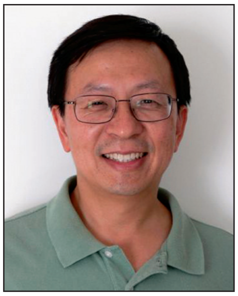

The Growing Pains of Cloud Storage

Yih-Farn Robin Chen • *AT&T Labs Research*

Cloud storage is growing at a phenomenal rate, fueled by multiple forces, including mobile devices, social networks, and big data. Content is created anytime and anywhere on billions of smartphones and tablets; high-resolution photos and videos are frequently uploaded to the cloud automatically as soon as they're captured. A Gartner report predicts that consumer digital storage will grow to 4.1 zettabytes in 2016, with 36 percent of this storage in the cloud.[1] Social interactions and transactions on the Internet are frequently captured and analyzed for targeted advertising. In addition to social networks and e-commerce, big data analytics are growing in many other sectors, including government, healthcare, media, and education. An IDC forecast suggests that big data storage is growing at a compound annual growth rate of 53 percent from 2011 to 2016.[2]

The growth in cloud storage has made it an expensive cost component for many cloud services and today's cloud infrastructure. Whereas raw storage is cheap, the performance, availability, and data durability requirements of cloud storage frequently dictate sophisticated, multitier, geo-distributed solutions. Amazon Simple Storage Service (S3) offers 11 nines of data durability (99.999999999 percent), but some other services demand even more stringent requirements due to the sheer number of objects being stored in the cloud (1.3 billion Facebook users, uploading 350 million photos each day) and to the data's importance (who can afford to lose a video of their baby's first steps?). Data is frequently replicated or mirrored in multiple datacenters to avoid catastrophic loss, but copying it across datacenters is expensive. The networking cost is frequently proportional to the distance and bandwidth requirements between datacenter sites.

Traditional storage systems use dedicated hardware and networking to guarantee preservation of the quality-of-service (QoS) requirements, such as throughput, latency, and IOPS (total number of input/output operations per second). Unfortunately, these dedicated resources are frequently underutilized. Cloud computing promises efficient resource utilization by allowing multiple tenants to share the underlying networking, computing, and storage infrastructure. However, providing end-to-end storage QoS guarantees to individual tenants is difficult without mechanisms for avoiding interference. Typically, in a cloud environment such as Openstack, multiple tenants share the backend block storage (Linux's logical volume manager or a Ceph RADOS block device [RBD], for example) through a storage virtualization layer such as Cinder, which attaches virtual machines (VMs) to individual storage volumes. Providing customized storage QoS to meet different tenant needs is challenging. One exception is all-SSD storage arrays; some vendors (such as Solid Fire) let different tenants allocate storage volumes with different QoS types and dynamically change them, but all-SSD solutions (on the order of US$1,000 per terabyte) are expensive compared to HDD-based solutions. Moreover, an IOPS guarantee in the backend isn't sufficient because there might be contention for network bandwidth or CPU capacity from other tenants.

Finally, to operate any Web-scale solutions, infrastructure service providers are moving to scale-out solutions based on commodity hardware, instead of expensive storage appliances, which are frequently more expensive and difficult to adapt to changing workload or specific QoS requirements. Any cloud solution architect must understand the tradeoffs among the performance, reliability, and costs of cloud storage to provide an effective overall solution.

Emerging trends are sweeping through the storage industry to address these issues. Here,

1089-7801/15/$31.00 © 2015 IEEE

I discuss two software-based solutions: erasure-coded storage and software-defined storage (SDS).

Erasure-Coded Storage

Erasure coding has been widely studied for distributed storage systems. Various vendors, companies, and open source software systems have adopted it recently, including EMC, Cleversafe, and Amplidata; Facebook, Microsoft, and Google; and Ceph, Quantcast File System (QFS), and a module of the Hadoop Distributed File System (HDFS-RAID), respectively. The primary reason for this adoption is that erasure-coded storage uses less space than fully replicated storage, while providing similar or higher data durability.

To understand why erasure coding is becoming crucial in storage systems, I must explain some basics. Erasure coding is typically controlled by two key parameters: k and n. A file or file segment is typically broken into k chunks, erasure coded, and expanded into n chunks ($n > k$) that are distributed over n storage servers or hard disks. Any k chunks are sufficient to reconstruct the original file, which can tolerate up to a loss of $m = n - k$ chunks without any data loss. One way to think about erasure coding is to consider a system of over-specified linear equations. You're essentially given n linear equations to solve for k variables. Picking any k out of these n equations would be sufficient to determine the values of those k variables. We frequently refer to the first k chunks as *primary chunks*, and the m chunks as *parity chunks*. Because we can vary k and m arbitrarily, a general erasure-coded storage solution in the form of (k, n) or $k + m$ has much higher flexibility in terms of the tradeoffs between storage space and reliability compared to the popular RAID 6 system, which uses only two parity blocks and is equivalent to a $k + 2$ erasure-coded scheme.

A scalable distributed storage system, such as HDFS or Swift, stored on multiple racks or sites typically uses triple redundancy (three copies of each data block) to improve both availability and durability. As the cloud storage volume continues to grow exponentially, the triple redundancy scheme becomes expensive. As an example, the QFS system uses 6 + 3 ($k = 6$ and $m = 3$) erasure coding and is designed to replace HDFS for MapReduce processing. HDFS uses triple replication and incurs 200 percent storage overhead, but it can only tolerate up to ANY two missing blocks of the same data. A 6 + 3 erasure code, on the other hand, can tolerate up to ANY three missing coded blocks with only 50 percent storage overhead. Such significant cost savings, while maintaining the same or higher reliability, is why many storage systems are now incorporating erasure codes.

One concern with erasure-coded storage is the extra overhead caused

Technical cosponsor:
IEEE COMMUNICATIONS SOCIETY

IEEE Internet Computing
IEEE Computer Society Publications Office
10662 Los Vaqueros Circle
Los Alamitos, CA 90720 USA

Editorial. Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Internet Computing* does not necessarily constitute endorsement by IEEE or the IEEE Computer Society. All submissions are subject to editing for style, clarity, and length.
Submissions. For detailed instructions, see the author guidelines (www.computer.org/internet/author.htm) or log onto *IEEE Internet Computing*'s author center at ScholarOne (https://mc.manuscriptcentral.com/cs-ieee). Articles are peer reviewed for technical merit.
Letters to the Editors. Email lead editor Brian Brannon, bbrannon@computer.org
On the Web. www.computer.org/internet/.
Subscribe. Visit www.computer.org/subscribe/.
Subscription Change of Address. Send requests to address.change@ieee.org.
Missing or Damaged Copies. Contact help@computer.org.
To Order Article Reprints. Email internet@computer.org or fax +1 714 821 4010.
IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

by the encoding/decoding time, which depends heavily on the erasure-coding scheme's strength. For a fixed k, higher m or n incurs more computation overhead while providing higher reliability. As computing servers gain in performance, the computation overhead of commonly used erasure codes becomes more manageable, and the bottleneck is frequently shifted to the disk or network throughput.

Another concern is the repair cost. Given that erasure coding of 6 + 3 requires six chunks to repair one chunk, the networking cost of repairing a chunk is six times that of a simple replication scheme. Some Facebook experiments use a 10 + 4 erasure-coding scheme, which incurs even higher repair costs (but lower storage overhead at 40 percent). Several repair schemes (such as Xorbas[3] and Hitchhiker[4]) have been proposed to reduce the repair bandwidth, with or without additional storage overhead.

As data durability becomes increasingly important for cloud storage, erasure coding can also play an important role in cloud storage geo-distribution. It allows chunks of an erasure-coded file to be placed in multiple datacenters or racks to increase data durability. For example, a 9 + 15 or (9, 24) erasure-coded storage system could put six chunks each in New Jersey, Illinois, Texas, and California (east, north, south, and west areas of the US). Because any file can be reconstructed from nine chunks, it can be reconstructed by retrieving those chunks from any two datacenters. Thus, it will tolerate up to two datacenter failures, even during a major natural disaster such as 2013's Hurricane Sandy, which caused a loss of 68 billion dollars and affected 24 states. On the other hand, because each file retrieval requires accessing chunks from two datacenters, it might incur longer latency and significant communication costs, which is fine for archival storage, but not ideal for frequently accessed storage. Alternatively, if we know certain files' access patterns, and it turns out that most accesses come from New Jersey, we can place nine chunks in New Jersey and five chunks each in Illinois, Texas, and California. This would allow users to complete most accesses with low network latency and slightly lower reliability, given that a datacenter loss has the potential to lose nine instead of six chunks. The chunk-placement issue in erasure coding affects latency, cost, and reliability in geo-distributed storage systems and is currently an active research field.

To realize the vision of a virtualized datacenter, we need software-defined storage that virtualizes storage resources and separates storage management software from the underlying hardware.

Software-Defined Storage

Cloud computing started with the virtualization of computing resources, followed by recent advances and rapid innovations in software-defined networks (SDNs), which aim to virtualize networking resources and separate the control plane from the data plane. To truly realize and complete the vision of a virtualized datacenter, however, we need software-defined storage that virtualizes storage resources as well and separates storage management software from the underlying hardware.

Unfortunately, unlike SDN, there isn't a clear definition of what software-defined storage really is, although many storage vendors claim that they have SDS solutions. Most SDS definitions include a list of desirable attributes.[5,6] Here, I summarize those that pertain to multitenant cloud storage solutions, what I call the S.C.A.M.P. principles of SDS.

Scale-Out

SDS should enable a scale-out (horizontal scaling of low-cost, commodity hardware) instead of a scale-up (vertical scaling using more powerful hardware) storage solution as the workload grows or changes dynamically over time. A scale-out solution is best implemented in a cloud environment with large computing, networking, and storage resource pools. A cloud storage solution is never just about storage — all the necessary computing and networking resources must also scale accordingly to support common storage operations: deduplication, compression, encryption/decryption, erasure coding/replication, and so on.

Customizable

SDS should allow storage system customization to meet specific storage QoS requirements. This lets customers purchase storage solutions based on their specific performance and reliability constraints and avoid unnecessary over-engineering, which frequently happens when a cloud storage service provider tries to meet the needs of multiple customers with diverse requirements. In a multitenant cloud with a shared backend storage, guaranteeing the desired storage QoS is particularly difficult. The latest version of Openstack Cinder, which provides a block storage service, now allows multiple backends with different QoS types (such as different IOPS or throughput numbers) to partially address this issue.

Automation

Once storage QoS requirements are clearly defined, SDS should automate the complete provisioning and deployment process without human intervention. The current practice is that a storage architect or system administrator is intimately involved in designing and installing the storage system. This process is typically error-prone and not amenable to adapting to changing workloads or requirements in real time.

Masking

SDS could mask the underlying storage system (physical or virtualized) and distributed system complexity (single or multiple-site) as long as such systems can present a common storage API (block, file system, object, and so on) and meet QoS requirements. This gives infrastructure service providers greater flexibility in restructuring their resource pools or architecting storage systems. For example, Ceph can present a block device API even though the underlying implementation is done in its RADOS object storage.

Policy Management

SDS software must monitor and manage the storage system according to the specified policy and continue to meet storage QoS requirements despite potential interference from other tenants' workloads. It must also handle failures and autoscale the system when necessary to adapt to changing workloads. As stated previously, however, guaranteeing end-to-end storage QoS in a multitenant cloud is a hard problem that requires protecting resources on the entire path from a VM to the storage volume. Microsoft's IOFlow[7] aims to provide an SDN-like controller to control storage bandwidth allocation at multiple points of such a path.

SDS Definition

By combining the S.C.A.M.P. principles, we can now define SDS: an SDS solution should automatically map customizable storage service requirements to a scalable and policy-managed cloud storage service, with abstractions that mask the underlying storage hardware and distributed system complexities.

Incidentally, erasure coding is a crucial technology that can help meet the SDS customization requirement. For a fixed k, varying n (or m, the number of parity chunks) increases the reliability and replication factor (and hence the storage cost). At the same time, it increases the overall encoding/decoding time, hence the required computation capacity, and perhaps reduced performance. This lets an automated storage architect look at the storage QoS requirements and pick particular erasure-code parameters (k and m) to meet the minimal reliability and performance requirements with the least amount of storage overhead.

The rapid growth of cloud storage has created challenges for storage architects to meet different customers' diverse performance and reliability requirements while controlling costs in a multitenant cloud environment. Erasure-coded storage and SDS could address these challenges and open up new opportunities for innovation. Moreover, erasure coding could play a crucial role in offering design tradeoffs in certain SDS solutions. These two technologies, working together, have a huge potential to address the growing pains of cloud storage and help ease the transition from traditional IT storage solutions — given that cloud storage will likely support a large portion of all IT storage needs in the future.

References

1. "Gartner Says that Consumers Will Store More than a Third of Their Digital Content in the Cloud by 2016," Gartner, press release, 25 June 2012; www.gartner.com/newsroom/id/2060215.
2. "Big Data Drives Big Demand for Storage, IDC Says," *Business Wire*, 16 Apr. 2013; www.businesswire.com/news/home/20130416005045/en/Big-Data-Drives-Big-Demand-Storage-IDC.
3. M. Sathiamoorthy et al., "XORing Elephants: Novel Erasure Codes for Big Data," *Proc. VLDB Endowment*, 2013, pp. 325–336.
4. K. Rashmi et al., "A Hitchhiker's Guide to Fast and Efficient Data Reconstruction in Erasure-Coded Data Centers," *Proc. 2014 ACM Conf. SIGCOMM*, 2014, pp. 331–342.
5. B. Earl et al., "Software-Defined Storage," *5th Usenix Workshop on Hot Topics in Storage and File Systems*, panel, 2013; www.usenix.org/conference/hotstorage13/workshop-program/presentation/earl.
6. M. Carlson et al., "Software-Defined Storage," Storage Networking Industry Assoc. working draft, Apr. 2014; http://snia.org/sites/default/files/SNIA%20Software%20Defined%20Storage%20White%20Paper-%20v1.0k-DRAFT.pdf.
7. E. Thereska et al., "IOflow: A Software-Defined Storage Architecture," *Proc. 24th ACM Symp. Operating Systems Principles*, 2013, ACM, pp. 182–196.

Yih-Farn Robin Chen is a Lead Inventive Scientist at AT&T Labs Research. His research interests include cloud computing, storage systems, mobile computing, and distributed systems. Chen received a PhD in computer science from the University of California, Berkeley. He's a vice chair of the International World Wide Web Conferences Steering Committee (IW3C2) and a member of the editorial board of *IEEE Internet Computing*. Contact him at chen@research.att.com.