already allows files to be shared between users by specifying ACLs that permit an individual to read that file. Currently these are decoupled from the ownCloud sharing database: modifying a file’s ACL in dCache has no effect on whether the file is shared in ownCloud; likewise, sharing or unsharing a file in ownCloud has no effect on dCache’s ACLs. Future work will link these two, allowing consistent permissions whether the file is read through ownCloud or directly via dCache. That ownCloud and dCache use a common database technology (PostgreSQL) will greatly facilitate this integration.

Another issue is that of data integrity and preventing data corruption. The concept of preserving data integrity by computing a checksum is already integrated into dCache, with background checking (scrubbing) and automatic checking on integrity when transferring files. Such concepts currently do not exist within ownCloud. Working with the ownCloud developers, we will introduce such checks to ensure synchronisation is safe from corruption.

One design feature of dCache is that it will redirect the client to read or write data directly to the computer that stores (or will store) the request data. Such redirection allows dCache to scale to large storage capacity. Currently ownCloud does not support redirection and we will be working with ownCloud to add this support into their sync client.

One final point is with file change notification. Currently the ownCloud client polls for changes. Since most times the request reveals that nothing has changed, this introduces unnecessary load on the server. To reduce the impact of this load, the client checks every 30 seconds whether there have been any changes. However, this results in the client becoming slow to react to remote changes. Introducing a notification scheme would allow clients to react quickly while not placing too great a burden on the server.

7. Conclusion
We have presented the DESY Cloud service: a sync-and-share service in production at DESY using open-source software. While the project is still being rolled out and currently has a limited number of active users, our experience gives us confidence it will scale to support all DESY users.

The service is provided freely to DESY users, with no charge at point of use. While no decision has been made on how the final service will be funded, we anticipate this policy will continue for a base-line service. Fees might apply for additional features; e.g., archival storage using tape or low-latency access using SSDs.

The improvements and bug fixes that were developed while bringing this service into a production-like state and the goal of running unpatched software results in a service that other sites can replicate. For example, sites that already have a dCache instance can extend that service to include a sync-and-share service.

Acknowledgments
Work described in this paper was funded by the LSDMA project, DESY, Fermilab and NDGF.

References
[1] Fuhrmann P and Gültow V 2006 Euro-Par 2006 Parallel Processing (Springer) pp 1106–1113 doi:10.1007/11823285_116
[2] The ownCloud project website https://owncloud.org/ accessed: 2015-05-12
[3] Shepler S, Eisler M and Noveck D 2002 Network File System (NFS) version 4 minor version 1 protocol http://tools.ietf.org/html/rfc5661
[4] The fts v3 http://fts3-service.web.cern.ch/ accessed: 2015-05-12
[5] The globus transfer service https://www.globus.org/ accessed: 2015-05-12