Reliability and Security of Large Scale Data Storage in Cloud Computing

C.W. Hsu [1] ***C.W. Wang*** [2] ***Shiuhpyng Shieh*** [3]

Department of Computer Science
National Chiao Tung University
Email: [1]vicnobody@gmail.com
[2]cwwangabc@gmail.com

Taiwan Information Security Center at
NCTU (TWISC@NCTU)
[3]ssp@cs.nctu.edu.tw

Abstract

In this article, we will present new challenges to the large scale cloud computing, namely reliability and security. Recent progress in the research area will be briefly reviewed. It has been proved that it is impossible to satisfy consistency, availability, and partitioned-tolerance simultaneously. Tradeoff becomes crucial among these factors to choose a suitable data access model for a distributed storage system. In this article, new security issues will be addressed and potential solutions will be discussed.

Introduction

"*Cloud Computing*", a novel computing model for large-scale application, has attracted much attention from both academic and industry. Since 2006, Google Inc. Chief Executive Eric Schmidt introduced the term into industry, and related research had been carried out in recent years. Cloud computing provides scalable deployment and fine-grained management of services through efficient resource sharing mechanism (most of them are for hardware resources, such as VM). Convinced by the manner called "*pay-as-you-go*" [1][2], companies can lower the cost of equipment management including machine purchasing, human maintenance, data backup, etc.

Many cloud services rapidly appear on the Internet, such as Amazon's EC2, Microsoft's Azure, Google's Gmail, Facebook, and so on. However, the well-known definition of the term "*Cloud Computing*" was first given by Prof. Chellappa in 1997 [3]. He suggested that it would be a new "computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits alone." To this day, "*Cloud Computing*" is not just a computation model allowing us to break the technical restrictions of computation. It also establishes a new kind of business models (e.g. a convenient way to develop applications, a data center for large scale storage) and merges related technologies together into a single term [14][15][16].

According to the definition of NIST [5], essential characteristics of cloud computing include on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. To achieve some of above features, a distributed storage system plays a vital role in cloud computing. Google said "*The Data Center Is the Computer* [6]." As we

know, information must be stored in some sort of recording devices such as memory, hard disk, tape, and cache so that they can be reused or queried in future. Without reliable distributed storage system, cloud computing cannot provide reliable service to users. Consequently, this raises the concern about the reliability of large-scale data storage.

Reliability of distributed data storage, including I/O performance [8][9] and security [10][11][12][13][18], had been studied in decades. Although people had developed much more concrete sense in security and hardware had become more powerful, many problems still lie in the reliability of cloud computing. Due to the new computation model, traditional solutions to achieving high reliability may not be appropriate for modern applications in cloud computing. In this article, we focus on two important features of distributed storage: 1) capability of distributed storage, 2) information security in cloud computing. First of all, we will introduce Brewer's CAP Theorem, a natural tradeoff between consistency and availability in distributed data storage systems. Secondly, we summarize current implementations of some popular NoSQL database [17]. Lastly we briefly discuss some security issues accompanied by cloud computing.

Brewer's CAP Theorem

Traditional database for web services generally provide ACID-guarantees for reliability [19]. A mechanism satisfying ACID principle must include following characteristics:

1. Atomic: Operations commit or fail in their entirety.
2. Consistent: All replicas in system have the same value.
3. Isolated: Uncommitted transactions cannot access data that has been modified during an unfinished transaction.
4. Durable: Data can be recovered to survive any kind of system failure.

By following the ACID manner, storage system can protect committed data in a single node.

In 2000, Eric Brewer pronounced the CAP conjecture [21], it is impossible for a distributed service to provide consistency, availability, and partition-tolerance at the same time. To provide consistency, all copies of data in the system must be observed coherently by external users. As for availability, the storage system must be able to tolerate node failures (e.g. hardware crash) and always responds to every valid request. Partition-tolerance implies that the system must continue to operate even if it suffers from network failures, which partition the system into disconnected sub-networks. It could be realized by a simple idea: since replicas locating on partitioned sub-networks cannot be synchronized at all, it is impossible to provide availability and consistency under such circumstances [20]. However, it is feasible to achieve any two of the three properties.

CAP theorem had become a principle for designs of NoSQL databases. However, such databases are inherently designed to be scaled horizontally so that they can be used in cloud computing. We will introduce following notable implementations of NoSQL database products including Google's BigTable, Amazon's Dynamo and Apache Cassandra. These implementations are categorized by CAP properties.

Consistency + Partitioned-tolerance system

Bigtable [22] is designed to scale to petabytes of data across thousands of commodity servers. Because of the dynamic data layout (from URLs to web pages to satellite imagery) and the efficient latency of large-scale data access (relate to traditional database), many applications of Google are based on it (i.e. web indexing, Google Earth, and Google Finance). The data model of BigTable is a variety of potential uses so that users can rearrange their data model by adding a new column in any ColumnFamily. Moreover, Bigtable can cooperate with MapReduce, a framework for running large-scale parallel computations developed at Google, to enrich the ability of users' database. Unfortunately, Google does not release source code of BigTable for developers. Thus, HBase and Hypertable [23], two open-source implementations, which approximately emulate most components of Google's BigTable, appeared as substituting solutions for BigTable-like systems.

Availability + Partitioned-tolerance system

Dynamo [24], a highly available key-value storage to provide "always on" experience, solves reliability and scalability at Amazon.com. They believe that even the slightest outage has significant financial consequences and impact customer trust. Amazon.com, one of the largest ecommerce operations in the world, sacrifices consistency under certain failure and provide high availability of financial services. To achieve scalability and availability at the same time, Dynamo uses consistent hashing to index data more flexible and object versioning to keep loose consistency. Replicas of data in Dynamo are synchronized eventually when clients access this data for resolving conflicts. The polity of conflict resolution is "last write wins" because of their practical experience. For decentralization, Dynamo adapts a ring structured peer-to-peer system, and every node should have the same set of responsibilities as its peers. Due to its P2P architecture, Dynamo tolerates most kinds of system failure for high availability and allows service owners to scale up and down easily.

Availability + Partitioned-tolerance/Consistency + Partitioned-tolerance system

Cassandra [25][26] is a highly scalable, eventually consistent, distributed, and structured key-value store. It combines the characteristic of eventually consistency from Dynamo and the data model (ColumnFamily-based) from Google's BigTable. Cassandra is an open source data storage system designed by Avinash Lakshman (one of the authors of Amazon's Dynamo)

and Prashant Malik (Facebook Engineer). Depending on the intended usage, Cassandra can be opted for Availability + Partitioned-tolerance or Consistency + Partitioned-tolerance mode. Clients can specify a desired consistency level (ZERO, ONE, QIORUM, ALL, ANY) for each read/write operation. Consistency level ZERO provides the lowest latency but the operation shall be processed asynchronously. Hence, there is no guarantee for data consistency. Consistency level ONE makes sure that the transaction will be written into the commit log at least on one server. QUORUM requires more than half replicas are written onto servers. Cassandra becomes consistent storage when users choose consistency level ALL. With flexible configuration, Cassandra can be adjusted to satisfy user's requirement dynamically and deployed easily as Dynamo due to its P2P architecture. Thus it is suitable for those users who want to take a glance at NoSQL systems.

Information Security in Clouds

Cloud computing is revolutionizing since developers no longer require capital outlays in hardware to deploy their services. However, it comes with security problems [28][30] including traditional and new ones. Both security and privacy are most concerned by users leveraging cloud services. The following is the related work focusing on information security in cloud storage from the users' point of view.

Users can place more trust on cloud-storage providers as long as they can be assured that their data on the cloud cannot be modified, or at least they can be aware of the modification. HAIL [27] is a distributed cryptographic system, which allows a set of services to assure a client that a stored file is intact and retrievable. It is a remote-file integrity checking protocol and ensures file availability against adversaries. Proofs of retrievability prevent users' data from corruption or irretrievability.

Just like Amazon's EC2, people can rapidly setting up their own web services by leveraging virtual-machine images containing users' applications. Integrity of these images must be assured because they present the startup state of VM including running process, network status, memory, virtual hard disk, and so on. Differences in states shall result in unexpected event, or even worse, a compromising one. To avoid this problem, Mirage [29], an efficient image management, provides a secure image access control mechanism to ensure users' privacy and application security. Moreover, it can filter malicious images and remove sensitive information so that users require no expert experience of security. In the way, the risk of being compromised can be decreased and secure application runtime can be guaranteed.

Conclusions

In this article, we addressed the reliability and security issues of cloud computing. Cloud computing brings productive development and elastic resource management. However, there exists the natural restriction of distributed storage system, which had been proved by Eric Brewer and formalized to CAP Theorem. It is impossible to satisfy consistency, availability, and partitioned-tolerance simultaneously. Tradeoff becomes crucial among these factors to choose a suitable data access model for a distributed storage system. Various implementations of NoSQL database have been introduced and their characteristics have been also categorized. Cloud computing increases the possibility for users to get exposed to risky environments. Therefore, information security plays a vital role in cloud services. Due to the new computing model, security issues had been studied by academia and hacker communities, such as Black Hat, in recent years. The field deserves much more attention since users will not place their trust on service providers without high reliability and security assurance.

Acknowledgement

This work is supported in part by National Science Council, NCP, TWISC, ITRI, III, iCAST, Chungshan Institute of Science and Technology, Bureau of Investigation, and Chunghwa Telecomm.

Reference

[1] Amazon elastic compute cloud. http://aws.amazon.com/ec2/

[2] Microsoft's Windows Azure platform. http://www.microsoft.com/windowsazure/

[3] A webpage of Ramnath K. Chellappa. http://www.bus.emory.edu/ram/

[4] Google app engine. http://code.google.com/appengine/.

[5] P. Mell and T. Grance. NIST definition of cloud computing. National Institute of Standards and Technology. October 7, 2009.

[6] Urs Hoelzle , Luiz Andre Barroso, "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines," Morgan and Claypool Publishers, 2009.

[7] G. Chockler, et al., "Reliable Distributed Storage," IEEE Computer, pp. 60-67, April 2009.

[8] Y. Gu and R.L. Grossman, "Sector and Sphere: the design and implementation of a high-performance data cloud," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 367, 2009, pp. 2429 -2445.

[9] A.W. Leung, M. Shao, T. Bisson, S. Pasupathy, and E.L. Miller, "Spyglass: Fast, scalable metadata search for large-scale storage systems," Proccedings of the 7th conference on File and storage technologies, 2009, pp. 153–166.

[10] Y. Chen, V. Paxson, and R. Katz, "What's New About Cloud Computing Security," University of California, Berkeley Report No. UCB/EECS-2010-5 January, vol. 20, 2010, pp. 2010–5.

[11] L.M. Kaufman, “Data security in the world of cloud computing,” Security & Privacy, IEEE, vol. 7, 2009, pp. 61–64.

[12] Y.Y. Yumin, “Application of Cloud Computing on Network Security,” Science, 2009, p. 07.

[13] M. Jensen, J. Schwenk, N. Gruschka, and L.L. Iacono, “On technical security issues in cloud computing,” Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, 2009, pp. 109–116.

[14] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and others, “A view of cloud computing,” Communications of the ACM, vol. 53, 2010, pp. 50–58.

[15] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” Grid Computing Environments Workshop, 2008. GCE'08, 2009, pp. 1–10.

[16] L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, A break in the clouds: towards a cloud definition,” ACM SIGCOMM Computer Communication Review, vol. 39, 2008, pp. 50–55.

[17] NoSQL systems. http://nosql-database.org/

[18] L.M. Kaufman, “Data security in the world of cloud computing,” Security & Privacy, IEEE, vol. 7, 2009, pp. 61–64.

[19] T. Haerder and A. Reuter, “Principles of transaction-oriented database recovery,” ACM Computing Surveys (CSUR), vol. 15, 1983, pp. 287–317.

[20] S. Gilbert and N. Lynch, “Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services,” ACM SIGACT News, vol. 33, 2002, p. 59.

[21] E.A. Brewer, “Towards robust distributed systems,” Proceedings of the Annual ACM Symposium on Principles of Distributed Computing, 2000, pp. 7–10.

[22] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R.E. Gruber, “Bigtable: A distributed storage system for structured data,” ACM Transactions on Computer Systems (TOCS), vol. 26, 2008, pp. 1–26.

[23] A. Khetrapal and V. Ganesh, “HBase and Hypertable for large scale distributed storage systems,” Dept. of Computer Science, Purdue University.

[24] D. Hastorun, M. Jampani, G. Kakulapati, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: amazon’s highly available key-value store,” In Proc. SOSP, 2007.

[25] A. Lakshman and P. Malik, “Cassandra: a decentralized structured storage system,” ACM SIGOPS Operating Systems Review, vol. 44, 2010, pp. 35–40.

[26] D. Featherston, “Cassandra: Principles and Application.”

[27] K.D. Bowers, A. Juels, and A. Oprea, “HAIL: A high-availability and integrity layer for cloud storage,” Proceedings of the 16th ACM conference on Computer and

communications security, 2009, pp. 187–198.

[28] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds,” Proceedings of the 16th ACM conference on Computer and communications security, 2009, pp. 199–212.

[29] J. Wei, X. Zhang, G. Ammons, V. Bala, and P. Ning, “Managing security of virtual machine images in a cloud environment,” Proceedings of the 2009 ACM workshop on Cloud computing security, 2009, pp. 91–96.

[30] H. Meer, N Arvanitis, and M. Slaviero. Clobbering the cloud. Black Hat USA 2009.