Peer-to-Peer Backup System

Anshi Agarwal, Ayushi Bansal, Bhargav Sundararajan, Mridula Gupta
Computer Science @ UC Davis

02/09/2019

1 Introduction

1.1 Motivation

In this era of information and technology, we have all become dependent on data, and this data needs to be readily accessible all the time. However, we also need to make sure that we dont lose this important data because of device crashes, theft etc. With internet becoming faster and more accessible, Distributed File Systems have been a norm in todays age and time. While there are many options, centralized cloud storage is one of the most famous one. Many companies like Amazon S3, iCloud, and more offer these services but all rely on large data centers and hence, come with some cost.

The motivation of the peer to peer or P2P backup systems is that the free resources on various users can be shared to avoid these costs and it being decentralized allows direct access between peer computers. The main idea behind P2P backup system is that the users provide their own resources like storage space and get to use the backup space for their own files on others systems. P2P systems offer a decentralized, self-sustained, fault tolerant and symmetric network for effective backing up services. A study performed by Microsoft in 2008 showed that about 40% of Windows users have more than half of their hard disk free and thus could be excellent candidates for using a P2P backup system. Another study [1] showed that many users are not willing to pay the high fees for server-based backup and about half our participants said they would consider using P2P backup instead. Thus, there is definitely potential for P2P backup applications.

1.2 Related Work and Shortcomings

There has been considerable amount of research in the area of peer to peer backup systems. The primary objective of these systems has been to ensure reliability and security in these systems. Pastiche [2] and Samsara [3] are two examples that share with DIBS (Distributed Internet Backup System) the goal of decentralized peer-to-peer backup systems. In addition, [4] proposed a decentralized storage cluster architecture, CoStore, where nodes share equal responsibilities. pStore [5] is also one of the secure distributed backup system based on an adaptive peer to peer network. pStore uses the personal hard drive space attached to the internet for reliable and efficient data backup.

A significant amount of work has been done in the field of data encryption to ensure confidentiality in a peer to peer network that constitute an untrusted environment (e.g.[5] and [6]), as well as authentication [7], in order to ensure privacy.

The distributed system can also be categorized on the basis of file or chunk storage. In BitTorrent [8], files are split up into fixed-size chunks (on the order of a thousand per file), and the downloaders of a file barter for chunks of it by uploading and downloading them in a tit-for-tat-like manner to prevent parasitic behavior.

FreeNet [9] is an adaptive peer to peer system which ensures anonymity of authors, data locations and the readers and data is stored in Least Recently Used (LRU) fashion. These characteristics introduce reliability and performance limitation in the system and may result in disappearance of an unpopular file.

Most of the systems discussed above fail to provide high bandwidth requirements. Data redundancy is also a major issue in replication as the same file is stored in different nodes which also increases memory overhead. These systems usually have used encryption for data security which might allow unauthorized access to data through brute force or dictionary attacks. Performance can also get affected in these systems due to Denial of Service attack.

1.3 Challenges

Although peer-to-peer system to backup files is better in many aspects than the trivial distributed file system, still there are many problems and issues that need to be taken care for better implementation. Two main aspects of efficient peer-to-peer system are the data security and the speed of data transfer. Most of the existing P2P systems favor one aspect over other. P2P being decentralized makes it

difficult to determine its behavior, especially in case of discovering peers that belong to a new joining group. Churning which is caused by rapid joining and leaving of nodes is a major problem which most P2P systems fail to consider. Other issues that need to be handled are load balancing problem, methods for resource allocation, scalability, and anonymity to ensure resistance to censorship.

1.4 Suggested Approach

One of the main drawbacks of P2P backup systems is the high redundancy level as compared to centralized backup systems. This is because, as P2P systems are generally less reliable, we need to make sure a single node failure does not result in data loss. Hence, earlier versions of P2P backup systems used file replication to store the multiple copies of each file in different nodes. However, this has led to an increase in memory overhead and it directly affects the performance of the whole system. To address this, we need to reduce the redundancy level as much as possible and at the same time maintain the reliability of the system.

Erasure codes can be used to solve this problem. Erasure Coding breaks down a file into k fragments of equal size, encodes the fragments with redundant data thereby increasing the total number of fragments to n ($n > k$). The encoding is done in a way that we only need k fragments out of the total n to reconstruct the whole file. The increase in the number of fragments (m) is often less than the initial number of fragments (k). Therefore, the redundancy level in this case will always stay below 2. One of the main drawbacks of this approach is that it is CPU intensive. Also, when $m+1$ nodes fail, there wont be sufficient fragments to reconstruct the whole file. In such a scenario, the missing fragments need to be re-generated from the available ones. This will in turn affect the performance of the system. However, with an intelligent choice of m, this can be avoided, We also assume that such a scenario is extremely rare and the performance lost will not affect the system in the long run.

As much as reliability is important for a system, security is also crucial to ensure robustness for the system and privacy of the user data. For this, we plan to implement AES encryption to secure the data. Here, the owner of the file will encrypt the whole file before encoding is done by erasure codes and recipients can verify this using the owners public key.

Lastly, we must ensure fairness in sharing of disk space among all peers. We plan to borrow the fairness policy of Samsara [3], where a symmetric sharing of disk is proposed to ensure fairness without the need of trusted third parties. The space overhead for the system is further reduced through claim forwarding where claims can be chained to reduce redundancy.

2 Timeline

- 02/15 - (Data backup and storage technique) Try to finalize upon the various techniques to be used in designing our system. This includes the Data redundancy mechanism and packet transmission mechanism as it forms the backbone of many P2P backup system. We will divide work among us and start working on the implementation aspects of these techniques.

- 03/01 - (Data retrieval and repair technique) We will compile our work and start testing our system for storage and backup of files. Then for next phase we will start working upon the retrieval and repair aspect of backup system.

- 03/15 - (Data security and evaluating results) We will try to make our system more robust for data security and if time persists, will try to implement a model where all the nodes of peers are not trustworthy. In this phase, we will side-by-side start comparing our system with other systems and will start preparing results.

3 Anticipated Results

One of the main aspect of the evaluation will be to see if we are able to improve upon one of the state of art backup systems. As we highlighted, there are many advantages of using P2P systems but they all come with one or the other shortcomings. So we anticipate that our results should be based on the following parameters -

- Reliability - Our system should be more reliable than few of the standard P2P systems available and as we show that this system is prone to churning, this is the most important aspect of any backup system that we will consider.

- Storage Overhead - Our system should be able to perform the backup efficiently without creating too much storage space in the peer systems i.e. creating more replicas than required.

- Communication Bandwidth - One of the other important aspect that we want to consider is that we are efficiently using the network bandwidth. This is important because the replication of the large backup files along with some of the underlying integrity can lead to low bandwidth and can impact performance.

- Data Security - While we are still in discussion about the level of trustworthiness within the peer nodes at this point of the project, this will also be an important metric in our system. We want to make sure no node is able to make any changes to the backed up files, and handle the cases in case any such attempt has been made.
- Flexibility - The users should have the flexibility to control various aspects of the backup services. Some examples could be deciding the backup space to be utilized in one's system, deciding the online time they will be available considering the performance impact the backing up can take, the no. of replicas to be made based on the content, etc.

We plan to use the above mentioned parameters and compare our model with rest of the available modes and attempt to make improvement in at least some of the categories, if not all.

References

[1] S. Seuken, K. Jain, D. S. Tan, and M. Czerwinski, "Hidden markets: Ui design for a p2p backup application," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 315–324, ACM, 2010.

[2] L. P. Cox, C. D. Murray, and B. D. Noble, "Pastiche: Making backup cheap and easy," in *Proceedings of the 5th Symposium on Operating Systems Design and implementationCopyright Restrictions Prevent ACM from Being Able to Make the PDFs for This Conference Available for Downloading*, OSDI '02, (Berkeley, CA, USA), pp. 285–298, USENIX Association, 2002.

[3] L. P. Cox and B. D. Noble, "Samsara: Honor among thieves in peer-to-peer storage," *SIGOPS Oper. Syst. Rev.*, vol. 37, pp. 120–132, Oct. 2003.

[4] B. Cohen, "Incentives build robustness in bittorrent," 2003.

[5] C. Batten, K. Barr, A. Saraf, and S. Trepetin, "pstore: A secure peer-to-peer backup system," *Unpublished report, MIT Laboratory for Computer Science*, pp. 130–139, 2001.

[6] P. Gauthier, B. Bershad, and S. D. Gribble, "Dealing with cheaters in anonymous peer-to-peer networks," 2004.

[7] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with cfs," in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, SOSP '01, (New York, NY, USA), pp. 202–215, ACM, 2001.

[8] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bit torrent," in *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation*, NSDI'07, (Berkeley, CA, USA), pp. 1–1, USENIX Association, 2007.

[9] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley, "Protecting free expression online with freenet," *IEEE Internet Computing*, vol. 6, pp. 40–49, Jan. 2002.