This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE P2P 2012 proceedings

FRIENDBOX: A *Cloudified* F2F Storage Application

Adrián Moreno-Martínez, Raúl Gracia-Tinedo, Marc Sánchez-Artigas and Pedro Garcia-Lopez
Universitat Rovira i Virgili, Tarragona (Spain)
{adrian.moreno|raul.gracia|marc.sanchez|pedro.garcia}@urv.cat

***Abstract*—Personal storage is a mainstream service used by millions of users. Among the existing alternatives, Friend-to-Friend (F2F) systems are aimed to leverage a secure and private off-site storage service. However, the specific characteristics of these systems (reduced node degree, correlated availabilities) represent a hard obstacle to their performance.**

We present FriendBox: a hybrid F2F personal storage system that combines resources of trusted friends with Cloud storage for improving the service quality achievable by pure F2F systems.

I. INTRODUCTION

Increasingly, users will require ubiquitous *personal storage* to handle their ever-growing digital lives. To meet this demand, the number of Cloud offerings is large and growing, ranging from Dropbox and the likes, to infrastructure providers like Amazon. However, some users are reluctant to move their data to the Cloud due to the large amount of control ceded to Cloud vendors. In consequence, other alternatives have been devised to provide a *secure and private* off-site storage service.

In this sense, Friend-to-Friend (F2F) storage systems constitute an alternative approach to leverage personal storage [1], [2]. The F2F paradigm is based on the synergy between social networks and storage systems: users store their data in a set of social friends. Thus, data is neither stored in a centralized server nor in unknown peers.

Nevertheless, the main drawback of F2F systems lies on the poor service quality they can provide. Actually, our research suggests that it is extremely difficult for a F2F system to provide adequate *data availability* and *transfer times* guarantees [3]. Specifically, there are two simultaneous issues that put the feasibility of F2F systems at risk:

- **Reduced node degree**. This refers to the *typically small number of available trusted friends to which store data*. For instance, over 63% of Facebook users have less than 100 friends [4]. Furthermore, most interactions occur only across a small subset of their social links.
- **Availability correlations**. Real measurements on user behavior from online social networks suggest that friends are significantly correlated in their connectivity patterns [5]. This implies that is rather probable to find all friends of a user simultaneously offline, particularly during night hours.

In this landscape, we propose a radically different approach: *resorting to Cloud storage services* for improving the performance of a F2F storage system.

To meet this need, we developed `FriendBox`: a hybrid F2F personal storage system. `FriendBox` is a F2F system that combines the resources of trusted friends with Cloud storage for improving storage service quality while preserving privacy. `FriendBox` provides a flexible and user-defined Cloud usage: *users are able to decide where to store their data*, which can be completely on friends, only in the Cloud or in a mix of them.

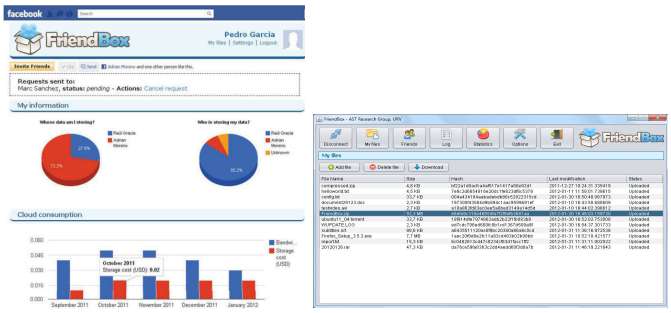

Fig. 1. On the left, the `FriendBox` Social Front-end integrated in Facebook; on the right, the Desktop Storage Client.

II. FRIENDBOX

`FriendBox` is divided into three components: `FriendBox` Social Front-end, `FriendBox` Storage Client and `FriendBox` Application State.

A. `FriendBox` Social Front-end

The Social Front-end of `FriendBox` is integrated within Facebook (see Fig. 1 left) and provides three main services: social relationships provisioning, access control and service information.

The fact of coupling `FriendBox` within a social network facilitates the establishment of social storage relationships among users, a critical aspect for a F2F system.

User management in a massive distributed application is a complex task. For this reason, we employ a social front-end as entry point of `FriendBox`: only those users which are members of the social network are capable of accessing to our F2F system. Thus, user management and access control issues are partially delegated to the social network avoiding additional complexity to the storage system.

Finally, the `FriendBox` Social Front-end provides a very intuitive GUI to expose the most relevant service information to the user. For instance, users can inspect their monthly Cloud resource consumption and the distribution of their data within the system.

B. `FriendBox` Storage Client

The heart of the `FriendBox` Storage Client (see Fig. 1 right) is implemented in Erlang, a programming language conceived to build massive distributed systems. Furthermore, its OTP framework permits to easily extend `FriendBox`'s functionalities by introducing independent modules. For example, currently the software includes two ways of providing data redundancy: replication and Reed-Solomon erasure codes (EC), but other mechanisms can be easily added.

Users must download the client in order to connect to the system. This software enables users to perform basic data operations, such as storing and retrieving files from the system (friends and/or Cloud). Moreover, this software acts

978-1-4673-2862-3/12/$31.00 ©2012 IEEE

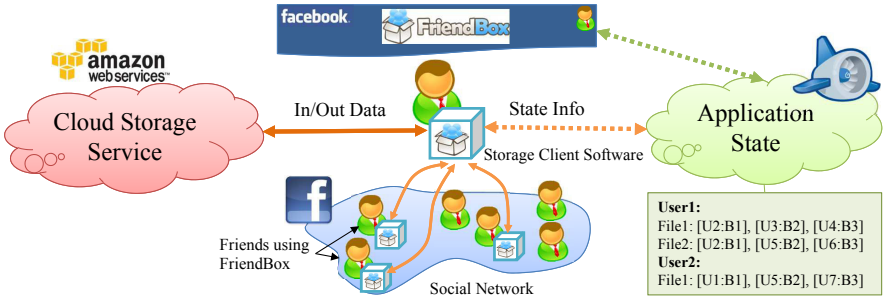

Fig. 2. Overview of `FriendBox` architecture. This picture illustrates how a user maintains storage links with some of his friends in a social network. Moreover, this user is able to store a fraction of his data in a Cloud storage service. The state information of a user's data is stored in the `FriendBox` Application State. Finally, users manage their storage relationships and check the state of their storage service in the `FriendBox` Social Front-end.

both as a client and a server in that our friends can store their information in the storage space we contribute to the system.

To enable communication among friends over the Internet, we implemented the Universal Plug and Play (UPnP) NAT traversal mechanism, which is supported by most home gateways.

The `FriendBox` Storage Client enables users to decide the distribution of data between their friends and the Cloud. This leaves to the user the control of his data and its placement within the system. In this line, the architecture of `FriendBox` supports simultaneous connections with several Cloud providers. This attractive feature opens the door for the development of new strategies to enforce privacy and avoid data lock-in.

In Fig. 2 we observe how a user distributes his data among friends and the chosen Cloud storage service.

Additionally, the `FriendBox` Storage Client can be used as a tool to perform simulations.

C. `FriendBox` Application State

This component has been hosted in Google App Engine. We employed this platform for hosting our application since it offers an elastic and scalable service.

Essentially, the `FriendBox` Application State maintains updated the data management information about users' files. This information expresses which friends store which files and the network address of each friend. The maintenance process is carried out by `FriendBox` Storage Clients installed at participants. Clients communicate with the `FriendBox` Application State and update their state information via a Representational State Transfer (REST) API, accessed through an *authorizing token*. In addition, this application is the backend of the `FriendBox` Social Front-end.

The role of the `FriendBox` Application State is illustrated in Fig. 2. In this figure, we show how a user communicates with the Application State to transfer state information.

III. Experiments

We evaluated `FriendBox` through a real deployment in our university campus with a group of ten users that are friends among them in Facebook. In our scenario, users perform and serve storage operations in the system.

The main parameters of our test are: file size of 200MB, EC original blocks $k = 40$ and a redundancy degree of 2.5. Note that nodes only need to gather k blocks to reconstruct the original file. The fraction of data stored in the Cloud (Amazon S3) $F_C = 0.5$ (20 blocks).

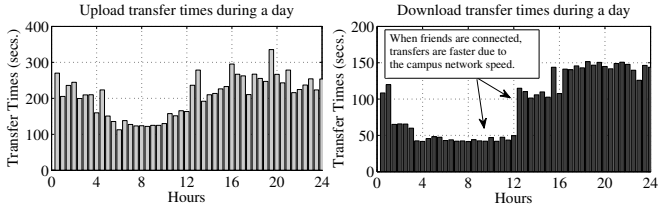

Fig. 3. Upload and download transfer times along one day

As can be seen in Fig. 3, the transfer times are faster during the periods of higher node availability. This means that accessing to friends in our campus is faster than resorting to the Cloud. This is due to the performance of the campus network (100Mbps).

Thanks to the use of the Cloud, storage operations are still possible even when friends are offline.

IV. Conclusions

In this work, we resort to a Cloud backend to overcome the limitations of pure F2F systems for providing a secure and private personal storage service.

In `FriendBox` we implemented hybrid data transfer scheduling and data availability mechanisms to mitigate the drawbacks of pure decentralization. We validated `FriendBox` in a deployment in our university campus.

As a result of this research, we will release in short the first fully-equipped `FriendBox` version, in order to provide a flexible and private Internet-scale personal storage service[1].

Acknowledgements

This work has been partly funded by the Spanish Ministry of Science and Innovation through projects DELFIN (TIN-2010-20140-C03-03) and RealCloud (IPT-2011-1232-430000).

References

[1] J. Li and F. Dabek, "F2f: Reliable storage in open networks," in *IPTPS'06*, 2006.

[2] D. N. Tran, F. Chiang, and J. Li, "Friendstore: cooperative online backup using trusted nodes," in *SocialNets'08*, 2008, pp. 37–42.

[3] R. Gracia-Tinedo, M. Sánchez-Artigas, A. Moreno-Martínez, and P. García-López, "FRIENDBOX: A Hybrid F2F Personal Storage Application," in *IEEE CLOUD'12*, 2012, p. To appear.

[4] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *EuroSys'09*, 2009, pp. 205–218.

[5] S. A. Golder, D. M. Wilkinson, and B. A. Huberman, "Rhythms of social interaction: Messaging within a massive online network," in *Communities and Technologies*, 2007, pp. 41–66.

[1] `FriendBox` Project Web page: http://ast-deim.urv.cat/friendbox