Harwinder Singh, Rajeev Bedi / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.1925-1929

Failures in Cloud Computing

Harwinder Singh*, Rajeev Bedi**
*(Department of Computer Science & Engineering
Beant College of Engg & Tech Gurdaspur)
**(Department of Computer Science & Engineeirng,
Beant College of Engg & Tech Gurdaspur)

Abstract— Cloud computing promises a more cost effective enabling technology to outsource storage and computations. Existing approaches for secure outsourcing of data and arbitrary computations are either based on a single tamper-proof hardware, or based on recently proposed fully homomorphic encryption. The hardware based solutions are not scaleable, and fully homomorphic encryption is currently only of theoretical interest and very inefficient.

In this paper we propose an architecture for secure outsourcing of data and arbitrary computations to an untrusted commodity cloud. In our approach, the user communicates with a trusted cloud (either a private cloud or built from multiple secure hardware modules) which encrypts and verifies the data stored and operations performed in the untrusted commodity cloud. We split the computations such that the trusted cloud is mostly used for security-critical operations in the less time-critical setup phase, whereas queries to the outsourced data are processed in parallel by the fast commodity cloud on encrypted data.

Keywords- **Cloud Computing failure; complexity of system**

I. INTRODUCTION

Cloud Computing has revolutionized the field of IT with its approach to provide service infrastructures. Cloud Computing is a technology that with the help of internet and central remote servers maintains data and applications and allows consumers and businesses to use applications without installation and access their personal files at any place. Cloud providers deal their services on cloud resources for money. These data and applications are made available to the users for a specific price that they need to pay to the cloud providers. Cloud provides facilities varying from platforms to infrastructure and applications deployed on the infrastructure.

Cloud providers impose a price on the cloud services, and, thus, economic concepts are included in the term of cloud services. The purpose behind the pricing on cloud services is to fulfill two criteria: (i) user satisfaction and (ii) cloud profit maximization. The concept of pricing not only allows users to be contended with its services but also gives an advantage to its business; eventually the main concern is cloud profit. Thus the pricing scheme is such that the user satisfaction is guaranteed and maximum cloud profit is obtained with price adapting to time change and user demand.

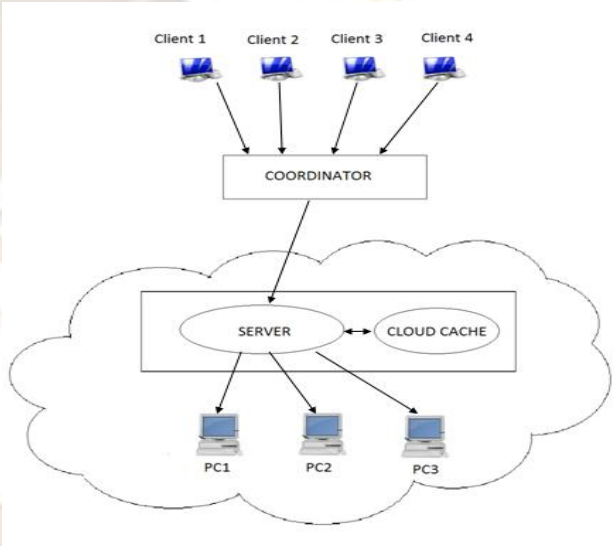

Fig.1. Cloud using cache

As seen in fig.1 clients pose queries to the cloud through a coordinator module. The cloud caches data and builds data structure in order to accelerate query execution. Price over the cloud cache ensures profit for the cloud. The price keeps on changing dynamically with the user demands; resulting into profit for the user and cloud service providers.

Nowadays clouds also focus on the provision of web services targeted to developers, such as Amazon Elastic Compute Cloud (EC2) [1], or the deployment of servers, such as GoGrid [2]. Emerging clouds such as the Amazon SimpleDB and Simple Storage Service offer data management services. Beneficial pricing of cached structures is central to maximizing profit for a cloud that offers data services.

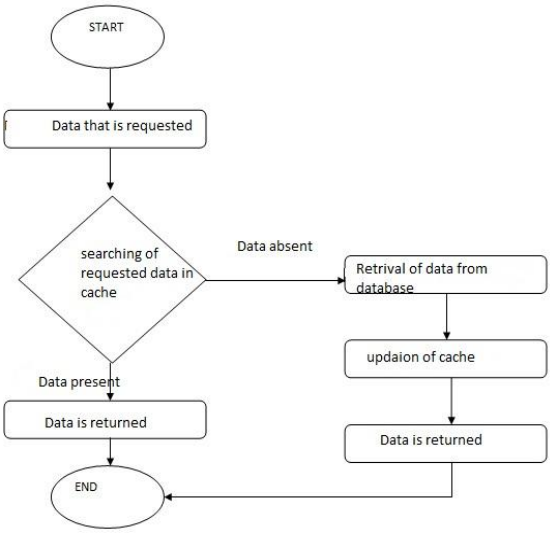

Fig.2. Flow chart

In Fig.2 the client requests the data that is to be fetched through a co-ordinate module. The cloud searches the requested data in the cache. If the data is found in the cache, it is returned to the client. And if the data is not present in the cache, the requested data is fetched from the database. The cache is updated with the requested data and the data is return to the client.

Nowadays, Cloud service failures have become a major concern and take place even though cloud computing has proved to be a boon and shown its importance. It can be said that cloud computing has a great future with big chances of failure. A big reason why major disasters occur is due to the failure in recovering the data on a large scale, resulting into major data and monetary loss by the organizations. There are many unknown real-production scenarios in which a failure recovery might not work.

"The best way to avoid failure is to fail constantly."
"Learn with real scale, not toy models."
– Netflix Engineers [3]

Cloud computing has matured. Nowadays more and more local computing applications are replaced by easy to- use on-demand services accessible through network i.e. cloud services. These services have a massive hardware infrastructures running behind them that monitor complex tasks like software upgrades. They can exhibit failures which, if not handled correctly, can lead to severe implications. In past outages, failures were often cascaded to other healthy clusters, dependent services went down, manual mistake-prone recovery code had to be quickly written during the outage, and users were frustrated and furious [4,5,6,7,8,9,10,11].

In many of the failures that have occurred the service provider had already expected recovery of that work as they had anticipated those failure scenarios. These incorrect expectations could arise because cloud service deployment is complex and many scenarios might have not been tested [11]. Popularity of a service results in handling and dealing with more requests, data, and failures. In other words, the scale of an actual deployment is typically orders of magnitude larger than the scale of a testing framework or a "toy model" [11].

The Netflix engineers' quote above sums up our motivation. Many real scenarios cannot be covered in offline testing, and thus a new paradigm has emerged: failures should be deliberately injected in actual deployments [11, 12, and 13].

The main principle of a cloud is to expect and plan for any failure that may occur in the system. Cloud providers architect their cloud in such a way that in any environment they can recover automatically.

Even if Cloud providers architect internal infrastructure to be elastic and fault tolerant that doesn't mean that the application can automatically handle failures and remove them from cloud.

II. REASONS OF CLOUD FAILURE

In the recent times it has become clear that the enterprises are opting for cloud. Earlier what we used to see as workshops or pilot projects are now seen as production implementation projects on cloud.
Opting for cloud is a good sign; but along comes the risks as the failure of some cloud projects has been witnessed. So far we have heard of only good things about Cloud, but now it is seen that the clouds are not living up to the expectations. Due to which there is uncertainty and doubt in the minds of enterprises. Enterprises should do a careful analysis of any perceived failures of Cloud projects, rather than simply blaming it on the Cloud as a concept [22].

The cloud providers should be aware of all the risks that are involved in providing services to the enterprises. Few of the points mentioned below summarize the common mistakes that take place during migration of on-premise platforms and applications to cloud [22].

- *Wrong motivation for selecting a cloud:* The stakeholders are not much into technical knowledge and are not fully educated about the benefits of the cloud. Cloud providers should take it as their responsibility and make plans for the enterprise strategies, in which each stakeholder should be clearly instructed about the maximum benefits that the cloud will have and where does it fits the most into the overall enterprise IT strategy[22]. The cloud providers might not be able to meet the expectations and doom

as a failure if they take the job of providing services to uninformed stakeholders just to make some quick money without planning for a long term strategy.

- *Use of Unskilled Resources for the cloud*: Cloud is a technical area and needs skilled resources with technical capabilities for integrating on-premise and cloud applications. There are times when there may not be people who are skilled at cloud platforms and their implementation. Thus choosing wrong people for this complex technology have huge chances of failure.

- *Non Involvement of IT in business:* It is seen that business has an upper hand in adopting cloud than IT. Business has raced ahead of IT in terms of adopting clouds due to its ability to meet their time-to-market demands and not waiting for the IT team to deliver best results [22]. Though the results are good but in few cases few of the non functional aspects of a cloud solution are neglected due to non involvement of IT. Cause of this the difference between actual implementations and marketing claims are ignored by the business.

- *Selection of wrong Deployment Model:* Few popular deployment models for cloud solutions [22]:
  - *Full cloud implementation*, where the entire solution runs on the cloud
  - *Full on-premise implementation* where most cloud solutions have packaged into a 'Cloud in a Box' concept in the form of an appliance, which can fully run on-premise on a private cloud
  - *Hybrid implementation* which is about keeping part of the solution on cloud and part On Premise.

Considering the above models it is not easy to tell which option is good for different applications in cloud migration and which model suits the most.

Sometimes the failure of a cloud migration is not really about the product or the concept but rather about not choosing a correct deployment model [22].

II. CLOUD AS COMPLEX SYSTEM

A complex system, according to Wikipedia is a system composed of interconnected parts that as a whole exhibit one or more properties (behavior among the possible properties) not obvious from the properties of the individual parts."

Complex Systems Have Complex Failures. That's Cloud Computing [14]. The complexity of the system failure is proportional to the complexity of the system, more complex system the more complex the failure. The most annoying problem of Cloud Computing is the complexity of its system. Those people who run large clusters of virtualized servers with extensive storage farms and networking backbones know exactly the meaning of complex failures [14].

There lots of cases in the past where companies who have had complex systems fail; usually accounting, ERP or supply chain management projects and those companies also fail and go out of business.

The failure of the Amazon EC2 has resulted in unveiling a number of technology strategies in cloud infrastructure as being not up to the mark. Although due to inadequate evidence, it seemed that EC2 was so complex that troubleshooting was extremely difficult to locate source of the problem. And complex systems also take a lot of time to recover. There is no solution for complexity, it can only be that the customer must balance of; value provided by the system must outweigh the risk of complex failure.

Examples of Failures in Designing Scalable IaaS: Some missteps in building IaaS cloud software and systems.

Multi-hypervisor: End-users don't care about hypervisors. Hypervisors are a commodity. They are the new 'bare metal'. Multi-hypervisor adds code, complexity, creates edge cases galore, and creates all kinds of new requirements such as maintaining VM image repositories for each kind of hypervisor [14].

Complexity with Redundancy: Every redundant system adds complexity. Redundancy can be evil if done wrong. Redundancy adds moving parts. For example, if we consider an individual compute node as disposable, investment in redundancies at the hardware level is not the best use of resources [14].

II. AMAZON FAILURES

Now days cloud computing technology is in great demand for fresher's and enterprise customers. This technology is in great demand for deploying their services on Amazon web services. This has increased the computing infrastructure in Amazon web services. Through Amazon Web services, the reliability of the cloud using those services can be known.

Cloud computing is one technology through which we can start our business. Different customers use different Amazon web services to offer their services. While developing the businesses, they should be aware of limitations of cloud and should design in such a way that they can carry their business even if

failure occurs [15]. Through such failures it brings into focus, the requirement of the architecture and design that are required in the cloud computing to sustain the failures in it.

Every expert in cloud computing will use a successful architecture that is design for failure. Many companies still didn't use this rule till certain limit. It would be due to less of technical knowledge for awareness for high availability cost of operating system and its setup in AWS.

In architecture of cloud some factors should be taken in consideration. The applications in cloud should always be available to the users. The system should not compromise on scalability. The setup of infrastructure should line up all the load of requirements to the system [15]. Failures to a system will result in additional cost.

The system should be design taking into all considerations of failure. If the architecture is well designed keep all the considerations then there won't be any possibilities of failures occurring in the system. With these entire considerations one can benefit in his business.

III. CLOUD FAILURES & DISASTERS

Cloud computing service providers went through a tough time in the past, which turned out quite challenging. It is not that they did not monitored and managed cloud related issues properly; it is simply that failures and disasters can occur at any point of time regardless of monitoring cloud servers so thoroughly and keenly. Nevertheless, cloud computing disasters and failures in the past include reputed and popular names like Microsoft and Amazon, Google, skype.

1. Amazon Elastic Cloud Computing Disaster

A cloud disaster had occurred when Amazon EC2 or Elastic Cloud Compute hit big time players like the Reddit, Quora, Hootsuite, and Sqaurefoot causing them to suffer tremendously. To add more to its numbers, approximately 170 SMBs also went through a major setback as they found it very difficult to run their businesses due to the downtime that Amazon EC2 cloud showed them. This had put almost all the IT organizations running their businesses or daily task on that platform to a standstill [18].

2. Cloud Disaster can be resulted by bad Weather Conditions

There had been cases when bad weather had struck and resulted in cloud failures. This has happened when both Microsoft and Amazon's cloud data centers had been blown off by a thunder lighting strike. Both the cloud servers collapsed because of it, leading big and small organizations to huge losses. Servers didn't give any access and remained non-functional for two consecutive days. Companies had a hard time in recovering and suffered monetary losses [18].

3. Microsoft's Office 365 Cloud Disaster

Microsoft had launched its Office 365 cloud productivity suite. The company faced a worldwide outage with DNS servers failing miserably [18].
These examples, prove that such failures and technical faults are a common issue even with a sophisticated and simple environment like cloud computing. Thus organizations need to keep working on enhancing and improvising their in-house IT infrastructures.

4. Google Doc's Disaster

Although, Google Doc is a public cloud platform where users can upload, share and access information, it still can face troubleshooting issues. Google Docs suffered approximately an hour outage due to which work in terms of daily tasks came to a standstill. It also made US organizations to suffer a lot from it as they could not access or share files with others [18]. This proved to be a major setback for Managed Cloud Provider in terms of monetary losses company suffered as well as their reputation in the market.

5. Google Docs get crashed in a row

Another cloud outage came into view when Google Docs collapsed in the Google HQ. The problems were instantly handled and rectified [18].

6. Skype faces overload

In December 2010, Skype faced some overload that caused 30% of the supernodes to go down. The rest of the supernodes were not able to handle the extra responsibility, creating a "positive feedback loop", which led to a near complete outage [16, 17].

CONCLUSION

After getting aware on cloud and its failure and disasters it may cause, it has become inevitable for the organization that have to put in extra efforts for running their businesses smooth. Organizations need to get well equipped for disaster scenarios by having better disaster recovery methods and having data backup plans. The trick to excel in the competition of business is to work proficiently and resourcefully with a good presence of mind and thus set an example for others at the same time.

Thus these managed Cloud Services are of utmost help in preparing for a better failure recovery, so that no business hours are lost which result into monetary loss. Distributing the backup and data in different geographical locations and multiple cloud providers are the most easy way to reduce downtime and ensure 100% Service level even at the time of disaster.

REFERENCES

[1] http://aws.amazon.com/.
[2] http://www.gogrid.com/.
[3] Netflix. 5 Lessons Weve Learned Using AWS. http://techblog.netflix.com/2010/12/ 5-lessons-weve-learned-using-aws.html, December 2010.
[4] Amazon.com. Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region. April 2011.
[5] GmailBlog. More on today's Gmail issue. http://gmailblog.blogspot.com/2009/ 09/more-on-todays-gmail-issue.html, September 2009.
[6] GmailBlog. Update on today's Gmail outage. http://gmailblog.blogspot.com/2009/ 02/update-on-todays-gmail-outage.html, February 2009.
[7] Google. Post-mortem for February 24th, 2010 outage. https://groups.google.com/group/ google-appengine/browse_thread/thread/ a7640a2743922dcf, February 2010.
[8] PayPal. Details on PayPals Site Outage today. https://www.thepaypalblog.com/2010/10/ details-on-paypals-site-outage-today, October 2010.
[9] Skype.com. CIO update: Post-mortem on the Skype outage (December 2010). http://blogs. skype.com/en/2010/12/cio_update.html, December 2010.
[10] WikiMedia.com. Wikimedia Technical Blog: Global Outage (cooling failure and DNS). http://techblog.wikimedia.org/2010/03/ global-outage-cooling-failure-and-dns, March 2010.
[11] Failure as a Service (FaaS): A Cloud Service for Large-
Scale, Online Failure Drills, Haryadi S. Gunawi , Thanh Do, Joseph M.Hellerstein , Ion Stoica , Dhruba Borthakur , Jesse Robbins . Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report No. UCB/EECS-2011-87,
http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-87.html, July 28, 2011.
[12] Alyssa Henry. Cloud Storage FUD: Failure and Uncertainty and Durability. In Proceedings of the 7th USENIX Symposium on File and Storage Technologies (FAST '09), San Francisco, California, February 2009.
[13] Todd Hoff. Netflix: Continually Test by Failing Servers with Chaos Monkey. http:// highscalability.com/display/Search? searchQuery=netflix+chaos+monkey, December 2010.
[14] http://etherealmind.com/complex-systems-complex-failures-cloud-computing/.
[15] http://articles.economictimes.indiatimes.com/2011-06-14/news/29657125_1_cloud-availability-zones-aws .
[16] Byung-Gon Chun, Frank Dabek, Andreas Haeberlen, Emil Sit, Hakim Weatherspoon, M. FransKaashoek, John Kubiatowicz, and Robert Morris. Efficient ReplicaMaintenance for Distributed StorageSystems. In Proceedings of the 3rd Symposiumon Networked Systems Design and Implementation(NSDI '06), San Jose, California, May 2006.
[17] Boston Computing. Data Loss Statistics. http:// www.bostoncomputing.net/consultation/databackup/statistics/.
[18] http://www.cloudways.com/blog/cloud-failures-disasters-of-2011/.
[19] GmailBlog. More on today's Gmail issue. http://gmailblog.blogspot.com/2009/09/more-on-todays-gmail-issue.html, September 2009.
[20] Google. Post-mortem for February 24th, 2010 outage. https://groups.google.com/group/google-appengine/browse_thread/thread/ a7640a2743922dcf, February 2010.
[21] Sanjay Ghemawat, Howard Gobioff, and Shun-TakLeung. The Google File System. In Proceedings of the 19th ACM Symposium on Operating SystemsPrinciples (SOSP '03), pages 29–43, Bolton Landing, New York, October 2003.
[22] http://cloudcomputing.sys-con.com/node/2174422(5 mistakes to avoid).