Email-based File System for Personal Data

Jagan Srinivasan[†], Wei Wei[*], Xiaosong Ma[‡*], Ting Yu[*]
[†]*EMC*[2], [*]*North Carolina State University,* [‡]*Oak Ridge National Laboratory*

ABSTRACT

Email services have been offering growing email storage capacity, reliable service, and powerful search capability, making them appealing as storage resources. In this paper, we present EMFS, which aggregates back-end storage by establishing a RAID-like system on top of virtual email disks formed by email accounts. By replicating data across accounts from different service providers, highly available storage services can be constructed based on already reliable, cloud-based email storage. EMFS provides a POSIX-like file system interface that allows ubiquitous data access. We have implemented a prototype of EMFS and conducted experiments in a campus network. Our results indicate that while EMFS cannot match the performance of highly optimized distributed file systems such as NFS and AFS, it performs quite closely to JungleDisk, a commercial cloud storage solution.
[1]

1. INTRODUCTION

Recently, cloud storage has received great attention for being an attractive solution to deliver storage as a service with scalability, reliability and cost-effectiveness. We have seen the emergence of many commercial solutions that provide different access interfaces to the back-end cloud storage. For example, Google Docs and Adobe Buzzword not only provide a variety of applications such as word processing and spread sheet, but also offer online storage for file backup. Meanwhile, Jungle Disk and Dropbox provide a generic file system interface for cloud storage access. However, they either tightly bind cloud storage with specific applications, where migration between service providers could present a challenge, or they solely rely on one underlying cloud provider, which cannot yet guarantee high data reliability/availability.

Rather than using separate cloud storage services, we believe that email service offers an appealing solution to personal cloud storage for several reasons. First, the capacity of a single email account has increased dramatically in recent years. For example, currently Google Gmail provides 7.4 GB, while Yahoo! and AOL Mail even provide unlimited storage. Second, email services are provided by many reliable and reputable online service providers, such as Google and Yahoo. Further, besides reliability, email services, even for uncharged accounts, are rather stable and long-lasting. Admittedly, email services are not immune to technical failures, as magnified by the recent incident of Hotmail. However, as users can easily obtain multiple accounts from different providers, replication techniques can be more naturally adopted for better reliability. Further, email-based cloud storage allows mail service providers to leverage their existing infrastructure to build light-weight value-added storage services and to utilize existing features such as data deduplication and personalized advertising.

In this paper, we propose EMFS, a personal cloud storage solution based on email services. EMFS is novel in that it views email accounts as virtual disks and employs RAID-like approaches for space aggregation, data striping, and data replication. We examined the feasibility of using email transfer protocols for general-purpose file access and providing traditional file system interfaces, and addressed many unique design challenges and issues, such as anti-spam usage restrictions, metadata and file data organization, and data placement. We implemented a prototype of EMFS via FUSE [2]. We evaluated this prototype with comprehensive experiments in a campus network. Our results indicate that while EMFS cannot match the performance of highly optimized distributed file systems such as NFS and AFS, it performs quite closely to JungleDisk, a commercial cloud storage solution.

2. SYSTEM OVERVIEW

Design Goals. We aim at building a personal cloud storage on top of email services. Our design goals include usability (generic file system interface), scalability (allowing the personal storage space to grow by adding more email accounts and handling growing number of files as time goes by), and reliability (continuous access despite failures of individual email services).

EMFS Architecture. EMFS is composed of two layers: the EMFS client and the email storage cloud. The email storage cloud treats email accounts as virtual disks. Such virtual disks can further be organized into a RAID system. By striping and replicating data across these "email disks", especially accounts with different service providers, we receive benefit in several aspects: performance, reliability, and capacity. The EMFS client presents an POSIX-like file system interface for the email storage cloud, which enables existing applications to run on top of EMFS without any modification.

3. SYSTEM DESIGN

[1]Wei is the only student, and Dr.Ma will present the poster.

Data Organization and Access. Data and metadata in EMFS are stored as contents of emails, either as attachments or as part of the body of the emails. There are two types of emails, metadata emails and data emails. As their names indicate, a metadata email stores metadata for files and directories, and a data email is used to stored file data. A metadata is stored in the body of an email, while its unique identifier is stored in the subject line; data is stored as attachments of emails. These emails can be sent to and received from email servers through standard protocols such as IMAP, SMTP and POP. We finally choose IMAP over SMTP and POP for several reasons. First, IMAP is not restricted by the spam policies enforced by email service providers (as uploading an email to a service provider through IMAP does not count as sending an email. Instead, it is only considered as saving a draft to an email account). Second, IMAP provides better performance than SMTP when uploading emails to servers. Third, IMAP is more powerful than POP. For example, it supports multiple client connections to the same mailbox, and access to individual parts of a message.

File Striping and Data Replication. EMFS views each email account as a virtual disk and builds a RAID systems on top of a group of such email disks. In our prototype implementation, we experimented with simple striping and mirroring strategies. Data blocks are striped across a group of email accounts to improve the aggregate throughput. EMFS further employs replication to mirror data across multiple of email account groups from different providers as it is highly unlikely that multiple service providers experience down time concurrently.

We experiment with two strategies for data accessing: (1) Read-one and Write-one: reads and writes from EMFS go to the same account that acts as a primary account. Other accounts are not used in file accesses unless the primary account fails; (2) Read-fast and Write-fast: reads and writes go to different accounts based on their upload and download speeds. This is based on our observation that some email services provide better performance for reads and others for writes. In both strategies, replication occurs lazily to all accounts. Because of the use of local caching, the Read-fast and Write-fast strategy will not cause inconsistency between reads and subsequent writes even though they go to different accounts.

Consistency and Failure Recovery. Considering the append-only nature of email services, EMFS adopts a mechanism similar to that used in LFS [1] to ensure the atomicity of updates. Whenever a file needs to be written to the server, one metadata email and as many data emails as needed, are sent out to the email server at the same time. This metadata email has a "status" field set for the file that is transferred, which indicates that the file is dirty. Once the system receives confirmation that the data blocks have been transferred successfully, it sends out another metadata email with the status field cleared. Such a pair of metadata emails help EMFS check the consistency of files and roll back if necessary.

4. EXPERIMENTAL EVALUATION

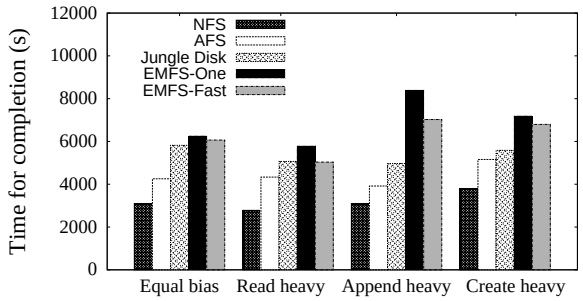

Figure 1: Postmark performance

We have implemented EMFS with two replication strategies - Read-One-Write-One (EMFS-One) and Read-Fast-Write-Fast (EMFS-Fast), in around 3000 lines of Python code via FUSE. We conducted our experiments in a campus network, and compare EMFS with NSF, AFS and Jungle Disk, a commercial cloud storage solution. Both NFS and AFS servers were configured on dedicated machines inside the campus network. Figures 1 shows the results for the Postmark benchmark, which were configured to use 200 files, 200 transactions, with file sizes ranging between 4 KB and 16 MB. We generated four workloads by varying the operation bias. The results show that EMFS offers comparable performance to Jungle Disk, especially for balanced or read-heavy workloads.

5. CONCLUSION

We presented EMFS, a personal cloud storage solution, which provides cost-effective, efficient, and highly available storage by leveraging the cloud infrastructure of leading email service providers, viewing email accounts as virtual disks and applying techniques such as striping and replication. Though email protocols are not designed for file transfer, we have found that EMFS achieves a significant fraction of NFS/AFS performance (with the latter running on dedicated servers within local networks) and approximately matches or outperforms Jungle Disk, a commercial cloud storage service.

6. REFERENCES

[1] Mendel Rosenblum and John K. Ousterhout. The design and implementation of a log-structured file system. *ACM Trans. Comput. Syst.*, 10(1):26–52, 1992.

[2] Miklos Szeredi. File system in user space. *http://fuse.sourceforge.net/*, 2006.