

Cloud Services for Synchronisation and Sharing (CS3)

Book of Abstracts

30th January - 1st February 2017

SURFSara, Amsterdam

Workshop on Cloud Services for Synchronisation and Sharing (CS3)

January 30st – February 1st 2017
Amsterdam, The Netherlands

Keynote:

“Metadata at Dropbox: a look at dropbox’s transactional databases”

P. Boros, Dropbox

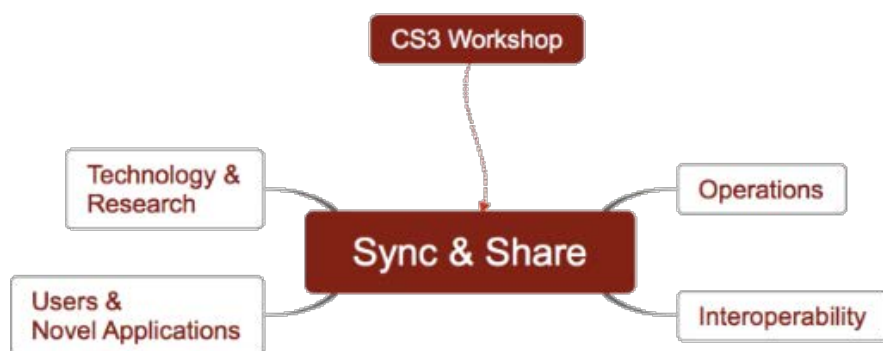
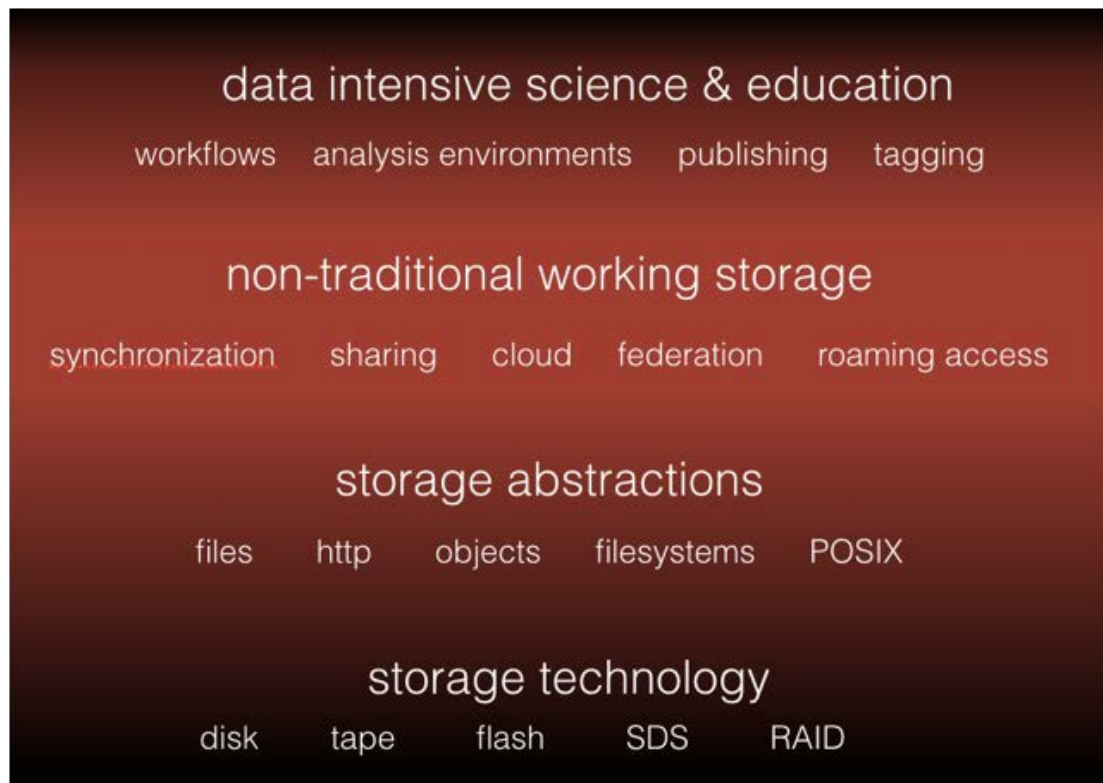


DOI 10.5281/zenodo.254064

<https://doi.org/10.5281/zenodo.254064>

Editor of Book of Abstracts: Belinda Chan (CERN)
Publication date: 2017-01-20

CS3 Workshop Landscape



Contents

Metadata at Dropbox: a look at dropbox's transactional databases	1
Boosting your IT infrastructure with SWAN.....	5
Web-based interactive analysis and visualisation of Earth Observation data at petabyte scale	5
Data Management System for Investigation of Heart Valve Diseases	6
File Syncing Technology Advancement in Seafire: Drive client and real-time backup server	9
Database clusters for sync and share services	10
Prototyping and testing HTTP/WebDAV synchronization protocol optimizations	10
Breaking the Monolith: An example of microservices for sync and share based on ClawIO.....	11
Next generation scalability	11
Scalable sync-and-share service with dCache	11
Scaling ownCloud beyond todays filecache.....	12
Evolution of CERNBox Sync&Share platform.....	12
Storage backends for scalable sync&share service based on Seafire	13
Towards smart file-based data stores	14
Automated error handling at Dropbox	15
Schrodinger's cloud storage - Defining the box used	15
System testing cloud services using Docker and Kubernetes: EOS+Cern Tape Archive (CTA) development use-case	16
Onedata - Eventually Consistent Virtual Filesystem for Multi-Cloud Infrastructures	19
Object storage and innovation as building blocks for sync & share applications	20
Open object storage as the foundation for open sync and share in Science	20
EOS - the CERN disk storage system driving CERNBox	21
IBMs Elastic Storage Server in Synch and Share environments.....	21
Manila, Sahara and Cloud integration in Openstack using NetApp NFS datastores.....	21
Enabling File sync and sharing in a multi domain user environment with OwnCloud and NetApp StorageGrid	22
SURFsarasitereport	25
Change Management : Seafire in University of Strasbourg	25
Cybera Site Report.....	26
DESY site report	26

Keeping university researchers from using Dropbox	27
Storage Operations at CERN: CERNBox	27
Keeper – Archive the way you work.....	28
HU Box – Cloud Storage. Synchronised data storage and file sharing for university use cases at the Computer and Media Service of the Humboldt-Universität zu Berlin (HU).....	29
B2DROP: The EUDAT Personal Cloud Storage.....	30
Polybox: Mobile working Use Case enabled by Sync & Share at ETH Zürich	31
Experience with running Owncloud on virtualized infrastructure (Openstack/Ceph) (continuation of talk at C3 2016)	31
A Staging Storage Sharing System for Data Handling in a Multi-site Scientific Organization	32
The good, the bad and the ugly: Experiences and learnings from years at the forefront of EFSS	34
Next steps in file sync and share security.....	34
Improving Pydio performances with PydioBooster.....	34
Seafire project: status and outlook	35
Next generation federated sharing	35
deutsche.hochschul.cloud – An initiative to promote the networking of German universities and research institutions	36
OCM	41
Federated sharing with the Open Cloud Mesh API.....	41
OwnCloud scalability and concept design	45
Scalable management with Seafire SPORE specification	45
ALLVIS, Uppsala Universitys official storage of scientific Data	47

Conference Keynote

Metadata at Dropbox: a look at dropbox's transactional databases

P.Boros¹

¹ *Dropbox*

Contact Details: pboros@dropbox.com

500 million people around the world use Dropbox to work the way they want, on any device, wherever they go. With 200,000 businesses on Dropbox Business, we are transforming everyday workflows of entire industries. In this keynote session, Peter Boros from Dropbox's databases team will give us a peek to the challenges they are facing to handle the metadata of 1,200,000,000 uploaded files every single day. Because of the sheer volume and growth, the team has to use automation, and think about optimizations all day, every day. We will discuss our design decisions over time, the improvements we made and their impact, and what do we have in the making.

Application & Users session

Boosting your IT infrastructure with SWAN

Danilo Piparo¹

¹ CERN

Contact Details: danilo.piparo@cern.ch

Since mid-2016 CERN offers to scientists a service for interactive web based data analysis: once logged in, a scientist can start analysing data without the need of any setup nor configuration. Prominent features of the service are the availability of a complete scientific software ecosystem, the possibility to synchronise the user space from the cloud to any device and vice versa and the tight integration with other widely adopted IT services such as mass and synchronised storage, batch system, software distribution vectors. In this contribution we review the usage patterns of the SWAN service adopted by CERN scientists and engineers coming from different backgrounds such as beam and high energy particle physics, accelerator technologies or IT. Concrete examples will be shown and discussed. Lessons learned and solutions adopted based on users' feedback are then discussed with the goal of highlighting what features of a SWAN-like service are most desirable in a big research laboratory.

Web-based interactive analysis and visualisation of Earth Observation data at petabyte scale

Armin Burger¹ ; Pierre Soille¹

¹ Joint Research Centre, European Commission

Contact Details: armin.burger@jrc.ec.europa.eu, pierre.soille@jrc.ec.europa.eu

The Copernicus programme of the European Union with its fleet of Sentinel satellites for monitoring land, ocean, and atmosphere with applications from environment monitoring to emergency response is generating Terabytes of free and open data on a daily basis. The Joint Research Centre (JRC) of the European Commission has developed a prototype Joint Earth Observation Data and Processing platform (JEODPP) to enable its knowledge production units to process and analyse global geospatial data at Petabyte scale in support to EU policy needs. In the framework of a collaboration between CERN and JRC, the EOS distributed file system enables high data throughput between the processing nodes and the storage servers. The performance of the JEODPP is analysed on use-cases in the context of interactive and batch processing based on docker containerisation and managed by the HTCondor workload manager. Web-based interactive analysis and visualisation of the Copernicus data on the EOS repository is obtained via Jupyter Notebooks connected to distributed backend processing servers. The process distribution and real-time visualisation of the results on interactive maps is achieved via custom interactive widget like ipyleaflet used in the Jupyter notebooks. This way the data analysis capability of the JEODPP can be shared with internal or remote user groups.

Data Management System for Investigation of Heart Valve Diseases

Author(s): Marian Bubak¹

Co-author(s): Daniel Harezlak²

¹ AGH University of Science and Technology, Krakow, PL

² ACC Cyfronet AGH

Contact Details: marian.bubak@gmail.com

The main goal of investigations in the framework of the the EurValve project [1] is to combine a set of complex modeling tools to deliver a workflow which will permit the evaluation of medical prospects and outlook for individual patients presenting with cardiovascular symptoms suggesting valvular heart disease. It should result in providing a decision support system which can be applied in clinical practice and require a dedicated problem solving environment of which a key component is a data management system. The nature of the tasks performed in the research environment includes both interactive and batch processes, some of which need to be invoked manually, whilst others can and should be automated. The entry point to the EurValve research environment is a portal integrated with external services and infrastructures. The File Store is a remotely accessible service that provides a user overlay for the secure storage components. It enables users to access, upload and share folders and files pertinent to EurValve. Externally, it mimics a standard WebDAV server and can thus be accessed by any WebDAV-compliant library or standalone clients [2]. In addition to the management of file based data, a requirement for the EurValve project is to manage structured medical data collected from each clinical center. This is achieved by provisioning the data into a web accessible data node developed within the VPH-Share project. Within this node all data is hosted within a MySQL database and a middleware layer exposes the data through a variety of protocols. These include SOAP, via a documented XML query structure, REST using a JSON query document, REST using SPARQL and REST using direct SQL. A final data access mechanism is provided through a graphical web based interface allowing both exploration and query of the data. All of these channels are secured and allow both read only and read write access to each of the data sets acquired by the project. To achieve this, the data management system is integrated with the EurValve security mechanisms which consist of an Identity Provider capable of validating that users are actually who they claim they are, Security Web Platform, which includes an IdP assertion consumer, a JSON Web Token issuer, a Policy Decision Point and a Policy Retrieval Point.

Technology session

File Syncing Technology Advancement in Seafile: Drive client and real-time backup server

Jonathan Xu¹

¹SeafileLtd.

Contact Details: jonathan.xu@seafile.com

We'll present two recent innovative file syncing technology in Seafile: the Drive client and real-time backup server.

Seafile Drive Client

First introduced by Dropbox in around 2007, file syncing has become a more and more common technology in the last few years. Services like Dropbox, OneDrive, Google Drive are more or less similar to each other: syncing/replicating files across users' computers. However, we believe there is another innovative and useful way to access files in cloud storage. Cloud storage can be mapped into user's computer as a virtual hard drive, without syncing files to client computer. Seafile Drive client is designed for this usage mode.

There are two main advantages for the Drive client:

- Users can use cloud storage like a local drive on Windows. It's the most intuitive and familiar way for most users.
- Local disk space can be freed up because users don't need to replicate files in the cloud to local disk.

With the Drive client, organizations can use Seafile to replace Samba/Windows Share. Users can access files in Seafile server just like accessing a Windows network drive. Seafile Drive client also has one advantage over Windows Share: files are cached in local disk. When users go offline, they can still access cached files.

The Drive client also opens up novel application in scientific research data management. Large volume of experimental data can be written directly to the cloud, through the Drive client.

Real-time backup server

We'll also give an introduction to the real-time backup feature. Data can be backup in a almost real-time manner, from one Seafile server to another Seafile server. Full history data is also backed up. Compared to traditional daily backup, this greatly reduce the backup window. It can also be used as multi-site replication mechanism, to provide higher availability for Seafile service.

Database clusters for sync and share services

Author(s): Holger Angenent¹

Co-author(s): Andreas Wilmer¹

¹ University of Muenster

Contact Details: holger.angenent@uni-muenster.de

Sync and share services like ownCloud often rely on database backends for storing metadata. Those databases should offer a high availability and performance. With a clustered database, both of these requirements can be fulfilled.

One method of running a database cluster is a master-master Galera replication. In real life, the database performs more robust, if write requests are sent to a single database node and read requests are distributed among all other nodes.

In such a setup we expect a read-only workload to scale linearly with the number of nodes. The drawback: writes have to be replicated to all participating nodes. Thus, a scalable performance can only be expected if the rate of modification is not too high. The scalability of the performance is therefore limited to low writes versus reads ratios.

This talk will address the question of how the performance of a database cluster scales with the number of nodes. We will investigate typical sync and share-workloads as well as more pathologic writes-reads-ratios. Additionally, the influence of the speed of the network interconnect will be measured. Finally, the quantitative difference between a single database server and a clustered solution is presented.

Prototyping and testing HTTP/WebDAV synchronization protocol optimizations

Author(s): Piotr MRÓWCZYŃSKI¹

Co-author(s): Jakub Moscicki²; Thomas Müller³

¹ ownCloud, TU Berlin / KTH Royal Institute of Technology

² CERN

³ ownCloud

Contact Details: piotr.mrowczynski@yahoo.com

Research of performance of cloud synchronization services like ownCloud, Seafile and Dropbox has shown, that on-premise services show better performance characteristics than public clouds syncing big files (higher transfer rates in both upload and download could be obtained due to simple implementation and smaller activity of users for specific bandwidth) and are very competitive syncing mixtures of files.

Unlike typical web services, cloud sync and share is characterized by requests load/number much outreaching the typical loads to the web server per user in some specific activity cases. Underutilized upload/download bandwidth and long distribution tails (penalizing transfers of small files over WAN) are characteristic for services using current ownCloud synchronization protocol. Important factor in synchronization performance is also number of operations performed per single-file request on the web-server. In some specific mime-type cases, the complete file has to be uploaded or downloaded because of a small change in the file body, consuming server resources.

In the contribution, the tests of sync optimizations prototypes – requests bundling, http/2, delta-sync and requests scheduling - addressing the above issues, will be presented.

Technology

Breaking the Monolith: An example of microservices for sync and share based on ClawIO

Hugo Gonzalez

Contact Details: cs3testing@hugo.labkode.com

A study of using a distributed microservices architecture for synchronization and sharing

Technology

Next generation scalability

Frank Karlitschek

Contact Details: karlitschek@gmail.com

One of the main challenges of on premise file sync and share solutions is scalability. It is essential that solutions scale from small to very big installations with millions of users and petabytes of files. This talk will present the current approaches to scale a system including a case study how to scale to millions of users. It also presents a new approaches to bring the scalability of on premise file sync and share solutions to the next level. Part of the talk will be the presentation of a new architecture that enables close to unlimited scaling of a Nextcloud instance.

Presenter: Frank Karlitschek Frank founded the ownCloud project in 2010 to put home users and enterprises back in control of their data. To improve the company-community balance and accelerate the project he founded Nextcloud in 2016 and has been tirelessly working to realize his vision ever since.

Technology

Scalable sync-and-share service with dCache

Patrick Fuhrmann¹ ; Paul Millar¹ ; Tigran Mkrtchyan¹

¹ DESY

Contact Details: paul.millar@desy.de

Scientific exploration and exploitation of data is undergoing a revolution as communities explore new ways of analysing their data. One solution that is being used increasingly is sync-and-share, where data, presentations, graphs and code are shared in an ad hoc fashion. This allows communities to explore data in new and innovative ways.

Sites that have already invested in dCache to solve their large data storage requirements are keen that their services integrate seamlessly with sync-and-share systems. Such combined systems should function without any degradation of either system.

The combined system, involving ownCloud and dCache, is also attractive for sites that have not yet invested in dCache as they will want a solution that scales as well as dCache.

Naturally, dCache has many years of experience handling multiple petabytes of data: we know how to scale such data services. Although large data is problematic for (own|next)Cloud, the own-Cloud+dCache deployment at DESY ("DESY Cloud") shows such a deployment is practical.

Although a simple deployment functions well enough, we see opportunities to improve the performance; in particular, there is the possibility to improve the interface between (own|next)Cloud and dCache; to avoid duplication of information and avoid potential bottlenecks.

We will present a summary of the ownCloud+dCache hybrid system, identifying problem areas and present our solutions to solve these issues.

Technology

Scaling ownCloud beyond today's filecache

Jörn Dreyer¹

¹ ownCloud GmbH

Contact Details: jfd@owncloud.com

ownCloud uses the filecache table to propagate filechanges in the file hierarchy. Under heavy usage this causes the table to become a bottleneck because multiple UPDATES might wait for a lock on the same tuple. By storing the metadata directly in a filesystem's extended attributes and ACLs we can completely get rid of the filecache table. Using existing filesystem capabilities we can scale ownCloud beyond what is currently possible.

Technology

Evolution of CERNBox Sync&Share platform

Jakub Moscicki¹

¹ CERN

Contact Details: jakub.moscicki@cern.ch

We started CERNBox in 2013 as a small prototype based on a simple NFS storage and one of the initial versions of the Owncloud server. Some 3 months and 300 users later we have had enough of enthusiastic feedback to consider to open the sync&share service at CERN. Since then we witnessed a rapidly growing service in terms of number of accounts, files, transfers and daily accesses. At the same time we have been evolving the architecture of CERNBox in order to cope with new requirements, well beyond traditional sync&share services which are usually focused on office documents. This included not only the increasing performance expectations but also integration of the sync&share capabilities into diverse daily workflows of CERN users: from desktop applications and home directories to scientific data analysis.

Current CERNBox architecture integrates very closely with the EOS backend storage with a built-in support for HTTP-based synchronisation protocol used by Owncloud synchronisation clients. This allows to harmoniously integrate the native EOS storage access, such as filesystem, with the synchronisation layer. In this model the storage is exposed to the end users for direct access and thus it is not solely controlled by the sync and share layer. We have also been evolving the Owncloud web server to take into account such architectural changes.

In this presentation we will describe further evolution of CERNBox. Implementation of sharing directly on the storage, using EOS native access control mechanisms and metadata propagation features, is the next logical step to provide improved user experience. For the internal architecture we investigate a model based on micro-services to get more flexibility to evolve and improve individual functional subsystems on a longer run. To name just a few examples, we consider evolution of the synchronisation protocol including file metadata synchronisation and efficiency improvements, especially for high-latency, low bandwidth and unreliable network connections. For web fronted we are revisiting the handling of metadata and pre-processed files, such as image preview, in a large-scale storage environment. The growing scale of operations is calling for efficient methods to detect and debug user problems remotely.

Storage backends for scalable sync&share service based on Seafile

Author(s): Maciej Brzezniak¹

Co-author(s): Krzysztof Wadowka¹

¹ *PSNC Poznan Poland*

Contact Details: maciekb@man.poznan.pl

Seafile is a scalable and reliable sync&share solution. Its synchronisation engine and data model is based on git concept adapted to dealing with large files and datasets. Seafile synchronises data based on filespace snapshots rather than per-file or per-data object versioning and involves deduplication with Content Defined Chunking algorithm. The architecture and implementation introduces small overheads as the relational database usage is reduced to minimum - only head commit ID and user-library mappings are kept there, while the actual data and meta-data are handled by the storage back-end.

Well-optimised synchronisation engine of Seafile has a potential to put a lot of stress on the storage back-end while serving a large number of I/O operations. In fact it constitutes an interesting killer application for the storage system.

Seafile deployment at PSNC targets a country-wide scale, therefore we expect to deal with large user base as well as millions of files and I/Os to be served on time. While Seafile supports various storage back-ends including filesystem and object storage as well as enables Load-Balancing and High Availability for the synchronisation engine, the decisions on choosing and configuring storage back-end for the planned scale are not trivial.

In our presentation we will overview and summarize the I/O requirements of Seafile server as well as analyse several storage systems in this context, including traditional and clustered filesystems based on Fibre Channel disk arrays as well software defined storage systems based on disk servers and 10Gbit Ethernet.

We will share the results of our analysis and benchmarks performed with Seafile server as well as draw out general conclusions and lessons learnt on architecting storage back-ends for sync & share services.

Towards smart file-based data stores

Author(s): Reginald Cushing¹

Co-author(s): Adam Belloum¹; Marian Bubak²; Spiros Koulouzis¹

¹ *University of Amsterdam*

² *AGH Krakow*

Contact Details: r.s.cushing@uva.nl, s.koulouzis@uva.nl

Nowadays, due to the data deluge and the need for the high availability of data, online file-based data stores have gained an unprecedented role in facilitating data storage, backup and sharing [1]. Up to date, the role of these file storage systems has been, largely, passive i.e. they host files and serve files to clients upon request. The simplistic approach of these file data stores means that they are easily deployed and integrate into other applications but may have limitations when hosted files are part of larger distributed data-oriented computations. First, data locality plays a crucial role on the performance of a data-oriented application, file servers may be too far from the computation or may have unreliable network between computation and data which will introduce bottlenecks and overhead in the running application. Second, larger computations tend to produce many intermediate result files which can easily inundate a file data store either from capacity or network limitations. Our proposed approach tackles these two points by proposing a hybrid data-compute store where data stores can have a limited role in computing thus bringing together computation and data; this is extension of the concept presented in [2] and [3]. The main concept of our solution is that, in many scientific applications, data and computation are tightly coupled thus it makes sense to store the functions alongside the data in a unified database. One simple example is transcoding of images e.g. two same image files with different resolution. By capturing this information at the data store as part of the file metadata we can introduce some optimization routines. Instead of storing multiple images at different resolutions we can store one raw image and a set of transcoder functions that get called by the database when a particular image with a resolution is requested. The implication of mixing functions and data together means that datastores can prioritize on storage space by, safely, removing data which can be regenerated from the stored functions. As one can imagine this concept can be extended to larger computations such as work where one file is subsequently transformed into many other files which are all linked together through workflow functions.

References [1] S. Koulouzis, A. Belloum, M. Bubak, P. Lamata, D. Nolte, D. Vasyunin, C. de Laat, Distributed Data Management Service for VPH Applications, IEEE Internet Computing 20 (2), 34-4, 2016

[2] R. Cushing, M. Bubak, A. Belloum, C. de Laat, Beyond scientific workflows: Networked open processes, IEEE 9th International Conference on eScience, 357-364, 2013

[3] R. Cushing, A. Belloum, M. Bubak, C. de Laat, Towards a data processing plane: An automata-based distributed dynamic data processing model, Future Generation Computer Systems 59, 21-32, 2016

Technology

Automated error handling at Dropbox

Maxim Bubliss

Contact Details: satori@dropbox.com

At Dropbox, with 1000s of MySQL servers, failures like hardware errors are normal, not exceptional. There is no day passing by without replacing at least 1 server with some kind of hardware error. Our on-call engineers are not alerted for these, they are alerted if the automation is not working properly. This kind of automation is harder with stateful systems, so we wrote a general framework for that called Wheelhouse. In this framework, state machines are describing the good states of systems, and the transition steps between them. In this talk we will show the following:

- What happens with a slave in case of hardware error
- What happens with a master in case of hardware error
- What happens when we would like to upgrade kernels
- How are we using this framework to coordinate schema changes between shards
- How are we using this framework to verify data consistency

Technology

Schrodinger's cloud storage - Defining the box used

David Jericho¹

¹ AARNet

Contact Details: david.jericho@aarnet.edu.au

Experiences of containerising a traditional software stack

Synopsis

This presentation will be about the experiences of AARNet of converting what is a traditional mono- lithic software stack to run inside a fully containerised and dynamically provisioned Docker based container system.

The entire stack, from the front end TLS proxies to the backend scale out storage, the metrics, reporting and orchestration all run inside containers.

History

AARNet has like many other NRENs, deployed a sync and share platform built on both ownCloud and FileSender, and it has been adopted at scale by our users.

AARNet is in a somewhat unique position amongst NRENs, having a few clusters of dense population on a very large continent with tens of milliseconds between cities. This has meant that AARNet's software infrastructure needs to be spread in order to minimise network latencies between nodes, and this has created its own challenges in orchestrating and managing the environments.

Problem Scope

Growth of users and the type of usages has resulted in a substantial increase in the amount of hardware required to provide a reliable responsive service, and this is multiplied by 3 due to the 3 concurrent sites. The automatic connecting to the nearest node is managed by BGP Anycast, which means a user doesn't live at any one site, but all three simultaneously. Many of the research groups in Australia are geographically distant too, in some cases up to 90ms apart by network paths, yet they all want seamless and consistent performance experiences.

Managing the hardware and software required for this means keeping service state and versioning, to which using containers as a packaging format has become an obvious solution.

Solution

Containerisation through Docker, orchestrated via Rancher, has resulted in a stable and scalable software stack that allows tiered definitions of a software stack and all its constituent components. Ansible and Cobbler to deploy simplest-possible server environments, followed by automatic deployment onto compute resources as they become available has resulted in software upgrade times in the order of seconds with a single click of a mouse button, and instant rollback capability. The actual software is ephemeral and doesn't live permanently on any one server as an assigned task.

Due to the ephemeral nature of the environment, work has been put into ensuring logs and metrics are all centrally collected and visible, with servers themselves treated as entirely disposable.

By being entirely ephemeral, secret management also comes to the fore in the minds of developers, so security of hidden components is increased.

Additionally due to the advanced networking capabilities of Docker and Rancher, scaling out onto third-party clouds is a trivial activity.

Problems

The largest problem with such an infrastructure, strangely is also one of its strengths. The ephemeral nature of the software stacks, and the minimal tooling inside a container means that there can be extra steps or repeated steps to debug issues.

Another problem is advocating and modifying the thinking of developers to understand the benefits and changes in approach they have to adopt.

Summary

Containerising the software hasn't been an entirely smooth process, but this has been related to the software stack deployed and developer education, rather than a fault of the model. The benefits have far outweighed the difficulties, and the view is now that the task has been done there is no intention to revert the management model. Security has been increased, reproducibility and idempotency is much improved, and speed of deploying both updates and new applications has been massively improved.

Technology

System testing cloud services using Docker and Kubernetes: EOS+Cern Tape Archive (CTA) development use-case

Author(s): Julien Leduc¹

Co-author(s): Steven Murray¹

¹ CERN

Contact Details: julien.leduc@cern.ch

The IT Storage group at CERN develops the software responsible for archiving to tape the custodial copy of the physics data generated by the LHC experiments. This software is code named CTA (the CERN Tape Archive). It needs to be seamlessly integrated with EOS, which has become the de facto disk storage system provided by the IT Storage group for physics data.

CTA and EOS integration requires parallel development of features in both software that needs to be synchronized and systematically tested on a specific distributed development infrastructure for each commit in the code base.

This presentation describes the full continuous integration work flow that deploys and orchestrates all the needed services in docker containers on our specific kubernetes infrastructure.

Storage technologies session

Onedata - Eventually Consistent Virtual Filesystem for Multi-Cloud Infrastructures

Author(s): Łukasz Dutka¹

Co-author(s): Bartosz Kryza¹

¹ AGH University of Science and Technology, Academic Computer Centre Cyfronet AGH, Krakow, Poland

Contact Details: Łukasz Dutka (lukasz.dutka@cyfronet.pl)

Onedata [[1]] is a global high-performance data management system, that provides easy and unified access to globally distributed storage resources and supports wide range of use cases from personal data management to data-intensive scientific computations. Due to its fully distributed architecture, Onedata enables creation of complex hybrid-cloud infrastructure deployments, including private and commercial cloud resources. It allows users to share, collaborate and publish data as well as perform high performance computations on distributed data.

Onedata system comprises zones (Onezone) which enable establishment of federations of data centers and users, storage providers (Oneprovider) who expose storage resources and clients (Oneclient), who can access their data via a virtual POSIX file system. Onedata manages all operations on files at the level of variable sized blocks, ensuring highly efficient data access to files available remotely and giving the users an eventually consistent view of the filesystem from anywhere. In order to efficiently propagate local changes to other storage providers, who support specific user spaces, we employ tree propagation algorithm, which means that each storage provider sends out the local modifications events only to a subset of all providers who can be affected by this change. Onedata introduces the concept of space, a virtual volume, owned by one or more users, where the data is stored. Each space can be supported by a dedicated amount of storage supplied by one or multiple storage providers. Storage providers deploy Oneprovider instance near the storage resources, register it in selected Onezone service to become part of a federation and expose those resources to users. By supporting multiple types of storage backends, such as such as POSIX, S3, Ceph and OpenStack Swift, Onedata can serve as a unified virtual filesystem for multi-cloud environments.

For flexible collaboration and data sharing, Onedata provides fine-grained management of access rights, including POSIX-like access permissions and access control lists (ACLs), that allow users to share entire spaces, directories or files with individual users or user groups. Onedata allows integration with several identity providers, by means of OpenID Connect protocol, enabling users to login using their existing accounts, while all authorization decisions within Onedata are based on bearer tokens (Macaroons) generated by Onezone service.

Currently Onedata is used in INDIGO-DataCloud [[2]] project as a federated data access solution, aggregating computing centres and infrastructures; and in EGI-Engage [[3]], as the basis of EGI Open Data Platform, supporting various open science use cases such as open data curation (metadata editing), publishing (DOI registration) and discovery (OAI-PMH protocol).

Acknowledgements: This work has been partially funded under Horizon 2020 EU projects: INDIGO-DataCloud (Project ID: 653549) and EGI-Engage (Project ID: 654142).

REFERENCES

Onedata project website.

INDIGO-DataCloud (Integrating Distributed data Infrastructures for Global Exploitation).

EGI-Engage (Engaging the Research Community towards an Open Science Commons).

Object storage and innovation as building blocks for sync & share applications

Author(s): Andrea Marchi ¹

¹ Cynny Space

Contact Details: a.marchi@cynnyspace.com

Cynny Space, one of the few 100% European cloud object storage providers, engineered an innovative object storage platform with a unique hardware/software infrastructure designed to meet the rising need of storage with an unprecedented level of efficiency.

Most innovative storage platform The building block of the innovation is the smallest micro- server in the world, the first built with an ARM® CPU designed for efficient data management. Each micro-server is independent and directly connected to the internet and manages a single storage unit. 500 nodes (micro-servers and storage unit) are inserted in a rack and work in parallel to provide a high-density real-time archiving system. Each micro-server is as smart as the others thanks to designed swarm intelligence methodology distributed across all the 500 nodes. There are no single points of failure and the system takes into consideration hardware fallibility.

Real world benefits These innovations deliver several advantages compared to traditional storage solutions. ARM CPUs provide remarkable energy efficiency and lower heat generation. The overall result is a positive impact on the environment, reducing CO2 emissions by 74,8Kg per year for each TB stored, and lower costs for the storage user (0,01€/GB/month – space, redundancies, request and transfers included). Another benefit is the lack of maintenance. The high redundancy and distributed file system allows the servers to run with close-to-zero levels of maintenance and close to perfect durability levels (99,999999999%) Moreover, all data are stored in Europe and compliant to the EU regulation on data management.

Key applications of Object Storage Main application of Object storage is to store large quantities of information, making it available across users or devices with simplified interfaces. Interconnectability across multiple devices, from servers to IoT, is assured by the access via RESTful APIs and a comprehensive set of SDKs. Versioning of files ensures long lasting consistency of data thanks to its intrinsic quality of making files immutable.

Open object storage as the foundation for open sync and share in Science

Author(s): Simon Traill ¹

Co-author(s): Mario Blandini¹

¹ SwiftStack, Inc.

Contact Details: strail@swiftstack.com, mario@swiftstack.com

Like many of the open-source sync and share software stacks, OpenStack Swift is an open source engine. Object Storage is an ideal architecture for storing unstructured data. SwiftStack is a software product built-on OpenStack Swift that can be deployed on standard server hardware to build a durable object storage cluster. It offers OpenStack Swift and AWS S3 API support, and is used by many File Sync and Share software stacks both open and commercial. This session will cover how the architecture of OpenStack Swift is optimized as a repository backing file sync and share, share common deployments examples and specific customer examples with different software stacks, and provide information management best practices for OpenStack Swift storage.

Storage technologies

EOS - the CERN disk storage system driving CERNBox

Andreas Joachim Peters¹

¹ CERN

Contact Details: andreas.joachim.peters@cern.ch

EOS is a disk-based storage system providing high-capacity and low-latency access for users at CERN. It is the online storage system for all LHC and most non-LHC experiments at CERN.

Today EOS provides over 160 PB of raw disk space. The software is developed since 2010 at CERN and available under GPL license. EOS drives CERNBox as back-end storage system and provides sync-and-share facilities to CERN users. It provides multiple views/protocols to the same namespace and storage backend - via the OwnCloud synchronization client, as a mounted filesystem or latency optimized wide-area file access protocols.

The presentation will introduce core features of EOS and highlight the current development status and future roadmap.

Storage technologies

IBMs Elastic Storage Server in Synch and Share environments

Olaf Weiser¹

¹ IBM

Contact Details: olaf.weiser@de.ibm.com

The objective of this technical oriented presentation is to share experiences of IBMs Elastic Storage Server in Synch & Share Environments & to give insides of how ESS specific functionalities such as GNR/GPFS Native Raid will have a very positive impact on parallel workloads. If you want to understand how Block-Storage can be integrated into a FileSystem Storage Environment , how to improve the organisation of MetaData, what the RDMAValue Prop for GRID Cluster Environments is and if you finally want to find out the comparison of TCP/IP vs Infiniband RDMA...

Storage technologies

Manila, Sahara and Cloud integration in Openstack using NetApp NFS datastores

Author(s): Tom Stuijt¹

Co-author(s): Clemens Siebler¹

¹ NetApp

Contact Details: stuijt@netapp.com

The OpenStack Shared File Systems project (Manila) provides basic provisioning and management of file shares to users and services in an OpenStack cloud. The OpenStack Data Processing project (Sahara) provides a framework for exposing big data services, such as Spark and Hadoop, within an OpenStack cloud. Natural synergy and popular demand led the two project teams to develop a joint solution that exposes Manila file shares within the Sahara construct to solve real-world big data challenges. This contribution examines common workflows for how a Sahara user can access big data that resides in Hadoop, Swift, and Manila NFS shares. It also demonstrates how you can replicate Manila fileshares on-prem (in your private OpenStack cloud) to public clouds (such as AWS, or Azure),

Storage technologies

Enabling File sync and sharing in a multi domain user environment with OwnCloud and NetApp StorageGrid

Author(s): Tom Stuijt¹

Co-author(s): Clemens Siebler¹

¹ *NetApp*

Contact Details: clemens.siebler@netapp.com

NetApp StorageGrid delivers a multisite, multi-tenant, policy based, software defined S3 object store. Combined with the OwnCloud file sharing capabilities we propose a solution for multi-domain file syncing and sharing.

Services & Site Reports session

SURFsara site report

Ron Trompert

Contact Details: ron@sara.nl

In this presentation we will discuss the developments of the past year related to sync-and-share services and other SURFsara data services. We will also discuss the issues we have encountered and our successes.

Change Management : Seafile in University of Strasbourg

Author(s): Simon Piquard¹

Co-author(s): Eric Laemmer¹

¹ *University of Strasbourg*

Contact Details: simon.piquard@unistra.fr

Working in a University, in a diverse and dynamic community, makes it possible to observe the emergence of new digital uses. Dropbox and similar applications were the pioneers of file synchronization and sharing softwares, which use was largely utilized for research teams and teaching.

Indeed, the cloud storage is a very powerful system to develop collaborative work and increase autonomy. Moreover, this kind of tools is easily accessible for all users, thus regardless of the user computer skills.

However, external storage services are pointed out to be a potential security threat and a privacy issue. Furthermore, most of these services do not provide a data loss backup service.

That's why; we deployed Seafile as a cloud storage service at the University of Strasbourg for **employees, researchers and teachers.**

We started in May 2015 with 150 users, which some of them were identified as previously Dropbox users. Since then, in November 2016, we have more than 1000 different Seafile users per month.

In this paper, we focused on describing dedicated effort made on documentation, usability and training provided to heterogeneous panel of users: such as employees, researchers and teachers, experts or novices.

In the first part, we present and detail the change management methods defined at the Strasbourg University. This includes:

Trainings: How we organized an experimental phase before deployment with advanced users: feedback, identification of levels and trainings strategy.

Communication: How we present the tool and which channels we chosen: meetings with management and users, websites, mailing, print etc.

Documentation: How we wrote documentation and why it is important for some users

Feedback and User community: How we drive the community: use or satisfaction surveys, special contacts and educational tours.

In the second part, we present Seafile service evolution at Strasbourg University. The service is a great success. We believe that its use will continue to grow, especially with the arrival of SeaDrive and the opening of the service for students.

Thereby, we plan to set-up a new tool to drive a huge user community. We are currently working on a discussion forum based on Discourse. Seafile will be one the most important item. We hope this forum will facilitate exchanges between users, and help us to have interesting feedback.

For documentation, we would like to create some Seafile “How to” with French and German universities. We want to collaborate for the creation of multi-language adapted resources, e.g.:

How to begin with Seafile?

How to improve my use of Seafile?.

Finally, we will create video tutorials.

References (french):

Online documentation:

<https://services-numeriques.unistra.fr/documentations/toutes-les-documentations/outils-collaboratifs/synchro-et-partage-de-fichiers.html>

Satisfaction survey: <https://seafile.unistra.fr/f/9f90cf5572/>

Example of press release with one of our partner:

<http://www.lactu.unistra.fr/index.php?id=22479#c103028>

Services & Site Reports

Cybera Site Report

Joe Topjian

Contact Details: joe.topjian@cybera.ca

This short talk will give the attendees an update on what Cybera has been doing with storage technologies since the last CS3 conference.

I will quickly highlight three main areas:

Cybera recently migrated our OpenStack Block Storage Service (Cinder) from GlusterFS to LVM. I'll talk about the reasons why we did this as well as how this benefits users (spoiler: opt-in high availability as well as opt-in encrypted storage).

Speaking of encryption, OpenStack's Object Storage Service (Swift) has recently announced support for encrypted objects at rest. We have plans to implement this.

Finally, I will discuss a novel way in which we plan on offering ownCloud to our members and users later this year (maybe even by the time CS3 starts): as an application in the OpenStack Application Catalog.

Services & Site Reports

DESY site report

Author(s): Peter van der Reest¹

Co-author(s): Patrick Fuhrmann¹; Quirin Buchholz¹; Tigran Mkrtchyan¹

¹ DESY

Contact Details: peter.van.der.reest@desy.de

In this report we will discuss the current setup of cloud storage infrastructure at DESY. We will look at the experiences of the last two years, user expectations and service development. Lastly we will present our thoughts on future development of DESYcloud(TM) storage.

http://it.desy.de/services/storage_services/desycloud/index_eng.html

Keeping university researchers from using Dropbox

Moritz Schlarb¹ ; Thomas Scheffczyk¹

¹ Johannes Gutenberg-Universität Mainz

Contact Details: schlarbm@uni-mainz.de, thomas.scheffczyk@uni-mainz.de

Users have grown accustomed to easy-to-use sync-and-share applications like Dropbox, Google Drive and many more, which tightly integrate into their operating systems and platforms. Naturally, an obvious need arises to use these tools at the workplace, as well. However, there are strong arguments against embracing such public cloud services, especially in the context of scientific and technical research or even personal data, where special data protection laws apply. IT staffs at universities are challenged to provide solutions for syncing and sharing of data that work in a comparable way to the aforementioned solutions.

Like many other German universities, the Johannes Gutenberg University of Mainz has decided to use Seafile Pro Edition for this task not only due to low resource consumption but also because of the agile implementation of new features commissioned by the Data Center (ZDV) of our university like Shibboleth authentication. Additional benefits in terms of usability are the ability to invite external guests and the introduction of distinct roles for students, staff and faculty members into Seafile .

Seafile is not only accessible to users at the local university but to all users of universities throughout the whole of Rhineland-Palatinate that each have their own authentication infrastructure. In this talk, we are going to present the specialties of our setup of Seafile with special regards to performance aspects in our fully virtualized clustered setup with load balancing and to the multi-tenancy aspects of our federated setup. Our software stacks consists of the Seafile Pro Edition in a clustered setup, a MariaDB and Galera cluster, Memcached, Nginx, Apache and Shibboleth. We are currently experimenting with Ceph as the backend storage solution.

Storage Operations at CERN: CERNBox

Luca Mascetti¹

¹ CERN

Contact Details: luca.mascetti@cern.ch

CERNBox is a cloud synchronisation service for end-users: it allows syncing and sharing files on all major mobile and desktop platforms (Linux, Windows, MacOSX, Android, iOS) aiming to provide offline availability to any data stored in the CERN EOS infrastructure.

The success of EOS/CERNBox has been demonstrated by the high demand in the community for such easily accessible cloud storage solution which recently crossed 8000 users and by its role as integration point for different CERN services.

The system has been integrated in major work-flows for scientific computing and with existing scientific data repositories at CERN. It provides an authenticated file access (KRB5, GSI) using a range of access protocols and tools: physics data analysis applications access CERNBox via xrootd protocol; Jupyter Notebooks interact with the storage via file-system interfaces provided by EOS FUSE mounts; Grid jobs can access using GridFTP protocol and Windows clients can profit from the SAMBA endpoint.

We report on our experience with this technology and applicable use-cases, also in a broader scientific and research context and its future evolution into a CERN Home directory service.

Keeper – Archive the way you work

Author(s): Friederike Kleinfercher¹

Co-author(s): Vladislav Makarenko¹

¹ Max Planck Digital Library

Contact Details: kleinfercher@mpdl.mpg.de

The Max Planck Digital Library started a project at the end of 2015 to build up a service for all Max Planck Researchers to archive their data long term compliant and according to the rules for good scientific practice (<https://www.mpg.de/232144/rulesScientificPract.pdf>) of the Max Planck Society (MPS).

The technical aspects of this service were clearly specified but, as we experienced in the past, could be a bigger problem was, how do we get the researcher to deposit their data to Keeper and why is everybody using Dropbox despite of all of the known drawbacks and that it is not allowed to be used in the MPS. Therefore we decided that the aspect of a seamless integration into the working environment would be one of the key aspects whether our service would succeed or not. After the evaluation of several software (gitlab, owncloud, pydio a.o.) we identified Seafile as the most appropriate solution for our use case.

In our talk we would like to give insight into with which infrastructure we turn Seafile into a long term compliant archive and how Seafile functions as an easy-to-use interface to this infrastructure. Currently we enhanced Seafile software with two new functionalities, first an automated certificate creation after the user provided a basic set of metadata and a project catalog site where one can overview all projects stored in Keeper. The Keeper Service is in productive use since November 2016 after a beta phase of 6 months, were researchers from about 15 different Max Planck Institutes tested our Keeper service. Currently we have two Max Planck Institutes using the Keeper Service whereupon a Max Planck wide rollout is planned for the year 2017.

The Max Planck Digital Library: The Max Planck Digital Library (MPDL) is a central scientific service unit within the Max Planck Society (MPG) dedicated to the strategic planning, development and operation of the digital infrastructures necessary for providing the institutes with scientific information and for supporting web-based scholarly communication. The MPDL represents the Max Planck Society's goal of creating a modern, electronic infrastructure for supplying institutes with information, storing data, publishing research results and establishing web-based, scientific collaboration, while taking into account the interests of the institutes and their libraries. It is thus an instrument in safeguarding the competitiveness of the Max Planck Society in the world of international science.

HU Box – Cloud Storage. Synchronised data storage and file sharing for university use cases at the Computer and Media Service of the Humboldt-Universität zu Berlin (HU)

Enno Gröper¹; Karsten Asshauer¹

¹*Humboldt-Universität zu Berlin*

Contact Details: enno.groeper@cms.hu-berlin.de, karsten.asshauer@cms.hu-berlin.de

The project “Online-Speicher” (cloud storage – file synchronisation and sharing service) initiated by the Computer and Media Service (CMS) in 2013, is intended to establish a data storage solution for university use cases at HU. The main scenario at HU is to provide a collaborative data exchange solution for groups as well as personal synchronised data storage across all major device platforms. The HU Box (<https://box.hu-berlin.de>), based on the software Seafile (<http://www.seafile.com>), provides these functions similar to Dropbox for university use cases. Seafile can, like all sync and share services, be used in very different scenarios. It can, in principle, replace most central storage or file services and therefore become the main storage infrastructure. On the other side, it can also be used as a simple collaboration tool, where working groups share documents or jointly write articles. Besides the main scenario, we already identified several new ones. These are e.g. in the fields of research data management, user management, additional identifiers to be assigned as well as interfaces to existing systems, data privacy levels or collaborative editing features. These will be explored individually in separate projects.

The technical setup of the HU-Box – The CMS at HU currently provides the HU-Cloud incorporating the open source software OpenStack as an Infrastructure as a Service (IaaS) environment and CEPH as a distributed object storage.. The HU-Box is the first service that uses this infrastructure to offer a Seafile Sync and Share service for our users. Right now the Seafile setup consists of a HA load balancer pair, two worker nodes and a background node for document preview and full text search. For database requirements the central database server of the CMS is used. The Seafile nodes are using CEPH backend storage directly through librados, independent from OpenStack. Seafile service is scaled horizontally on demand. To achieve this, an ansible deployment stack is used, that complies with the official Seafile cluster deployment recommendation to use one worker node as a golden image. Shibboleth (Single Sign On) is used for authentication for HU-accounts and HU-externals who are member of the DFN-All federation (in progress).

Further, the HU corporate design was applied successfully for the Seafile Web-GUI with the integrated template framework (Python Django web framework). The CMS is also involved in the German translation of the Seafile clients and the web-GUI, via the Transifex platform.

The project blog of the HU-Box provides general information about Seafile and a central FAQ for helping users at HU (<https://blogs.hu-berlin.de/hu-box/>). The actual service is complemented by a training program for the HU staff.

In 2015 the community for the educational users of Seafile has been founded, which enables communication among educational users and a direct contact to the developers of the Seafile software for specific requirements (<https://hu.berlin/seafile-edu-community>).

B2DROP: The EUDAT Personal Cloud Storage

Author(s): Benedikt von St. Vieth¹

Co-author(s): Thijs Cobben²

¹ Forschungszentrum Jülich GmbH

² SARA

Contact Details: b.von.st.vieth@fz-juelich.de

B2DROP is a service offering from the EUDAT (EUropean DATa Infrastructure, eudat.eu) project. EUDAT is a collaborative pan-European infrastructure providing research data services, training, and consultancy for researchers, research communities, (national) research infrastructures, and data centres. It currently provides these services:

- B2DROP - Sync and Exchange Research Data,
- B2ACCESS - Manage Identity and Authorisation,
- B2SHARE - Store and Share Research Data,
- B2SAFE - Replicate Research Data Safely,
- B2STAGE - Move Data to Computation,
- B2FIND - Find Research Data

The service B2DROP is technically based on ownCloud - a self-hosted file sync and share software. It provides access to data/files via a web interface, sync clients, and WebDAV. With these functionalities it is a user friendly entrypoint to other EUDAT services.

Within the B2DROP development team there is ongoing effort to customize the ownCloud WebUI, so that it fits into the harmonized, branded visual identity of all user-facing EUDAT services. Another focus is on the integration of B2DROP with the EUDAT services suite, for example with B2SHARE and B2ACCESS.

One use-case for the integration with B2SHARE are researchers that work on a publication, synchronizing it across devices and sharing it with a limited number of users using B2DROP. After this publication is finalized, a user can simply click on a button in the WebUI and the final document is then transferred directly from B2DROP to B2SHARE for publishing purposes. During our presentation we will show how this integration works, which ownCloud framework parts we use and what we plan for the future.

For the integration with B2ACCESS, EUDAT's Authentication and Authorization Infrastructure, we have spent effort into extending the ownCloud authentication mechanisms with SAML (Security Assertion Markup Language) features. We will present our experiences with the Nextcloud user_saml plugin that wants to provide the aforementioned capabilities and that let us stop our effort, and we will present how this could change the B2DROP service in the future.

Current operational aspects will also be shared, for example deployment model and current usage.

Services & Site Reports

Polybox: Mobile working Use Case enabled by Sync & Share at ETH Zürich

Author(s): Tilo Steiger ¹

Co-author(s): Gianluca Caratsch ¹ Urs Gubler ¹

¹ *ETH Zürich*

Contact Details: steiger@id.ethz.ch, gianluca.caratsch@id.ethz.ch, urs.gubler@id.ethz.ch

Since summer 2013 the service „polybox“ based on owncloud got operative as a storage service. The polybox service is appreciated by many ETH users – students, PHD students, Professors and all staff members. With 50GB quota per user we have about 21.000 users using nearly 63TB of storage today.

Users of scientific departments asked for enabling a mobile working use case. Today state of the art mobile devices are providing IT resources (CPU, local storage) to users. Therefore, users want to be able to “preprocess” data (e.g. data being measured offline) and synchronize & share these scientific data sets beside their personal data.

It is proposed to switch from users “home directories” based on NAS services to only polybox like storage making some processes easier.

Additionally, the scalable infrastructure where polybox is configured on will be shown.

Keywords: infrastructure; service; storage; users; requirements

Services & Site Reports

Experience with running Owncloud on virtualized infrastructure (Openstack/Ceph) (continuation of talk at C3 2016)

Christian Schnidrig¹

¹ *SWITCH*

Contact Details: christian.schnidrig@switch.ch

Owncloud is quite popular among sync & share service providers. As the name implies, Owncloud was build with home users in mind. It can run on devices as small as a raspberry pi. At the same time this product is also sold to service providers who support with one single Owncloud instance more then 20k users. This already being an astounding achievement, it is not yet good enough. Service providers would need Owncloud to scale up to 100k users or even more.

At the CS3 in Zürich 2016, we presented our service SWITCHdrive. A sync & share service based on Owncloud which we run ontop of our IAAS offering (SWITCHengines) based on Openstack/Ceph. We discussed its advantages but also its limitations.

This year we will present how the service evolved. It grew quite a bit and we replaced the database (from PostgreSQL to Galera cluster). We'll talk about the motivation for that change and what our experience was. There is still one major pain point left: the NFS servers. We are currently addressing that problem and we will explore possible solutions in our talk.

A Staging Storage Sharing System for Data Handling in a Multi-site Scientific Organization

Author(s): Francesco iannone¹

Co-author(s): Irene Bellagamba ¹

¹ ENEA

Contact Details: francesco.iannone@enea.it

Abstract— In large scientific organizations, the laboratory experiments produce a huge amount of data and their processing and storage management are a challenging issue. Cloud architectures are exploited here and there for storage solutions and data sharing as well, in order to realize a collaborative worldwide distributed platform. Whilst large experimental facilities manage themselves Information and Communications Technology (ICT) resources such as: compute, networking and storage, small experimental laboratories are demanding more and more departmental ICT resources for their own scientific instruments aided by data acquisition and control systems, specially in terms of storage and sharing/publishing data solutions. ENEA Staging Storage Sharing (E3S) system has been developed over the ENEA ICT infrastructure using Owncloud as architectural component for file syncing and sharing. E3S provides a homogeneous platform able to store and share heterogeneous data produced by many different laboratories geographically spread on several sites and working on collaborative projects. The cloud storage technology has allowed to design an architecture based on concepts such as: i) data integrity and security, ii) scalability, iii) reliability. A first deployment of E3S works in a project for cultural heritage diagnostics involving several laboratories in different ENEA sites producing schema-free data. The paper presents the first deployment of E3S and a performance analysis of the architectural components. The performance analysis has been carried out with customized benchmark tools on a test bed consisting of a HPC cluster over Infiniband mounting a high performance storage. **Index Terms —** Cloud Storage, Linux, AFS, GPFS

Industrial session

Industrial

The good, the bad and the ugly: Experiences and learnings from years at the forefront of EFSS

Christian Schmitz¹

¹ *ownCloud Inc*

Contact Details: cschmitz@owncloud.com

ownCloud has been at the forefront of on premise EFSS deployments across the globe for the past couple of years.

I would like to take the opportunity to roll back time and draw the overarching story from the naive beginnings to the hard lessons to the concrete future concepts.

Industrial

Next steps in file sync and share security

Lukas Reschke¹

¹ *Nextcloud*

Contact Details: lukas@nextcloud.com

One of the main benefits of an on premise file sync and share solutions is the enhanced security it provides. This talks gives an overview over the latest security threats and hardenings. Examples of threats are current cross site scripting, DDoS and Spamming attacks and the consequences of insecure browser extensions. The talks showcases how the Nextcloud project is challenging this issues but also points out how other products and projects can benefit from improved security features and hardenings. Examples include new technologies like same site cookies and content security policy v3.

Presenter: Lukas Reschke Lukas has been contributing to the ownCloud/Nextcloud code since 2012, and is responsible for many of the security hardenings and features in the code. He has worked as security assessment and forensic investigator, reviewing security, giving trainings and dealing with breaches at Fortune 500 companies and several of the largest Swiss financial institutes.

Industrial

Improving Pydio performances with PydioBooster

Charles du Jeu

Contact Details: charles@pydio.com

Pydio is a well-known open source software for synchronisation and sharing. It is written in PHP which brings many advantages, including ease of deployment and ease of hacking. Even though the last version of the PHP engine improved performances greatly (php7), this language still has some limitations that are inherent to its design. To overcome these limitations, the Pydio team decided to develop a «companion» tool that runs beside the standard LAMP stack. **Pydio Booster** is written in Go, the server language pushed by Google. It is a compiled dependency-free binary that can be very easily started on any server. The tool is useful to many admin's, from non-tech-savvy home users (bringing one-click auto-configurable features), to professional sys-admins working on large scale architectures (bringing modularity via micro-services).

In this talk, we will describe the advantages of we get "best-of-both-worlds" out of PHP and Go, and how it is allowing to boost Pydio deployments. We will also have a quick tour of the latest new features of Pydio.

Industrial

Seafile project: status and outlook

Author(s): Daniel Pan¹

Co-author(s): Jonathan XU¹

¹ *Seafile Ltd.*

Contact Details: daniel.pan@seafile.com, jonathan.xu@seafile.com

In this presentation we'll present a few topics about Seafile project.

- Seafile's current features, especially features related to educational and research institutes.
- Seafile's architecture
- Future development plan

Industrial

Next generation federated sharing

Björn Schiessle¹ ; Frank Karlitschek¹

¹ *Nextcloud*

Contact Details: bjoern@nextcloud.com, frank@nextcloud.com

Federated sharing is the foundation of the open cloud mesh initiative. It was developed by Björn Schiessle and Frank Karlitschek starting 2013. This talk will cover the latest improvements in federated sharing and also the status of the API standardization effort in the OCM initiative. The goal is to have a common standard that works across vendors but also gives different products the opportunity to innovate. Other main topics of this talk will be the latest security and performance improvements as well as the protocol enhancements to support global user discovery and auto completion. This will be done by a new open source server component and a federated sharing protocol enhancement. Presenter: Frank Karlitschek, Björn Schiessle

Frank Karlitschek Frank founded the ownCloud project in 2010 to put home users and enterprises back in control of their data. To improve the company-community balance and accelerate the project he founded Nextcloud in 2016 and has been tirelessly working to realize his vision ever since.

Björn Schiessle Björn has been developing federated technology for ownCloud and Nextcloud since 2013. Computer Scientist, graduated at University of Stuttgart and Open Source and Open Standards Evangelist for privacy respecting, distributed and federated networks, FSFE's Deputy Coordinator Germany has a deep understanding of the technical and social aspects of technology.

deutsche.hochschul.cloud – An initiative to promote the networking of German universities and research institutions

Christian Sprajc¹

¹ PowerFolder

Contact Details: sprajc@powerfolder.com

The starting point is the ever-increasing networking of the research and education landscape, especially at the application level, in the form of so-called File Sync & Share services. In order to be able to respond to today's modern research landscape, it is important to provide the appropriate tools for users in education and research to support organisation-wide and cross-organisational cooperation. Over the last five years, enormous progress has been made. Several services have been successfully put into operation - in particular in Baden-Württemberg (bwSync and Share), Baden-Württemberg (bwSync and Share), Baden-Württemberg (bwSync and Share), Baden-Württemberg (bwSync and Share) and Baden-Württemberg (bwSync and Share), with a reach of million users. In addition, innumerable colleges and research institutes operate their own services. Another important step has been taken by the DFN e.V. with the DFN-Cloud initiative, which allows educational and other facilities to access other members' cloud services without operating their own File Sync & Share service. These are, however, predominantly regional or island solutions.

File Sync & Share is an established tool for the exchange of digital learning and teaching content, scientific studies, as well as administrative processes. However, a cross-departmental relationship between users of commercial monolithic cloud applications, such as Dropbox, is still missing.

To this end, we present the initiative "deutsche.hochschul.cloud" (DHC).

The aim of the "deutsche.hochschul.cloud" (DHC) initiative is to promote the regional, national and international networking of universities. The digitisation of educational infrastructures in working groups has already been discussed at the National IT Summit 2012, in Essen. Nowadays, the Internet has become more and more a part of everyday life for students, teachers and university administrators. The globalisation of education is taking place.

The purpose of the DHC is to coordinate and promote the networking of existing clouds and software development on the basis of trustworthy services. Important key points are the establishment of trust centres at the institute, state and federal level, the security of the cloud services through encryption and the use of existing security infrastructures (PKI + AAI); In addition, the rapid networking of the other universities should not be lost sight of. In a nationwide context, the DFN e.V. could play an important key role as a central trust centre.

The initiative presented here also has the following aims:

- The technical implementation of the necessary software functions, which is usually referred to as the "Federated Cloud".
- To win the support of the existing and new service operators.
- To create consensus and support at the political level.
- To work in conjunction with similar initiatives such as "Open Cloud Mesh".

Several intuitions, such as Karlsruher Institute of Technology (bwSync&Share), GWDG, Leibniz Rechenzentrum, RRZE (FAUBox) and several more institutions, already joined the DHC initiative and provide support in form of use case descriptions, functional testing and financial contributions. Altogether the starting members provide a reach of about one million users in academia in Germany (of 2,7 million total). This would enable the initiative "deutsche.hochschul.cloud" to make an important social contribution to digitalization in Germany.



Figure 1: deutsche.hochschul.cloud Logo



Figure 2: deutsche.hochschul.cloud Institutions

Projects and collaborations session

Projects and collaborations

OCM

Peter Szegedi¹

¹ *GEANT*

Contact Details: szegedi@terena.org

Open Cloud Mesh (OCM) is a joint international initiative under the umbrella of the GÉANT Association that is built on the open Federated Cloud Sharing application programming interface (API) - first initiated and implemented by ownCloud Inc. - taking Universal File Access beyond the borders of individual clouds and into a globally interconnected mesh of research clouds without sacrificing any of the advantages in privacy, control and security an on-premises cloud provides. OCM defines a vendor-neutral, common file access layer across an organization and/or across globally interconnected organizations, regardless of the user data locations and choice of clouds.

Projects and collaborations

Federated sharing with the Open Cloud Mesh API

Author(s): Dimitri van Hees¹

Co-author(s): Joost Farla¹

¹ *Apiwise*

Contact Details: dimitri@apiwise.nl, joost@apiwise.nl

In this topic we'll cover our approach to enable federated sharing between different cloud services using the vendor independent Open Cloud Mesh API specification. The spec is an Open API (fka Swagger) Specification from the Open API Initiative and gives new insights on how federated sharing could work using the web's standards of 2017.

New site services session

New site services

OwnCloud scalability and concept design

D Pennings¹

¹ 360/CT

Contact Details: d.pennings@360ict.nl

This is a session about OwnCloud scalability and its architecture in general. We needed to design a Nextcloud environment for 10.000-20.000 users and decided to do a concept design. This resulted in a comparison of Owncloud implementations presented at CS3, which we presented at the Nextcloud conference; <https://conf.nextcloud.com/conference/NextcloudConference2016/program/proposal/12>

I also had a lot of questions, which Frank Karlitschek (the founder of Owncloud) answered in the session himself.

I want to do the same presentation, and hopefully have some more data to compare the customers who actually are running Owncloud in production. For example, I had only the data that was available since the last CS3 meeting and with some help with the CS3 participants I will try to gather some more current data and some more on their actual load/usage.

New site services

Scalable management with Seafile SPORE specification

Sébastien Finkbeiner¹; Vincent Lucas¹

¹ *Strasbourg University*

Contact Details: s.finkbeiner@unistra.fr, lucas@unistra.fr

The Strasbourg University provides a file synchronization and sharing service via Seafile software. This service is delivered to more than 11000 researchers, teachers and employees which are distributed across more than 900 structures, such as schools, laboratories and departments. At this scale, a centralized management of the accounts, guest accounts, groups, shared repositories and space quotas is not an option.

The idea of this paper is to propose a simple and generic mechanism to distribute and delegate this management tasks to each structure or user requesting a custom configuration.

Providing such a scalable and distributed management solution is an interesting problem, because it must:

- at one side, request the University structured data repository in order to compute the rights relative to a given user;
- at the other side, be able to communicate with Seafile to inject the computed custom parameters.

To this end, we decided to create the Seafile SPORE description file (cf. [seaa]) based on the well documented Seafile REST API (cf. [seab]).

A SPORE description (cf. [spo]) is a simple JSON document, which describes a service HTTP API, in order to dynamically or statically generate high level client objects. Lots of SPORE implementations are available, such as: Clojure, Javascript, Lua, Nodejs, Perl, Python, Ruby, Livescript and Groovy. Moreover SPORE description is the main format used to define, manage and interconnect Strasbourg University software.

Therefore, the first section of this paper presents the Seafile SPORE specification publicly available (cf. [seaa]).

Then, the second section details the guest accounts management, which includes:

- The interconnection process between the University LDAP and the Seafile service: it is implemented in Python on top of the Britney Python SPORE client (cf. [bri]).
- The dedicated web pages: this part is developed as a “uWSGI” service with “bottle” a Python web framework (cf. [bot]) and integrated to the HTTP “nginx” server which is used for the Seafile frontend.
- The integration inside Seafile web interface: the link to create guest accounts is added to Seafile web interface and the authentication is done directly by Seafile Django service.

This functionality is heavily used for building external research collaboration and to create student accounts.

Furthermore and based on the same mechanisms, the third section depicts on going feature implementations such as group, shared repositories and quotas management.

Finally, we conclude with the benefits and limitations of this approach and reviews which load of management tasks can be delegated.

References:

[bot] Bottle: Python web framework. <http://bottlepy.org>

[bri] Python implementation of spore (based on spyre). <https://github.com/agrausem/britney>

[seaa] Seafile spore description file. <http://rest-api.u-strasbg.fr/seafile/description.json>

[seab] Synchronization algorithm | seafile server manual.

https://manual.seafile.com/develop/web_api_v2.1.html

[spo] Spore - specification to a portable rest environment. <https://spore.github.io/>

ALLVIS, Uppsala Universitys official storage of scientific Data

Davor Vusir¹

¹ *Uppsala University*

Contact Details: davor.vusir@uadm.uu.se

Excerpt from the university directors decision: "Propose how services for cost effective and secure, long term storage of scientific data are made available to scientists with the possibility to integrate, in a flexible manner, with local, national and international systems for storage of large data quantities". With her decision in mind, project ALLVIS, Uppsala Universitys official storage of scientific Data, was started. Allvis is a wise dwarf in the nordic mythology but in the modern world it is an acronym: all visdom (eng. all wisdom). Here we gather our scientists findings for the good (and gain) of future generations.

As all of you, we have come to the conclusion that a CS3 solution is a must. I would like to present our design thoughts.

We will use ownCloud as middleware because of the flora of clients and its modularity; running on Linux adds possibilities to tie different forms of storage, DAS with suitable file systems or network based file systems like NFS or SMB. And the possibility to use various catalogues for authentication. In our design the usage of External Storage is central. The External Storage feature is a good tool that gives us the opportunity to both segment storage and add other sources out of need or will to gather various resources under one umbrella. SMB is the main protocol. This widespread protocol with its richness gives us an instrument to achieve and apply granular access control and yet give the scientific community easy access.

Other important components of the design is the usage of Active Directory as the user catalogue, authentication source and the base for access groups for both ownCloud and file shares on SMB servers, Microsoft DFS (Distributed File System) to gather various SMB servers under a Global Name Space. This will, of course, enable an easy to remember point of origin for all file server resources. The same design principle applies to NFS; a gateway to gather various NFS exports.

But this design comes with an inherited cost and areas of concern and possible solutions will be briefed.