Distributed File Storage Systems
Project Report 1

Bryan Dixon Brendan Kelly Mark Lewis-Prazen Andy Sayler
University of Colorado
first.last@colorado.edu

March 4, 2012

Abstract

The new generation of applications requires the processing of terabytes and petabytes of data. This processing has largely been achieved by distributed processing architectures and methodologies, driving the rise of companies such as Google, Amazon and Yahoo, which have embraced such technologies over the past decade and made them a central part of their business models. The increased importance of large scale data processing and a corresponding explosion in alternative storage architectures and service strategies requires computer science researchers and professionals to become well versed in the available alternatives and trade-offs in making technological architectural choices these areas. While storage technology has evolved significantly over the past five years, a deep understanding of both technical and business storage issues has lagged behind as many in these fields still view data storage as a mundane activity still the purview of back office technocrats. The current trend in distributed cloud storage and the focus of delivering IT infrastructure and applications bundled with data storage has led to an as a service model being used to promote such products.

The intent of our project is to examine the current state of both core storage and related service offerings on both a technical and business level to understand (1) what is technically available, (2) what are current research and developmental directions in storage technology, and (3) how core storage architectures are being coupled with associated services to produce bundled storage product offerings that purport to add value above and beyond core offerings. We believe that by performing this analysis and the subsequent implementation of a distributed proof of concept storage model, we will gain a better understanding of evolving storage technologies as well as the tradeoffs that need to be made when implementing bundled storage services in a production environment.

1 Problem Definition

Along with the infrastructure and network based applications, data storage is recognized as one of the major components of information technology systems. Functionally, storage typically services a wide range of requirements, spanning the spectrum from caching to archival needs. Combining networking and storage has created a platform with numerous possibilities allowing Distributed Storage Systems (DSS) to adopt roles which can vary considerably and may fall well beyond traditional data storage needs. Most DSSs as we understand them today were in their inception known as Distributed File Systems (DFS). Networking infrastructure and distributed computing share a close relationship. Advances in networking are typically closely followed by new distributed storage systems, which are designed to better utilize the enhanced capabilities of the network. Currently, the next generation Internet based systems are encountering numerous challenges, among them longer delays, unreliability, unpredictability and the potential for malicious behavior, all because of the need to operate in a public shared environment. To address this new set of challenges, innovative new storage architectures and algorithms are being suggested and developed, offering a myriad of improvements in areas such as security, consistency and routing. As storage systems continue to evolve, they naturally increase in complexity. Hence, the requisite expertise required to implement and operate them also increases dramatically.

Industry storage practices have evolved considerably over the past decade. The amount of data being actively managed continues to expand at around 20exceeding 50capacity every two to three years. Meanwhile, IT budgets are generally in decline. Why isnt industry-wide chaos rampant? In a word, this is due to consolidation efforts. Consolidation is being aided by continued evolution of high-density magnetic media as well as bigger, faster, and less expensive solid-state drives. Virtualization is also helping by easing management of aggregated storage pools that are using available capacity more efficiently. Optimization technologies like data reduction, thin provisioning, and automatic tiering are also providing significant benefits.

What has become increasingly apparent in production environments is that the storage decision is no longer relegated to a back office bureaucracy which manages an organizations storage needs en masse. Rather, as organizations increasingly decentralize and decouple decision-making regarding many applications either in the development or maintenance cycle, storage decisions are being made on a more ad hoc basis. Also, as more organizations consider or adopt partial or full cloud or virtualization storage strategies, storage decisions have become more application-centric. Since full migrations to the cloud or to virtual storage models are generally infeasible in the short term, except in the case of smaller organizations, more firms are increasingly managing a tiered storage model. Recent published research by International Data Corporation (IDC) and others demonstrate that storage utilization rates achieved by most U.S. companies are typically 40lower. Hence there appears to be considerable room for improvement in both storage management and implementation practices across the industry.

The above noted disruptive forces in the storage industry are creating a unique set of opportunities and challenges. New firms are appearing that offer storage solutions and services bundled in ways which were virtually unheard of five to seven years ago. At the same time organizations and IT managements under increasing pressure to reduce storage costs, increase storage utilization rates and provide both a secure and transparent accessibility to their data users regardless of the ultimate location of a particular data set. Our project is focussed on providing methods and techniques to allow the computer science professionalto better understand and navigate this new and complex storage landscape and to discover the choices and tradeoffs that one must consider in designing and implementing a sensible organization-wide storage solution.

2 Introduction

Distributed storage systems have evolved from providing a means to store data remotely, to offering innovative services like publishing, federation, anonymity and archival. To make this possible networks have also evolved, generally as a leading indicator of storage evolution. With network infrastructure undergoing another quantum leap recently, and outpacing the bandwidth capability of processors and hard-drives, this provides a platform for future distributed storage systems to offer more services yet again.

The emergence of Cloud Computing, one of the new variables in the storage arena, requires organizations to move from server-attached storage to distributed storage. Along with variant advantages, the distributed storage also poses new challenges in creating both a secure and reliable data storage and access facility over insecure or unreliable service providers. The security of data stored in the cloud is one of the challenges to be addressed before the pay-as-you-go Storage as a Service (STaaS) model can become widespread in the industry. In the enterprise, STaaS vendors are now targeting secondary storage applications by promoting STaaS as a convenient way to manage backups. The key advantage to STaaS in the enterprise is in cost savings in personnel, in hardware and in physical storage space. For instance, instead of maintaining a large tape library and arranging to vault (store) tapes offsite, a network administrator that used STaaS for backups could specify what data was to be relegated to outsourced storage and storage service providers would manage the data from that point. If the company's data ever became corrupt or was lost, the network administrator could contact the STaaS provider and request a copy of the data. For these reasons, STaaS is generally seen as a good alternative for a small or mid-sized business that lacks the capital budget and/or technical personnel to implement and maintain their own storage infrastructure. STaaS is also being promoted as a way for all businesses to mitigate risks in disaster recovery, provide long-term retention for records and enhance both business continuity and availability. Obviously, for larger businesses, the movement to a STaaS model or a partial such model is a longer term proposition as storage equipment reaches the end of its useful life and storage agreements expire. The existence of long term contracts and sizable amortization schedules often make movement to more flexible business storage strategies less likely in the short term. Hence many large organizations are maintaining tiered stage models; at least in the short term planning horizon.

Other trends are equally disruptive and compelling. Full provisioning, the practice of provisioning all the capacity of an external disk to a given app has been accepted practice in industry circles for some time. This practice ensures that a given application has sufficient storage to meet projected growth potential. However, though it typically results in poor utilization rates as noted above. So essentially, a surplus of storage capacity is being acquired, which generally translates into more space and cooling costs in addition to higher overhead costs since unused capacity still needs to be monitored and managed. When applications reach capacity limits and re-provisioning is necessary, the costs increase even more. In adding additional capacity, complex management tasks can be involved. Finally, when an app is taken offline to re-provision capacity, it is then unable to serve business needs and can lead to revenue loss. Thin provisioning has been offered as a solution to address many of the issues noted above. By automatically allocating system capacity to applications as needed, this technology can result in storage utilizations levels of up to 90simultaneously reducing power consumption. This technique allows users to allocate a large amount of virtual capacity for an application, regardless of the physical capacity actually available. This on-demand method not only optimizes storage utilization, but also greatly simplifies capacity planning and management. In order to help users easily monitor capacity utilization, storage systems automatically notify when the capacity utilization is reaching some pre-defined limit. A decision to expand capacity can be done seamlessly.

Under traditional provisioning practices, it is difficult to move data across logical partitions in a storage architecture. When thin provisioning is used, storage capacity from differentlogical partitions can be combined, enabling storage to be dynamically allocated. Conversely, this means that the storage controller can move data dynamically across logical partitions based on how resources are designed to function. Also, thin provisioning allows other advances in storage design, including automated storage tiering. Storage tiering involves grouping data into different categories and assigning these categories to different types of storage media in order to optimize storage utilization. Automated tiering ensures applications have access to the performance levels they need so they can be properly paired up. For example, high performance applications can be assigned to high performance storage tiers featuring drives such as SSDs or SAS, while applications requiring less performance can be assigned to lower tiers featuring low performance drives such as SATA. This ensures that storage resources are not squandered and that applications function effectively. Finally, the new technology helps automatically migrate data based on usage patterns. So, if data in higher storage tiers has not been used for an extended period of time, it is demoted to lower storage tiers. Conversely, if data in lower tiers is frequently accessed, it is promoted upward. Such techniques can greatly improve storage efficiency.

Obviously, the RDBMSs of today aren't going away, certainly not in the short term. However, storage requirements for the new generation of applications are dramatically different. The Semantic Web / Web 3.0 is going to be full of semi-structured data on an even larger scale, so it's prudent for applications to take advantage of the upcoming technologies as soon as possible. Future killer applications and appliances will have to connect to the cloud and hence will be written with distributed storage in mind, whether the applications run on the desktop or on the web. Undeniably, the design of distributed storage poses many challenges - scalability, hardware requirements, query model, failure handling, data consistency, durability, reliability, efficiency, etc. The landscape of storage architectures/software we describe in this paper are helping address many of the concerns raised with respect to current shortfalls in distributed storage architecture and methodology. But, clearly, the future of data and storage is a virtyual model. Our project explores many of these issues as well as the available options and tradeoffs inherent in these choices which are currently shaping Distributed Storage Systems. In the next section we examine the current literature on distributed storage.

3 Related Work

The literature on distributed storage is a long one dating back to the late 1980s. Early works typically provide perspective as well as insight into issues related to building toward the DSS of today. More recent works focus on more cutting edge research generally in areas like peer-to-peer and data grid systems. They also focus on some of the more important issues occurring like routing, consistency, security, auto management and federation.

A number of studies over the past decade have designed and evaluated large- scale, peer-to-peer distributed storage systems. Redundancy management strategies for such systems have also been well evaluated in prior works. Among these, several compared replication with erasure codes in the bandwidth -reliability tradeoff space. The analysis of Weatherspoon and Kubiatowicz demonstrated that erasure codes could reduce bandwidth use by an order of magnitude as compared with replication. The authors showed that in high- turnover scenarios erasure codes provide large storage benefits but the bandwidth cost is too high to be practical for a P2P distributed storage system. In low-churn environments, the reduction in bandwidth is fairly small. In moderate-churn environments, there is some benefit, but this is generally negated by the added architectural complexity that erasure codes introduce into the overall architecture. Other types of systems with publish/share features include NFS, Coda, xFS and Ivy [Muthitacharoen et al. 2002]. Unlike pure archival systems where the storage service aims to be persistent, the publish/share category is somewhat volatile as the main objective here is to provide a capability to share or publish files. The volatility of storage is usually dependent on the popularity of the file. True DSSs in the performance category are typically used by applications which require a high level of performance. The large proportion of systems in this category would be classified as Parallel File Systems (PFSs). PFSs typically are found within a computer cluster, satisfying storage requirements of large I/O-intensive parallel applications. Systems which fall into this category include PVFS [Carns et al. 2000], Lustre [Braam 2002] and GPFS [Schmuck and Haskin 2002.

Finally, the custom category has been created for storage systems that have come into vogue in the past few years, and typically possess a unique set of functional requirements generally customized for a particular environment. Systems in this category may fit into a combination of the above system categories and exhibit unique behavior. Google File System (GFS) [Ghemawat et al. 2003] and OceanStore [Kubiatowicz et al. 2000; Rhea et al. 2003], are such systems. GFS was designed and built with a particular functional purpose which is reflected in that design. Similarly, OceanStore presents itself as a global storage utility, providing many interfaces including a general purpose file system. To ensure scalability and robustness in the event a failure occurs, OceanStore employs P2P mechanisms to distribute and archive data. Likewise, Freeloader [Vazhkudai et al. 2005] combines storage scavenging and striping, achieving good parallel bandwidth on shared resources. The collection of features available from Freeloader, OceanStore and GFS all exhibit unique qualities for a specific set of functional and technical requirements.

In the subsections below we summarize the recent literature with respect to some individual storage system architectures. The architecture is important as it determines the applications operational boundaries, ultimately driving both behavior and functionality. As well as functional qualities, the architecture also has an impact on the means a system may use to achieve consistency, routing and security. One can from the systems discussed below that the evolution of architectures adopted by DSSs have gradually moved away from centralized to more decentralized approaches, primarily as a result of the need for scalability as well as the challenges encountered in operating across a dynamic global network.

3.1 Tahoe File System

The Tahoe-LAFS (Tahoe Least-Authority Filesystem) is a secure, distributed filesystem designed by Zooko Wolcox-O'Hearn and Brain Warner of allmydata.com. Tahoe-LAFS (henceforth referred to as 'Tahoe') is designed around the Principle of Least Authority, and aims to allow one to deploy a trusted distributed filesystem using untrusted, or even actively malicious, nodes. Tahoe is also designed to be fault-tolerant, and to run on commodity hardware where failures may occur frequently. Tahoe uses Reed-Solomon erasure coding to obtain redundancy across nodes. It uses convergent encryption in conjunction with a capability-based access control model, to obtain and maintain security [34].

Tahoe presents a web-API to users of the system that can be used to administrate the system, as well as to read, write, or verify files and directories on the system. This web-based API makes Tahoe suitable for use as the backend for various services and applications in domains such as backup and cloud storage. Tahoe is still actively maintained and is Free Software under the GPL and TGPL. It served as the backend for allmydata.com until allmydata.com went out of business. It is currently deployed for personal use be a number of individuals, and serves as the storage backend for several other systems [23].

3.2 Ceph

Ceph is an object-based parallel file system with a scalable metadata implementation with data replication to allow for data recovery and recovery from hardware failures. Ceph is similar to that of many other distributed file systems in it has metadata servers that a client uses to determine where the file objects live. The key designs for Ceph are providing the ability to scale easily which make it great for use in cloud storage or in cases where hardware for the system may be added or removed. I see the upside of Ceph over Tahoe would be that ceph provides a FUSE based client for use in mounting a Ceph storage system instead of needing to access via a web interface. Additionally it is possible that encryption could be applied by the client prior to storing their files to add some level of security to Ceph that only uses NFS level security making use of uid or gid. [33, 31, 29].

References

[1] K Mani Chandy and Leslie Lamport. Distributed snapshots: Determining global states of distributed systems.

[2] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable. *ACM Trans. Comput. Syst.*, 26(2):1–26, Jun 2008.

[3] H.E Chihoub, G Antoniu, and M S Perez. Towards a scalable, fault-tolerant, self-adaptive storage for the clouds. 2011.

[4] U Divakarla.... An overview of cloud computing in distributed systems. *American Institute of Physics ...*, Jan 2010.

[5] F. Chang et al. Bigtable: A distributed storage system for structured data. *OSDI*, 2006.

[6] Jun Feng, Yu Chen, and Pu Liu. Bridging the missing link of cloud data storage security in aws. pages 1–2, Oct 2009.

[7] A Fox. Above the clouds: A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, 28, 2009.

[8] S. GHEMAWAT, H. GOBIOFF, and S.-T. LEUNG. The google file system. *Proceedings of the 19th Symposium on Operating System Principles*, 2003.

[9] S Ghemawat, H Gobioff, and S.T Leung. The google file system. *ACM SIGOPS Operating Systems Review*, 37(5):29–43, 2003.

[10] Sanjah Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system.

[11] Y Han. Cloud computing: Case studies and total cost of ownership. *Information Technology and Libraries*, Jan 2011.

[12] J.G Hansen and E Jul. Lithium: virtual machine storage for the cloud. *Proceedings of the 1st ACM symposium on Cloud computing*, pages 15–26, 2010.

[13] Qinlu He, Zhanhuai Li, and Xiao Zhang. Study on cloud storage system based on distributed storage systems. pages 1332–1335, Dec 2010.

[14] R Hegarty, M Merabti, Q Shi, and B Askwith. Forensic analysis of distributed data in a service oriented computing platform. *PG Net The 10th Annual Postgraduate Symposium on The Convergence of Telecommunications, Networking & Broadcasting, Liverpool John Moores University*, 2009.

[15] CW Hsu, CW Wang, and S Shieh. Reliability and security of large scale data storage in cloud computing.

[16] Cao Kang. A distributed storage schema for cloud computing based raster gis systems. pages 1–19, May 2011.

[17] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM.*

[18] Ke Liu, Jingli Zhou, Leihua Qin, and Ning Lv. A novel computing-enhanced cloud storage model supporting combined service aware. pages 275–280, Jul 2010.

[19] C Maltzahn, E Molina-Estolano, A Khurana, A.J Nelson, S.A Brandt, and S Weil. Ceph as a scalable alternative to the hadoop distributed file system. *USENIX; login*, 35(4):38–49, 2010.

[20] A Marinos and G Briscoe. Community cloud computing. *Cloud Computing*, pages 472–484, 2009.

[21] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. pages 44–51, Aug 2009.

[22] Stephen Smaldone, Chetan Tonde, Liviu Iftode, Vancheswaran K Ananthanarayanan, and Ahmed Elgammal. The cyber-physical bike: A step towards safer green transportation.

[23] "Tahoe-LAFS Team". "tahoe-lafs website". `http://www.tahoe-lafs.org`, 2012.

[24] B Trushkowsky, P Bodk, A Fox, M.J Franklin, M.I Jordan, and D.A Patterson. The scads director: Scaling a distributed storage system under stringent performance requirements. *USENIX Conf on File and Storage Technologies*, pages 163–176, 2011.

[25] C Wang, Q Wang, K Ren, and W Lou. Ensuring data storage security in cloud computing. *Quality of Service, 2009. IWQoS. 17th International Workshop on*, pages 1–9, 2009.

[26] C Wang, Q Wang, K Ren, and W Lou. Towards secure and dependable storage services in cloud computing. *Services Computing, IEEE Transactions on*, (99):1–1, 2011.

[27] L Wang, J Tao, M Kunze, D Rattu, and A.C Castellanos. The cumulus project: Build a scientific cloud for a data center. *Cloud Computing and its Applications*, 2008.

[28] S.A Weil. Ceph: Reliable, scalable, and high-performance distributed storage. 2007.

[29] S.A Weil, S.A Brandt, E.L Miller, D.D.E Long, and C Maltzahn. Ceph: A scalable, high-performance distributed file system. *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 307–320, 2006.

[30] Sage A Weil. Ceph: Reliable, scalable, and high-performance distributed storage. *Dissertation.*

[31] Sage A Weil, Scott A Brandt, Ethan L Miller, and Darrell D E Long. Ceph: A scalable, high-performance distributed file system.

[32] Sage A Weil, Scott A Brandt, Alex J Nelson, Amandeep Khurana, Esteban Molina-Estolano, and Carlos Maltzahn. Ceph as a scalable alternative to the hadoop distributed file system.

[33] Sage A Weil, Feng Wang, Quin Xin, Scott A Brandt, Ethan L Miller, Darrell D E Long, and Carlos Maltzahn. Ceph: A scalable object-based storage system.

[34] Z Wilcox-O'Hearn and B Warner. Tahoe: the least-authority filesystem. *Proceedings of the 4th ACM international workshop on Storage security and survivability*, pages 21–26, 2008.

[35] Zooko Wilcox-OHearn and Brian Warner. Tahoe: the least-authority filesystem.

[36] Thomas B Winans and John Seely Brown. Cloud computing a collection of working papers. pages 1–33, Jul 2009.

[37] H Yoon, M Ravichandran, A Gavrilovska, and K Schwan. Distributed cloud storage services with flecs containers.