Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# Benchmarking and monitoring framework for interconnected file synchronization and sharing services[☆]

Piotr Mrówczyński [a,b,*], Jakub T. Mościcki [c], Massimo Lamanna [c], Frederik Orellana [b]

[a] Lodz University of Technology, Poland
[b] Technical University of Denmark, DeIC, Denmark
[c] European Organization for Nuclear Research CERN, Switzerland

## HIGHLIGHTS

- The usage of benchmarking, testing and monitoring framework for CSS is documented.
- Framework allows synchronization details to be collected, processed and displayed.
- The scope ranges from general overview, daily analysis to single sync transfers.
- The ownCloud protocol performance at different user loads and networks is analyzed.
- The streaming and metadata performance comparison of services is introduced.

ARTICLE INFO

*Article history:*
Received 2 May 2016
Received in revised form
23 November 2016
Accepted 4 March 2017
Available online xxxx

*Keywords:*
Cloud storage
Performance
Testing
Benchmarking
Monitoring
ownCloud
Seafile
Dropbox
Smashbox

ABSTRACT

On-premise file synchronization and sharing services are increasingly used in research collaborations and academia. The main motivation for the on-premise deployment is connected with the requirements on the physical location of the data, data protection policies and integration with existing computing and storage infrastructure in the research labs. In this work we present a benchmarking and monitoring framework for file synchronization and sharing services. It allows service providers to monitor the operational status of their services, understand the service behavior under different load types and with different network locations of the synchronization clients. The framework is designed as a monitoring and benchmarking tool to provide performance and robustness metrics for interconnected file synchronization and sharing services such as Open Cloud Mesh.



## 1. Introduction

There is a growing number of large-scale file synchronization and sharing services in the research collaboration networks—such as GÉANT [1] and NeIC [2]. Examples include SWITCHDrive, CERNBox, GARRBox, SURFDrive, CloudStor and many others [3]. Companies and organizations look for open source cloud storage solutions in order to satisfy their local customer needs and create dedicated services for research data.

Currently, two of the most popular cloud synchronization solutions in research collaboration networks are ownCloud [4] and Seafile [5]. Advantage of these solutions is that they provide an easy Dropbox-like user interface and also allow a high degree of customization of back-ends and front-ends. Furthermore, they also offer fast and easily accessible support in the form of a large community of users and contributors.

Open-source solutions offer flexible deployment options and allow easy introduction of modifications in software. However, this may lead to large differences and divergence between installations. In a long run, quality control of the overall service might be complex and involve assessing service availability and reliability, data integrity, network and hardware configuration and so on.

☆ The present work is based on the Bachelor's Thesis (Mrówczyński, 2016, [10]) of Piotr Mrówczyński, Lodz University of Technology, written under the supervision of Andrzej Romanowski during an internship at the Danish e-Infrastructure Cooperation, September 2015–February 2016.

* Corresponding author at:Technical University of Denmark, DeIC, Denmark.
*E-mail address:* piotr.mrowczynski@yahoo.com (P. Mrówczyński).

http://dx.doi.org/10.1016/j.future.2017.03.006

This problem becomes particularly important for interconnected cloud services, such as Open Cloud Mesh (OCM) [6], which aim at creating a cloud storage service ecosystem to share and synchronize files between different sites. OCM is a lightweight federation of on-site storage services, which may be deployed using different software stacks and technologies (e.g. PyDio [7]). This creates a new type of operational challenges because of interconnection between services. For example, shared files could be corrupted, unavailable, accidentally deleted, lost during synchronization, bounce in a conflict loop or made unavailable by a local server misconfiguration or overload. In such cases, prolonged file retrieval and synchronization may be hard to troubleshoot remotely and affect stability of other clouds in the mesh. Consistent and comparable quality and performance metrics across services and a continuous testing infrastructure could improve stability and robustness of Open Cloud Mesh.

In this paper we document the usage of an existing tool, Smashbox [8], to test, monitor and benchmark cloud storage services. The framework was initially developed at CERN [9] as an acceptance tool for ownCloud-based installations. The framework has been extended with benchmarking and monitoring functionality. Plugins for the other systems have also been implemented (Seafile, Dropbox) [10].

In the next sections, two production grade cloud storage services are compared: the research data service of **DeIC** (`data.deic.dk`) and **CERNBox** (`cernbox.cern.ch`). We compare these services from a perspective of a local network client as well as the clients on external networks with significant latency to the cloud servers. We also investigate the influence of different load types on the synchronization process. We demonstrate monitoring capabilities of the extended smashbox framework, which are important for day-to-day service management. Finally, we demonstrate a simple comparison between on-premise services based on different software stacks (Seafile, ownCloud, CERNBox/EOS [11], ClawIO) with public cloud services (`dropbox.com` and `seafile.cc`) hosted at AWS [12].

Our work is complementary to existing related research on synchronization services. Analysis of open synchronization protocols and service architecture is presented in *Internet Storage Sync Problem Statement* [13]. Quantitative service assessment with a focus on commercial and public cloud services from a perspective of a user located in Europe is presented in *Benchmarking Personal Cloud Storage* [14]. This paper also discusses the impact of different design choices and distinct synchronization features, such as bundling, de-duplication, compression and delta-sync. Benchmarking of cloud services with mobile client devices is in focus of *QuickSync: Improving Synchronization Efficiency for Mobile Cloud Storage Services* [15]. However this work is specific for mobile cloud storage environments.

## 2. Benchmarking, monitoring and testing framework [16]

Smashbox was initially developed for functional testing of ownCloud-based services. Smashbox is an end-to-end testing framework for the core storage functionality of a file synchronization and sharing service. It may be run interactively from a command line, via cron jobs or via scripting. Standard set of smashbox tests includes various scenarios—basic synchronization (upload/download/conflicts), concurrent client operations, race-conditions, sharing consistency, file integrity and protocol correctness.

We extended the original framework with monitoring and reporting capabilities to report timing information for different stages of ownCloud synchronization protocol, total execution time of the tests and the number of errors and fault conditions. We developed a monitoring component to capture transfer rates recorded on the local network interface during the synchronization. We also added a possibility to plug-in new types of synchronization clients into the framework (Dropbox and Seafile). These additions are integrated in the upstream smashbox framework and available as optional features in the configuration.

The monitoring has been integrated with Grafana web service and InfluxDB [17]. Visualization is also possible using python and matplotlib scripts [18].

Full testing and monitoring stack (Fig. 1) is available as a Docker container [19] to simplify local deployments.

## 3. Test setup

In this paper we describe results obtained with an extended version of simple test scenario, `test_nplusone` [20], in which two test clients are used—Fig. 2. During the synchronization two phases may be distinguished—data upload (client to server sync) and data download (server to client sync). First client uploads set of files to the server, whereas the second client downloads this set from the server. We collected the network traffic data during the test execution, recorded the number of synced files, test case specific errors, synchronization and overall duration of the test.

Tests were performed between 16 Dec 2015 and 28 Jan 2016 and were repeated more than 1000 times for each test configuration specified below.

For the purposes of this paper, 5 different data loads for the synchronization were used—1 file of 1 byte, 1 file of 100 MB 10 files of 10 MB, 1000 files of 10 kB and 100 files of variable size. We refer to these test loads using their abbreviated names: 1/1 B, 1/100 MB, 10/10 MB, 1000/10 kB and 100/var (Table 1).

The scenario with variable file size distribution (100/var) corresponds to the distribution observed [21] in CERNBox production system. This representative data load allows to study the synchronization performance of a mixture of small and large file sizes.

We used two test client hosts (Table 2), one at DeIC in Denmark and another at the CERN Wigner site [22] in Hungary. The tests involved servers at DeIC, Lyngby and at CERN, Geneva. We repeated the tests in different client–server configuration as shown in Fig. 2. Our tests included fully local network setups (both client and server at DeIC) as well as across larger latencies (∼25 ms and ∼50 ms). Fig. 2 shows the latency between the DeIC/CERNBox cloud services and both test clients at DeIC and at CERN Wigner.

## 4. Synchronization performance in the local network

The tests shown in this section were performed using the DeIC test client host (Ref. Table 2) located on the local area network of Technical University of Denmark, with 1 ms latency to the DeIC Data service. Five types of user loads were tested as previously shown in Table 1. For each test, the duration of upload and download of all files in the test set was recorded and presented as CDF (Cumulative Distribution Function) on Fig. 3. We also measured the network transfer rates observed during each test and the aggregated transfer rates are shown as CDF on Fig. 3.

Each point in the Transfer Rates CDF on Fig. 3 (left) is representing transfers per second in both upload and download observed in 1000 test synchronizations (equally distributed in time during the observation period, Ref. Section 3) and cumulatively represent the profile of how probable is it for specific transfer rates ranges to occur. Exemplary, in case of 100/var—upload (Ref. Table 1) 90% of transfer rates were observed in range 100 kBps and 20 MBps, moreover, 60% of observed transfer rates were in range 1 MBps and 20 MBps. On the other hand, in case 10/10 MB—download (Ref. Table 1) 95% of transfers per second were observed in range 20 MBps and 40 MBps.

Please cite this article in press as: P. Mrówczyński, et al., Benchmarking and monitoring framework for interconnected file synchronization and sharing services, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.006