Towards Autonomous Data Sharing Across Personal Clouds

Roland Tornyai[1] and Attila Kertesz[1,2]

[1] University of Szeged, Department of Software Engineering
H-6720 Szeged, Dugonics ter 13, Hungary
tornyai.roland@gmail.com
[2] MTA SZTAKI Computer and Automation Research Institute
H-1518 Budapest, P.O. Box 63, Hungary
kertesz.attila@sztaki.mta.hu

Abstract. Cloud Computing has reached a maturity state and high level of popularity that various Cloud services have become a part of our lives. Mobile devices also benefit from Cloud services: the huge data users produce with these devices are continuously posted to online services, which may require the use of several Cloud providers at the same time to efficiently store these data. Using Cloud-based storage services such as Personal Clouds for these purposes are free for certain amount of data; therefore uniting these separate storages can provide a suitable solution for these user needs. In this paper we propose a novel solution for autonomous data management among Personal Clouds. Our approach applies a continuous monitoring component to track the performance of the managed Cloud providers, and based on this measured historical information it manages user data across the interconnected providers in an autonomous way.

1 Introduction

Nowadays Cloud Computing has reached a maturity state and high level of popularity that various Cloud services have become a part of our lives. These services are offered at different Cloud deployment models ranging from the lowest infrastructure level to the highest software or application level. Within Infrastructure as a Service (IaaS) solutions we can differentiate public, private, hybrid and community Clouds according to recent reports of standardization bodies [8]. The previous two types may utilize more than one Cloud system, which is also called as a Cloud federation [9]. One of the open issues of such federations is the interoperable management of data among the participating systems. Another popular family of Cloud services is called Cloud storage services or Personal Clouds. With the help of such solutions, user data can be stored in a remote location, in the Cloud, and can be accessed from anywhere. Mobile devices can also benefit from these Cloud services: the enormous data users produce with these devices are continuously posted to online services, which may require the use of several Cloud providers at the same time to efficiently store and retrieve

L. Lopes et al. (Eds.): Euro-Par 2014 Workshops, Part II, LNCS 8806, pp. 50–61, 2014.

these data. The aim of our research is to develop a solution that unites and manages separate Personal Clouds in an autonomous way to provide a suitable solution for these user needs.

In this paper we address the open issue of data interoperability in Clouds, and propose a novel solution for interoperable personal data management in storage Clouds. Our approach applies a continuous monitoring component to track the performance of the managed Cloud providers, and based on this measured historical information it manages user data across the interconnected providers in an autonomous way. Therefore the main contributions of this paper are: (i) envisioning a solution for autonomous data management among Personal Clouds, (ii) the development of an application that is able to measure the performance of the interconnected providers and use this information to distribute user data among them, and (iii) the evaluation of our proposed approach with four providers.

The remainder of this paper is as follows: Section 3 presents an overview of the addressed Cloud storage providers and introduces our motivation for this work; Section 4 describes our approach for autonomous data management and presents our proposed application. Finally, Section 5 discusses the performed evaluations, and the contributions are summarized in Section 6.

2 Related Works

Regarding related works, the need for data interoperability and the extensive use of Cloud storage services have been identified by various research and expert groups (eg. [8,5,1]). Managing user data in the Cloud also raises privacy issues [10,6] that need to be taken into account during data processing. Nevertheless in this paper we refrain from legal issues and focus on interoperability problems. Dillon et. al [2] gathered several interoperability issues that need to be considered in Cloud research, and named a new category called Data Storage as a Service to draw attention to the problem of data management in Clouds.

Drago et al. [3] have already analysed the usage of Dropbox on the Internet, and showed that it is the most popular provider of Cloud-based storage services. They presented an extensive characterization of Dropbox in terms of system workload and typical usage scenarios. They concluded that the performance of Dropbox is highly impacted by the distance between the clients and datacenters. They also identified a variety of user behaviours, e.g. taking full advantage of its functionalities by actively storing and retrieving files. In a later work [4] they continued this investigation for comparing 5 providers. Their results showed that all considered provider services suffer from some limitations, and in some scenarios the upload of the same set of files can take much more time, so they also acknowledged performance differences among these providers.

Garcia-Tinedo et al. [7] have also addressed performance issues of Personal Clouds. They developed a tool for actively measuring three providers: Dropbox, Box.com and SugarSync. They performed measurements for two months with various data transfer load models to search for interdependency among data sizes, transfer quality and speed. They published their measurement data and

concluded that these providers have different service levels, and they often limit the speed of downloading. This work also served as a motivation for our research, but we decided to develop a more lightweight and easily extendible measuring tool to support our further research goal of autonomous data sharing among these providers.

3 An Approach for Autonomous Data Management among Personal Clouds

Besides IaaS Cloud solutions the largest amount of user provided data are stored at Cloud storage services also called as Personal Clouds [8,5]. Their popularity is accounted for easy access and sharing through various interfaces and devices, synchronization, version control and backup functionalities. The freemium nature [11] of these services maintain a growing user community, and their high number of users also implies the development of other higher level services that make use of their cloud functionalities. To overcome the limits of freely granted storage, users may sign up to services of different providers, and distribute their data manually among them, which situation leads to a provider selection problem – see Figure 1. In this situation tracking the amount and location of the already uploaded files and splitting larger files can be a difficult task for everyday users, which leads to the problem of Cloud provider selection – not to mention their different capabilities concerning data transfer speeds. These facts serve as a motivation for our research, and the main goal of this work is to propose a higher level service that helps users to better manage their data by providing automated access to a unified storage over these Clouds.

In this paper we addressed four providers, namely Dropbox [15], Google Drive [14], SugarSync [17] and Box.com [18]. Their main properties are shown in Table 1. The foundation of Dropbox is originated in a problem we still face nowadays.

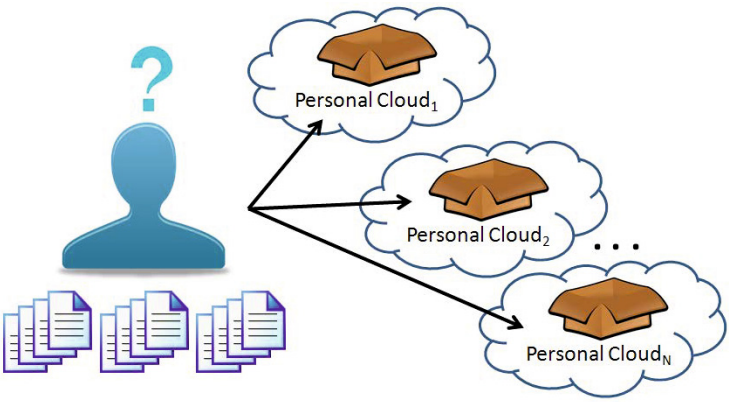

Fig. 1. Cloud provider selection problem

Drew Houston, one of the founders of the company, kept on leaving his pendrive at home during attending courses at MIT. Since he used several computers simultaneously, he had to email necessary files to stay updated at all devices, which he got tired of soon. Hence no suitable online data sharing solution existed by that time, he invented one. In 2007 he founded Dropbox Inc, and their service was kicked off in 2008. By 2011 it reached 14% market share by having 50 million registered users. According to the latest figures, this number exceeded 200 million in 2013 [16]. Its freemium model grants 2 GBs storage for a new registration that can be extended up to 8 GBs by inviting others or performing certain tasks. Concerning the main properties of the service, it is written in Python, supports version control, and applies the so called "delta encoding" technique, which only uploads the newly changed parts of a previously uploaded file. It supports a wide range of APIs and has several SDKs, as shown in Table 1.

Google Drive is a Personal Cloud solution of Google. It was initiated in 2012, but it has several predecessors such as Google Docs since 2006. It also serves as an in-house data store for several other Google services, therefore it provides 15 GBs freely for a new user. Thanks to the coupled services of Google, its web interface is capable of previewing numerous file formats in a browser. SugarSync was launched in 2009, but its predecessor Sharpcast Photos dates back to 2006. It provided 5 GBs free storage for a newly registered user till December 2013, when the owners announced to close freemium services till February 2014. Since then its free service is only valid for 30 days trial period. Box.com was founded as a startup company in 2005. Since 2010 it has a built-in file preview functionality. It provides 10 GBs of free storage for a new user.

Table 1. The main properties of the managed providers

Provider	**Initial Storage (GB)**	**Bonus (GB)**	**Max. Storage (GB)**	**Supported OS**	**Mobile Platforms**
Google Drive [14]	15	-	15	Win, Mac	iOS, Android
Dropbox [15]	2	0.5	8	Win, Mac, Linux	iOS, Android
SugarSync [17]	5	-	5	Win, Mac	iOS, Android
Box.com [18]	10	-	10	Win, Mac	iOS, Android

Provider	**Version Control**	**Encryption**	**Num. of devices**	**API**	**SDK**
Google Drive [14]	+	-	-	+	Java, Python, PHP, .NET, Ruby
Dropbox [15]	+	+	-	+	iOS, Android, Python, Ruby, Java, OS X
SugarSync [17]	+	+	1	+	Java
Box.com [18]	+	+	-	+	iOS, Android, Python, Ruby, Win, Java, C#

4 The Proposed Solution

Now that we have stated our motivation and introduced the considered Cloud providers in the previous section, we describe our proposed solution shown in Figure 2.

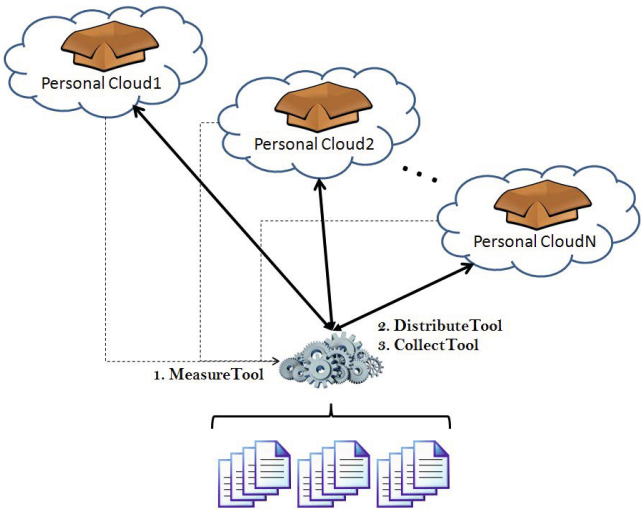

Fig. 2. The proposed solution

Our approach is demonstrated with an application written in Java, which uses the OAuth [12] standard to authenticate users. By using this protocol, client tools can act on behalf of certain users to access certain files without knowing their passwords, they use so called tokens instead with limited lifetime. Its version 2.0 is the latest since 2012. It is only a framework not a clearly defined protocol so it can be regarded as a guideline, therefore different providers have slightly different implementations. The application consists of three components:

- the MeasureTool component for performing monitoring processes,
- the DistributeTool component for splitting and distributing files,
- and the CollectTool component for retrieving splitted parts of a required file.

4.1 The MeasureTool Component

This component implements three basic functions: connecting to a user account at a certain provider, uploading and downloading certain files to and from the storage of this account. It has a plugin-based structure to separate methods for different providers and to enable further provider support.

A monitoring process for measuring the performance of a provider consists of generating a file of a predefined size with randomized content, uploading this file to the provider's storage under a given user account, then downloading this file back to the host of the application. The monitoring results and the measured performance data for the mentioned providers are shown and discussed later in Section 5.

4.2 The DistributeTool Component

The main task of this component is to apply certain policies for splitting up and packaging files to be distributed among the participating Cloud providers in an efficient way.

The file to be uploaded to the providers' storages is first split to a predefined number of files, what we call chunks, with equal sizes (large files are also supported, since only parts of a file are in memory at a time using buffering). The second step decides where to upload these file chunks. Once it has been determined and a chunk is uploaded, the DistributeTool component stores chunk identifiers (e.g. name, user token, file ID) to a local meta-data cache file. By using this meta-data file, the CollectTool component can later fetch the required chunk files from the different providers.

The provider selection in the second step is made upon the information gathered by the MeasureTool component. Historical performance values are also stored and taken into account, and it is the role of the application administrator to set the relevance (i.e. ratio) of historical and latest performance results for provider selection. The measured performance values are converted to the following format (denoting percentage shares – the sum of these values represent 100%) taking into account the aggregated historical performance values (h), the latest performance values (l) and their ratio (r) by evaluating ($h + l * r$), e.g.:

$$\{\textit{"googledrive" : 5392, "dropbox" : 1615, "box" : 1085, "sugarsync" : 292}\ \}$$

According to these configuration numbers, the DistributeTool component takes the sum of these values (sum) and generates a random number independently drawn from the range $\{0, sum\}$ for each chunk by using Gaussian distribution. The given number will determine the provider to be used for the actual chunk (e.g. the randomly generated number 4537 denotes Google Drive, while 7509 selects Box.com according to the example above ($5392+1615+502$)). This selection criteria can be easily expanded later if needed, e.g. incorporating the experienced number of failures during the measurements. Our further goal is to support scenarios, where not only freemium storages are considered. In this way provider selection could be optimized by payment minimization.

4.3 The CollectTool Component

As mentioned in the previous subsection, this component is able to collect the previously uploaded user files from the Cloud providers by using the meta-data

description file. Once the chunks of a required file are retrieved, they are unified with an optimized buffering technique.

5 Evaluation

We have performed our evaluations on a private IaaS Cloud based on OpenNebula. It has been developed by a national project called SZTAKI Cloud [13], which was initiated in 2012 to perform research in Clouds, and to create an institutional Cloud infrastructure for the Computer and Automation Research Institute of the Hungarian Academy of Sciences. Since 2013 it operates in experimental state, and since 2014 it is in production state available for all researchers associated with the institute. It runs OpenNebula 4.4 with KVM, and controls over 440 CPU cores, 1790 GBs of RAM, 66 TBs shared and 35 TBs local storage for serving an average of 250 Virtual Machines (VM) per day for the last month.

The application consisting of the previously discussed components has been deployed in a VM started at SZTAKI Cloud. The evaluation architecture is depicted in Figure 3.

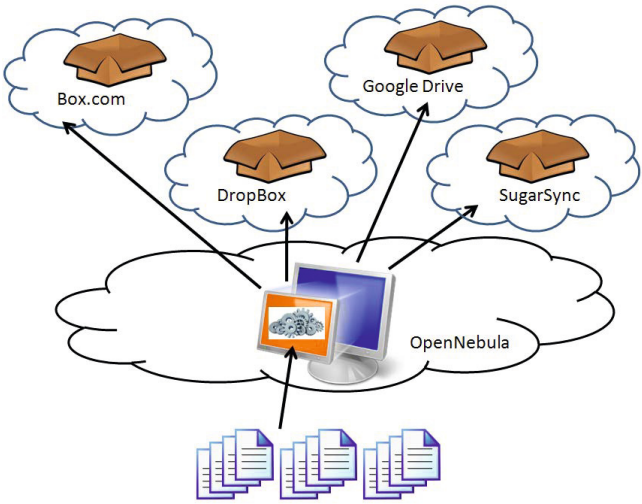

Fig. 3. Evaluation architecture

5.1 MeasureTool Evaluation

For users, the most important metric for measuring provider performance is the data transfer speed. Therefore we used this metric to monitor the providers, and to use as a base for autonomous file sharing. To perform an evaluation of the MeasureTool component, we up- and downloaded files to each Personal Cloud with the following data sizes: 5, 10, 50 and 100 MBs, considering the following

scenarios: (i) transferring two 5 MBs file or a 10 MBs file, (ii) transferring five 10 MBs file or a 50 MBs file, and (iii) transferring ten 10 MBs file or a 100 MBs file.

In this way we arrived to 6 different cases, and we could also measure data transfer performance for handling many small and few big files. We went through all cases systematically, and performed the same measurements several times (at least 5 for each case). Once the limit of the freemium storage of a provider got exceeded, we halted the measurement and deleted all files on that storage to start following tests. We performed the same measurements on different periods of a week, i.e. on weekdays and at weekends. For measuring failures, we omitted failed transactions caused by server-side errors. Finally, the measured time taken to upload and download the files incorporates the writing of the files to the storage discs at the providers' side (in case of Google Drive we could have omitted this interval, if we wanted to).

In the following diagrams we show the experienced performance values and provide a discussion on these results. Figure 4 shows detailed values concerning average, minimal and maximal transfer speeds. From these results we can see that Google Drive has the best performance values followed by Dropbox and Box.com, while SugarSync has the worst values, which is further acknowledged by detailed results shown in Figure 5.

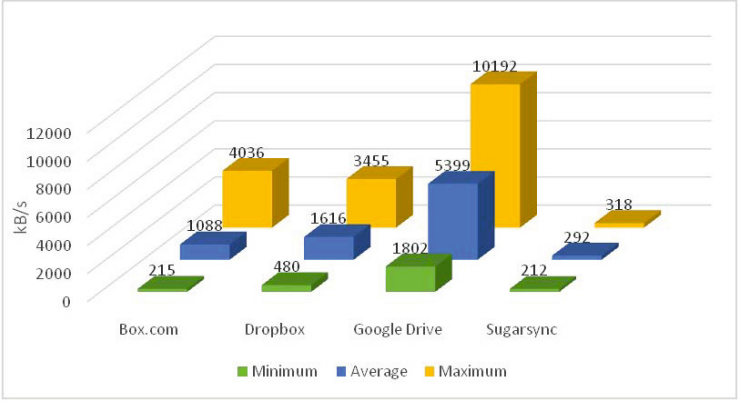

Fig. 4. Measured speed of the utilized Cloud providers

While the difference between Google Drive and SugarSync is obvious, it is not easy to compare Box.com and Dropbox. As this figure suggests, many small files are better handled by Dropbox, while bigger files are transferred faster by Box.com. It is also an interesting observation that transfer speeds are accelerating for larger files. This is caused by the fact that during transferring a small file the connection won't "speed-up" in time, but for bigger files it can utilize most of the available bandwidth. As mentioned before, the evaluation has been

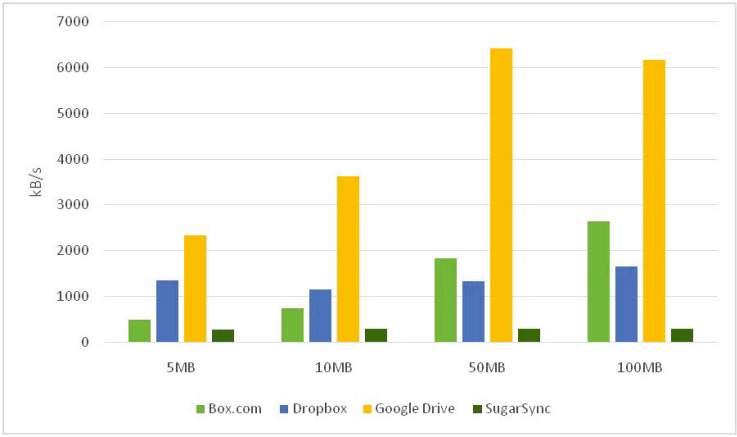

Fig. 5. Speed of providers for different amount of data

performed at different days of a week, but we experienced no major differences in these cases.

Table 2 depicts the amount of data transferred to and from the utilized providers. Of course, the same cases have been executed for all providers, the differences among them lies in transaction restarts caused by failures or storage limit exceeding (though "delta encoding" and similar techniques could save some amount of data transfers). The total amount of data moved to and from these providers for the whole evaluation was more than 100 GBs by utilizing freemium storages. Regarding reliability of the considered Cloud services, we also measured the number of failures experienced during up- and downloading the files. For Box.com we experienced a relatively high number of failures by downloading big files resulted in abortion of the transactions. On the other hand, SugarSync was proved to be the most reliable provider without a single failure.

Table 2. Data movements (in MBs) by Personal Cloud providers

Provider	Num. of Transactions	Num. of Failures	Uploaded	Downloaded	Sum
Google Drive [14]	1072	4	12100	12090	24190
Dropbox [15]	1106	8	11800	11800	23600
SugarSync [17]	567	0	4420	4415	8835
Box.com [18]	1014	120	14520	6570	21090

5.2 Data Distribution Evaluation

Based on the results of the evaluation of the MeasureTool component, our initial hypothesis that service quality levels differ for various Cloud providers has been proven. Now we continue with the evaluation of our proposed autonomous file distribution solution.

In Section 4 we have introduced how the DistributeTool component works for a sample configuration based on aggregated historical performance values, latest performance values and their predefined ratio. In this subsection we evaluate the performance of our proposed application with 4 different configurations (i.e. $r = 0, 0.1, 0.5, 0.9$) for user data distribution for the same set of files represented by the 6 cases introduced in the previous section, spread over the interconnected Personal Clouds. The computed values for these configurations are depicted in Figure 6.

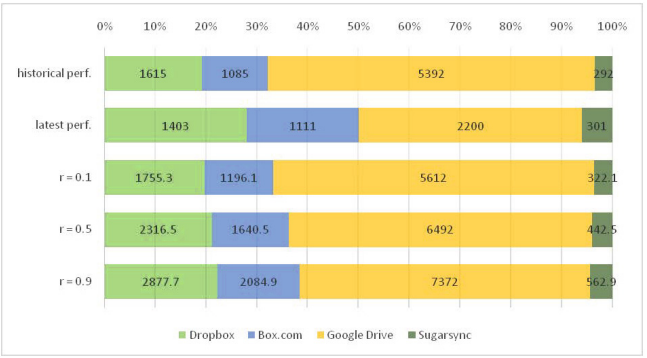

Fig. 6. Configurations for data distribution

During these measurements the DistributeTool component performed the splitting and packaging of the user files, selecting providers for the created file chunks based on the performance values and configurations, and uploading the files to these providers. The retrieval of the files was performed by the CollectTool component by using the meta-data description file created by the DistributeTool component. The average transfer speeds during the evaluation for the considered providers is shown in Figure 7 – which correlates to the ones gathered in the previous subsection. Furthermore we can also observe that transfer speeds achieved by our application by utilizing all providers are faster than single utilization of three providers (only Google Drive performs better alone).

The final evaluation results for the different configurations are shown in Figure 8. As we can see on this diagram, slight modifications on the ratio of historical and latest performance values (e.g. changing r from 0 to 0.1) do not imply big differences, but relying more on the latest performance values (i.e. using $r = 0.5$) results in faster uploading times for the overall user data.

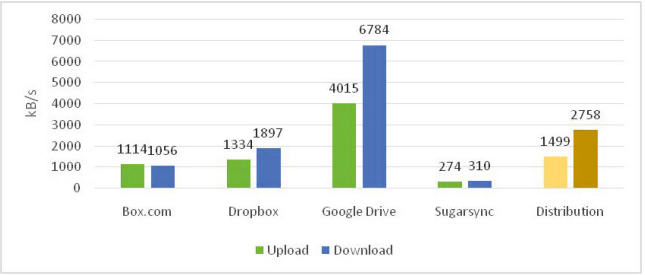

Fig. 7. The measured speed of Cloud providers during the evaluation

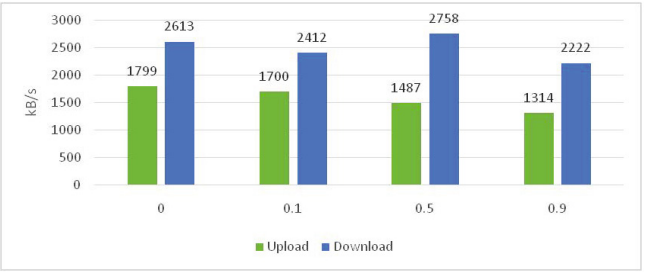

Fig. 8. Evaluation results for the proposed application with different configurations

6 Conclusion

The enormous data users produce with mobile devices are continuously posted to online services, may require the use of several Cloud storage providers at the same time to efficiently store and retrieve these data. The aim of our research in this paper was to develop a solution that unites and manages separate Personal Clouds in an autonomous way to provide a suitable solution for these needs. We have introduced our proposed application consisting of three components responsible for monitoring providers, managing and distributing user data to these providers, and retrieving user files in an autonomic way. Finally we evaluated our approach by utilizing four real Cloud providers, and concluded that our solution is capable of managing user data in a unified storage over these providers in an autonomous way, and still provides a good performance as well.

Our future work aims at further examining the configuration capabilities of our proposed application, and extending it with other service quality metrics, and investigating replication mechanism to eliminate dependability, and incorporate additional provider plugins to widen provider support.

Acknowledgment. The research leading to these results has received funding from the CloudSME FP7 project under grant agreement 608886.

References

1. Bozman, J.: Cloud Computing: The Need for Portability and Interoperability. IDC Executive Insights (August 2010)
2. Dillon, T., Wu, C., Chang, E.: Cloud Computing: Issues and Challenges. In: Proc. of the 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 27–33 (2010)
3. Drago, I., Mellia, M., Munafo, M.M., Sperotto, A., Sadre, R., Pras, A.: Inside Dropbox: Understanding Personal Cloud Storage Services. In: Proceedings of the 2012 ACM Conference on Internet Measurement Conference (IMC 2012), pp. 481–494. ACM, New York (2012)
4. Drago, I., Bocchi, E., Mellia, M., Slatman, H., Pras, A.: Benchmarking personal cloud storage. In: Proceedings of the 2013 Conference on Internet Measurement Conference (IMC 2013), pp. 205–212. ACM, New York (2013)
5. Fraunhofer Institute for Secure Information Technology. On THE Security of Cloud Storage Services, SIT Technical reports (March 2012), http://www.sit.fraunhofer.de/content/dam/sit/en/documents/Cloud-Storage-Security_a4.pdf
6. Gagliardi, F., Muscella, S.: Cloud Computing – Data Confidentiality and Interoperability Challenges. In: Cloud Computing. Computer Communications and Networks, pp. 257–270. Springer, London (2010)
7. Garcia-Tinedo, R., Sanchez-Artigas, M., Moreno-Martinez, A., Cotes, C., Garcia-Lopez, P.: Actively Measuring Personal Cloud Storage. In: The 6th IEEE International Conference on Cloud Computing (Cloud 2013), pp. 301–308 (2013)
8. Jeffery, K., Neidecker-Lutz, B.: The Future of Cloud Computing, Opportunities for European Cloud Computing beyond 2010. Expert Group Report (January 2010)
9. Kertesz, A.: Characterizing Cloud Federation Approaches. In: Mahmood, Z. (ed.) Cloud Computing - Challenges, Limitations and R&D Solutions. Springer Series on Computer Communications and Networks (accepted in 2014)
10. Kertesz, A., Varadi, S.: Legal Aspects of Data Protection in Cloud Federations. In: Nepal, S., Pathan, M. (eds.) Security, Privacy and Trust in Cloud Systems. Springer, Signals & Communication, pp. 433–455 (2014)
11. Wikipedia, Freemium (2014), http://en.wikipedia.org/wiki/Freemium
12. Wikipedia, OAuth (April 2014), http://en.wikipedia.org/wiki/Oauth
13. The SZTAKI Cloud project website (May 2014), http://cloud.sztaki.hu/en/home
14. Google Drive (May 2014), https://drive.google.com/
15. Dropbox (May 2014), https://www.dropbox.com/
16. Wikipedia – Dropbox (April 2014), http://en.wikipedia.org/wiki/Dropbox_%28service%29
17. SugarSync (May 2014), https://www.sugarsync.com/
18. Box.com (May 2014), https://www.box.com/