User-Assisted Cloud Storage System: Opportunities and Challenges

Xiaowen Chu, Hai Liu, Yiu-Wing Leung
Hong Kong Baptist University
Hong Kong, China
{chxw, hliu, ywleung}@comp.hkbu.edu.hk

Zongpeng Li
University of Calgary
Canada
zongpeng@ucalgary.ca

Min Lei
Beijing University of Posts and Telecommunications, China
byleimin@tom.com

1. Introduction

Cloud storage has recently attracted a substantial amount of attention from both industry and academia. Notable commercial cloud storage services include Amazon S3, Google Drive, Dropbox, Microsoft Skydrive, and Apple's iCloud. Compared with traditional storage systems, cloud storage offers several desirable advantages, including high data availability, high data reliability, and dynamic storage space.

To provide highly available and reliable storage service, cloud storage providers (CSPs) make tremendous investments in storage hardware and network bandwidth. To be profitable, CSPs need to recover these costs from reasonable charges to cloud users. At present, the typical charges include storage ($0.05-0.10 per GB/month), network traffic ($0.05-0.20 per GB), and requests ($0.01 per 1,000 or per 10,000 requests/month, depending on the request type). It is apparent that the major cost comes from storage devices and network bandwidth. When the number of users grows into the millions and the data volume into the exabyte scale, it becomes very critical for CSPs to decrease costs while maintaining the same level of data availability and reliability.

In traditional cloud storage systems, all of the resources are located at data centers. In this paper, we present the opportunities and challenges of a user-assisted cloud storage architecture that aims to reduce the cost of providing highly available and reliable cloud storage services by exploiting the underused storage and network resources of cloud users.

2. Opportunities: User-Assisted Architecture

The envisioned structure of the user-assisted cloud storage system is illustrated in Fig. 1. The CSP invests in several distributed data centers and exploits geo-redundancy to offer continuous service even under extreme cases like natural disasters or power grid failures. The CSPs use the available resources from cloud users as an added asset to augment their own data center resources. This is economically efficient because (1) most users have huge amounts of spare storage space on their PCs and (2) most users pay a fixed monthly fee for their broadband Internet connections. The main motivation behind using the storage space and network bandwidth of cloud users is that it helps CSPs save on both hardware and bandwidth costs. Compared with traditional centralized architecture, user-assisted architecture is more scalable in resource provisioning and hence more cost-effective.

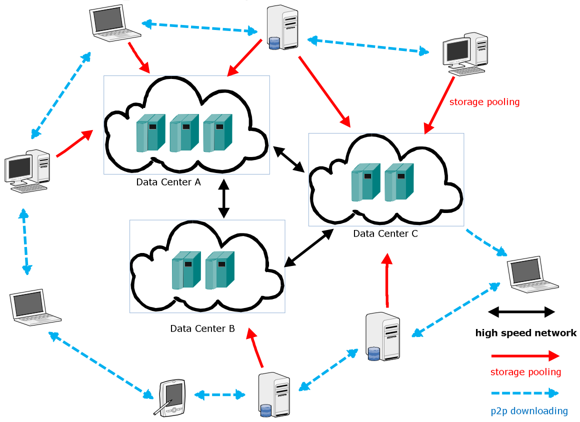


Figure 1 Architecture of user-assisted cloud storage system.

In this user-assisted architecture, CSPs provide data redundancy within their data centers to guarantee high data availability and reliability. The storage services should be resistant to all kinds of failure and system maintenance, such as disk/node/rack failure, power distribution unit failure, and even the failure of an entire data center. CSPs also actively distribute encoded data blocks to cloud users to the extent that each cloud user can download a large portion of his data from other users, hence reducing the network bandwidth costs significantly. Furthermore, a cloud user can better use his network bandwidth by simultaneously uploading/downloading data blocks to/from many other users.

3. Challenges

Although our user-assisted architecture is attractive and promising, it presents several challenges that must be overcome.

Incentive Design: An effective incentive scheme that can motivate cloud users to contribute storage space and outbound bandwidth is vital to the success of our user-assisted architecture. Private BitTorrent communities are a successful example of using a good incentive design to incent end users to contribute as much as possible [1]. These communities deploy a sharing ratio enforcement (SRE) mechanism to overcome the free-riding issue of traditional BitTorrents. SRE forces registered users to keep their upload-to-download ratios higher than a predefined threshold. A registered user is banned from a private BitTorrent community if his sharing ratio is lower than the threshold. In contrast, a user with a higher sharing ratio is rewarded.

In our user-assisted architecture, end users should be incented to contribute some of their storage space, and more importantly their outbound bandwidth. We must design a new incentive scheme, different from the upload-to-download ratio used in private BitTorrent communities, that can achieve a good balance among the cost of offering the service, the revenue collected from all users, and the reward to the contributing users. A simple design involves returning a portion of the storage and bandwidth cost savings to contributing users in proportion to the effective storage and upload traffic contributed by each user. How to reliably obtain true data storage and uploading traffic statistics without too much overhead is a challenging problem. Collusion attacks present another problem, as some malicious users may collaborate to generate upload traffic and thus receive more reward.

Availability and Reliability: Users are willing to pay for cloud storage because of its high availability and reliability. The cloud service should not be interrupted by any kind of system failure or even natural disasters, and the users' data should never be corrupted. Studies on improving reliability and availability by adding data redundancy have been very active. The simplest form of introducing data redundancy is replication, i.e., storing multiple copies of the original data item, as exemplified by the Google File system [2] and Amazon's S3 [3]. Data replication is simple but inefficient. A more cost-effective way of achieving the same reliability with much less data redundancy is to apply erasure coding [4, 5]. As an example, Windows Azure Storage uses Reed-Solomon (RS) code to reduce the storage redundancy level from 3x to 1.3x-1.5x [5]. Under erasure coding, the original data item is partitioned into k blocks, from which n ($n>k$) encoded blocks are generated for distributed storage. RS code is an example of maximum distance separable (MDS) codes, which possess a regeneration property: any k out of the n encoded blocks can be used to recover the original data item. In a distributed environment, while storage overhead remains a critical concern for system efficiency, the amount of data transfer that is required to replace a lost storage node (i.e., the repair traffic) becomes equally important. With MDS codes, when repairing a lost node u by regenerating its storage onto a new node v, v must first download enough blocks to recover the original data item. Regeneration codes were recently invented based on the concept of network coding to minimize the repair traffic while holding the regeneration property of MDS codes [6-8].

Previous studies have focused on the design of a single coding scheme for the storage system. We argue that different coding schemes should be used at different levels in our user-assisted architecture to handle different types of failures. First, within a storage node with multiple disks, RAID5 or RAID6 are good candidates for handling one or two disk failures. Second, more powerful erasure or regeneration codes should be used across the data center to handle node failures, rack failures, and regular node updates. A good understanding of the pattern of node failures is very important. There exists a tradeoff between the level of data redundancy and the volume of repair traffic, so the network bandwidth between storage nodes should also be considered when designing the coding scheme. Third, to handle the failure of a whole data center, traditional erasure codes may not work well because the repair traffic is usually the same as the original data size (e.g., petascale) whereas the bandwidth between data centers is limited (e.g., gigascale). How to code the data and distribute them among data centers becomes an important optimization problem. Hu et al. recently proposed the application of functional minimum-storage regenerating (F-MSR) coding to handle two cloud failures while reducing repair traffic [9].

User Experience: The user's experience largely depends on how fast he can upload/download files. In traditional cloud storage, the end user transfers data to/from a data center, which may not be able to fully use the available bandwidth of the user's Internet access link. In our user-assisted architecture, because a user can simultaneously transfer data to/from many other users, there is a much greater chance that the available bandwidth is used to the fullest. How to encode and place data to improve overall uploading/downloading performance becomes a challenging research problem. Another factor worth considering is computational overhead. If a user's CPU is always busy handling the coding/decoding process, he may resist using the service. One promising solution is to offload the coding tasks to GPUs, which are widely available on desktop PCs and mobile devices. Our previous studies have shown that network coding can be practically implemented on contemporary GPUs with very high throughput [10, 11].

4. Conclusion

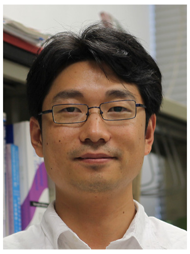
Cloud storage is a promising service that can offer economical and efficient solutions for highly reliable and accessible storage service. We propose using the spare storage and bandwidth resources of end users to save on service providers' storage and bandwidth costs.

How to design an effective incentive scheme, how to design different coding schemes at different levels, and how to enhance the user experience remain major challenges to this proposition.

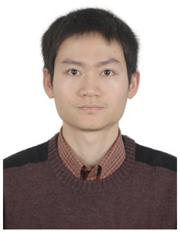
Acknowledgement
This work is supported by Hong Kong GRF grant HKBU 210412.

References

[1] X. Chen, Y. Jiang, and X.-W. Chu, “Measurements, Analysis and Modeling of Private Tracker,” in Proc. IEEE P2P 2010.
[2] S. Ghemawat, H. Gobioff, and S. Leung, “The Google file system,” in Proc. ACM SOSP 2003.
[3] DeCandia et al., “Dynamo: Amazon’s Highly Available Key-value Store,” in Proc. ACM SOSP 2007.
[4] R. Rodrigues and B. Liskov, “High availability in DHTs: Erasure Coding vs. Replication,” in Proc. IPTPS 2005.
[5] C. Huang et al., “Erasure Coding in Windows Azure Storage,” in Proc. USENIX ATC 2012.
[6] A. G. Dimakis et al., “Network Coding for Distributed Storage Systems,” IEEE Transactions on Information Theory, vol. 58, no. 9, Sep 2010.
[7] A. G. Dimakis et al., “A Survey on Network Codes for Distributed Storage,” IEEE Proceedings, vol. 99, no. 3, March 2011.
[8] K. W. Shum and Y. Hu, “Exact Minimum-Repair-Bandwidth Cooperative Regenerating Codes for Distributed Storage Systems,” in Proc. IEEE ISIT 2011.
[9] Y. Hu et al., “NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds,” in Proc. USENIX FAST 2012.
[10] X.-W. Chu, K. Zhao, and M. Wang, “Massively Parallel Network Coding on GPUs,” in Proc. IEEE IPCCC 2008.
[11] X.-W. Chu, K. Zhao, and M. Wang, “Practical Random Linear Network Coding on GPUs,” in Proc. IFIP Networking 2009.

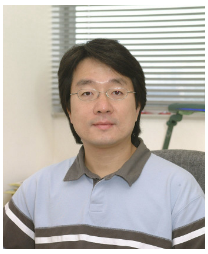


Xiaowen Chu received his B.E. degree in Computer Science from Tsinghua University, Beijing, P. R. China, in 1999, and the Ph.D. degree in the Computer Science from the Hong Kong University of Science and Technology in 2003. He is currently an Associate professor in the Department of Computer Science, Hong Kong Baptist University. His current research interests include Wireless Networks and Parallel and Distributed Computing.



Hai Liu received the BSc and MSc degrees in applied mathematics from South China University of Technology, in 1999 and 2002, respectively. He received the PhD degree in computer science from City University of Hong Kong in 2006. He is currently a research assistant professor with the Department of Computer Science, Hong Kong Baptist University. His research interests include wireless networking, mobile computing, and algorithm design and analysis.

Yiu-Wing Leung received the BSc and PhD degrees from the Chinese University of Hong Kong in 1989 and 1992, respectively. He has been working in the



Department of Computer Science of the Hong Kong Baptist University and now he is a full professor. His research interests include two major areas: 1) networking and multimedia which include the design and optimization of wireless networks, optical networks and multimedia systems, and 2) cybernetics and systems engineering which include evolutionary computing and multiobjective programming. He has published more than 70 journal papers in these areas, and most of which were published in various IEEE journals. He is a senior member of the IEEE.

Zongpeng Li received his B.E. degree in Computer Science and Technology from Tsinghua University (Beijing) in 1999, his M.S. degree in Computer Science from University of Toronto in 2001, and his Ph.D. degree in Electrical and Computer Engineering from University of Toronto in 2005. Since August 2005, he has been with the Department of Computer Science in the University of Calgary. In 2011-2012, Zongpeng was a visitor at the Institute of Network Coding, Chinese University of Hong Kong. His research interests are in computer networks, particularly in network optimization, multicast algorithm design, network game theory and network coding.

Min Lei is a lecturer at the School of Computer Science at Beijing University of Posts and Telecommunications (BUPT). Min received his Ph.D. degree in information security from BUPT. Prior to that, he received a M.S. degree in software engineering and theory in 2002 from BUPT and a BEng degree in computer science in 1999 from Nanchang University. Min’s research interests are watermarking, information hiding and information security.