My Research Activity: P2P File Backup System

Alessandro Duminuco
EURECOM
Sophia Antipolis, France
duminuco@eurecom.fr

***Abstract*—This is a short report illustrating my research activities. They are the results of my work at EURECOM as PhD student and they are mainly related to my PhD thesis. My thesis focuses on the various aspects concerning the design and the implementation of a file backup system based on a Peer-to-Peer approach.**

I. Introduction

The objective of my PhD thesis is the study of various issues related to the the design and the implementation of a Peer-to-Peer backup System. Peer-to-Peer systems have the interesting property of self-scaling, which means that the amount of resources grows with the number of participants.

P2P(Peer-to-Peer) systems have received a lot of attention in recent years. In particular, the research community has shown an increasing interest in the use of P2P systems for file storage and backup [1], [2], [5], [6]. Typically, file backup is done in a purely centralized manner. Such an organization requires a large amount of resources (disks, tape robot) and also some human intervention. On the other hand there is an increasing number of PCs each equipped with a local disk with a capacity of tens of Giga Bytes.

The goal of the thesis is to investigate how the local disks of a large number of PCs can be organized in such a manner as to allow a highly reliable file back-up system. The thesis first studied existing approaches for distributed file backup and then aims at designing and implementing its own backup system. It is envisioned to distribute each file that is backed up over multiple machines and to use error correcting codes for loss recovery. In order to evaluate the different design choices, a system model (machine availability, etc) has been defined and a performance evaluation will be carried out.

II. Specific Areas of Interests

The design of a complete peer-to-peer file backup system is a complex operation that covers several aspects. Our interests focus more specifically on the reliability of data. In particular we put our attention on the redundancy schemes and on the mechanisms adopted to monitor and maintain such redundancy. The next two subsections present our most relevant contributions, while section II-C briefly describes my activity during a 3-month internship at Microsoft Research Cambridge (UK).

A. Proactive Replication using Machine Availability Estimation

Distributed storage systems provide data availability by means of redundancy. To assure a given level of availability in case of node failures, new redundant fragments need to be introduced.

Since node failures can be either transient or permanent, deciding when to generate new fragments is non-trivial. An additional difficulty is due to the fact that the failure behavior in terms of the rate of permanent and transient failures may vary over time. To be able to adapt to changes in the failure behavior, many systems adopt a reactive approach, in which new fragments are created as soon as a failure is detected. However, reactive approaches tend to produce spikes in bandwidth consumption.

Proactive approaches create new fragments at a fixed rate that depends on the knowledge of the failure behavior or is given by the system administrator. However, existing proactive systems are not able to adapt to a changing failure behavior, which is common in real world.

We proposed in [4] a new technique based on an ongoing estimation of the failure behavior that is obtained using a model that consists of a network of queues. This scheme combines the adaptiveness of reactive systems with the smooth bandwidth usage of proactive systems, generalizing the two previous approaches. Now, the duality reactive or proactive becomes a specific case of a wider approach tunable with respect to the dynamics of the failure behavior.

B. Hierarchical Codes

As already mentioned, redundancy is the basic technique to provide reliability in storage systems consisting of multiple components. A redundancy scheme defines how the redundant data are produced and maintained. The simplest redundancy scheme is replication, which however suffers from storage inefficiency. Another approach is erasure coding, which provides the same level of reliability as replication using a significantly smaller amount of storage.

When redundant data are lost, they need to be replaced. While replacing replicated data consists in a simple copy, it becomes a complex operation with erasure codes: new data are produced performing a coding over some other available data. The amount of data to be read and coded is d times larger than the amount of data produced. This implies that coding has a larger computational and I/O cost, which, for distributed storage systems, translates into increased network traffic.

Peer-to-Peer storage systems are conceived to exploit commodity hardware. Also, while the participating peers typically have ample storage and CPU power, their network bandwidth may be limited. For these reasons existing coding techniques are not suitable for P2P storage and system designers usually prefer replication notwithstanding its storage inefficiency.

In [3] we explored the design space between replication and the existing erasure codes. We proposed and evaluated a new class of erasure codes, called Hierarchical Codes, which aims at finding a flexible trade-off that allows to reduce the network traffic due to maintenance without losing the benefits given by traditional codes.

C. Peer-to-Peer Backup System Prototype

During a 3-month internship at Microsoft Research (Cambridge,UK), I designed and developed a prototype of a Peer-to-Peer Backup System, using Network Coding. This work had two objectives: (i) showing the feasibility of storage system based on network coding and (ii) testing the features and the tools offered by the *Microsoft .NET Framework* to develop Peer-to-Peer applications.

III. Future Work

We are currently working on extensions of Proactive Replication and Hierarchical Codes. Moreover we plan to develop a more complete prototype of a P2P file backup system that implements the techniques we proposed. The idea is to develop a mixed system, in which some functionalities are simulated, while some others are implemented as in a real environment. This will allow us to test the aspects we are interested in a realistic scenario, without putting to much effort in all the implementation details.

References

[1] A. Adya, W. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer. Farsite: Federated, available and reliable storage for an incompletely trusted environment. In *5th Symposium on OSDI 2002*, 2002.

[2] F. Dabek et al. Wide-area cooperative storage with CFS. In *Proc. SOSP 2001*, Oct. 2001.

[3] A. Duminuco and E. Biersack. Hierarchical codes: How to make erasure codes attractive for peer-to-peer storage systems. Technical report, EURECOM, 2008. submitter to P2P 2008.

[4] A. Duminuco, E. Biersack, and T. En-Najjary. Proactive replication in distributed storage systems using machine availability estimation. In *Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2007.

[5] A. Haeberlen, A. Mislove, and P. Druschel. Glacier: Highly durable, decentralized storage despite massive correlated failures. In *NSDI05*, 2005.

[6] H. Weatherspoon. *Design and Evaluation of DistributedWide-Area Online Archival Storage Systems*. PhD thesis, University of California, Berkeley, 2006.