International Journal of Engineering & Technology, 7 (3.14) (2018) 432-436

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Comparative Analysis of Fault Tolerance Techniques in Cloud Computing: A Case of Armangerayan Co.

Mohammadreza Rasol Roveicy[1]* and Amir Massoud Bidgoli[2]

[1]Department of Computer, Tehran North Branch, Islamic Azad University, Tehran, Iran
rasoli@live.co.uk
[2]Department of Computer, Tehran North Branch, Islamic Azad University, Iran
am_bidgoli@iau-tub.ac.ir

Abstract

Cloud computing is the outcome of in demand service evolution in computing paradigm of large scale distributed computing. It is known as an adaptable technology as it furnishes integration of software and resources which are highly scalable, and most often are prone to failure. Fault Tolerance is concerned with all the techniques to enable system to tolerate software as well as hardware faults remaining in a system after its development. Recent studies have undertaken by various researchers depict the performance of some techniques such as HA proxy, SHelp, Azure, Hadoop etc. There is no evidence to show the overall performance of Elastic Load Balancing as compared with other architectures. The purpose of this paper is to study the fault tolerance techniques in cloud environment by using two different techniques namely HA proxy and Amazon ELB in order to improve availability and reliability when one server goes down for any reasons such as DOS attacks in the case of Armangerayan Co. Performance of two techniques have been compared while considering important factors as connecting time, processing time, waiting time and fail requests. Results reveal that the Amazon ELB in practice has been performing better than HA-proxy in our case study.

Keywords: *Fault Tolerance; Reliability; HA Proxy; Amazon Elastic Load Balancer; Cloud Computing*

1. Introduction

Cloud computing is an extension of the traditional internet, service oriented architecture, web services using virtual shared computing servers to deploy products, resources, and data bases as utility services (Roveicy and Bidgoli 2017).Resources presented on the cloud can be organized by the vendor, and used by the client (Mittal and Agarwal,2015).To date, wide range of applications in the area of research and development are dealing with big data is the concern of many giant companies. Therefore, the problem of processing such a high scale of data is a major issue. In cloud computing environment the clients outsource their data to the cloud that performs the computing operations and store the results. Hence dependability is of great concern in cloud computing.

Although cloud computing has been widely adopted by the industry, still there are many research issues to be fully addressed like fault tolerance, workflow scheduling, workflow management, security etc. Fault tolerance is one of the key issues amongst all (Bala and Chana,2012).Fault tolerance is the ability of the system to perform its function even in the presence of failures. Fault tolerance intend to accomplish robustness and dependability in the cloud environment (Mittal and Agarwal,2015).

In recent years, many studies regarding fault tolerance techniques in cloud computing have been adopted by various authors. Among those techniques are, SHelp, Hadoop ,Gossip, MPI and HA Proxy. Some of these techniques which are illustrated in Table. 1.

HA Proxy is used to handle server failures in fault tolerant cloud environment. HA Proxy is a free, very fast and reliable solution offering high availability, failover and load balancing mechanism, and proxying for TCP and HTTP based applications. When HA Proxy is running in HTTP mode, both the request and the response are fully analyzed and indexed, thus it becomes possible to build matching criteria on almost anything found in the contents. However, It is worth mentioning that HA proxy can handle multiple servers(Vishonika and Bala,2012).Whereas Amazon ELB offers three types of load balancing that all feature the high availability, automatic scaling and robust security. Among these main features. A Proxy lacks automatic security.

Why studying Amazon ELB?

By studying various documents including academic books on Elastic Load Balancing service(ELB),we arrived at the point that provides the capability to evenly distribute traffic among EC2 instances, in order not to overwhelm a single compute server with request and avoid bottlenecks. It also detects faulty instances and redirects traffic through healthy ones, when available (Martino et al, 2015). The study led us to conduct an experimental study. Moreover, since none of researchers have undertaken ELB architecture we were keen to include this architecture into our case study. Though, HA Proxy already examined by different authors, we were also interested to drive accuracy by re-examining adoption of HA Proxy results. However, the aim of this study is to provide a better understanding of fault tolerance challenges and identifies various tools and techniques used for fault tolerance. The two techniques implemented in our case study are HA Proxy and Amazon ELB to deal with the problem of fault occurrence.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Table. 1:. Comparison of the applied policies in each of the architectures

Fault Tolerance Architecture	Self-Healing	Preemptive Migration	Checkpoint-Restart	Replication	Job Migration	Self-Detection	Other Detection	Group Detection	Fault Mask	Node Recovery	Systemic Recovery	Main feature / Usage
Ha proxy (case study)	Y	Y	N	N	N	N	Y	N	N	N	Y	Uninterrupted
BFT-Cloud	N	N	N	Y	Y	N	Y	N	N	N	Y	Arbitrary Fault Detection
Gossip	N	N	N	Y	N	Y	N	Y	N	N	Y	Optimize than Byzantine
MPI	N	N	Y	N	Y	N	Y	N	N	N	Y	For Parallel Programming
FTM	N	N	Y	Y	Y	Y	Y	N	Y	N	Y	Full Policy
FTM-2	N	N	Y	Y	N	N	Y	N	N	N	Y	Separated Fault Detector
LLFT	N	N	N	Y	N	N	N	Y	N	Y	N	Low Latency
FTWS	N	N	Y	Y	N	N	Y	N	N	N	Y	Workflow
Candy	N	N	N	Y	N	N	Y	N	N	N	Y	Component-Based
Magi Cube	N	N	N	Y	Y	N	Y	N	N	N	Y	Low Redundancy
ELB (case study)	Y	N	Y	Y	Y	N	N	Y	N	Y	Y	Low latency

(Y=yes,N=No)

2. Related Works

(Vishonika and Anju, 2015) on the basis of failure scenarios, were able to determine how frontend HA Proxy server redirects requests that lead to the backend servers.
(Anju and Chana,2012) has used HA Proxy for server failover in cloud implementing a Linux prototype and evaluated it with two web servers, both running same application instances.
Various researchers have provided FT solutions specific to certain cloud delivery models for high availability frameworks, by either using virtual nodes for fault prediction or using user defined APIs to help optimize the cloud performance even in faulty situations.

For instance, (Das and Khilar2013), proposed a virtualization and reactive fault tolerance technique where the fault handler prevents the unrecoverable faulty nodes from having adverse effect on the system while (Tchana et al,2012) analyzed the implementation of fault tolerance by focusing on autonomic repair. Similarly, Kaur et al. examined the implementation of fault tolerance in a complex cloud computing environment with a focus on first server and shortest-job-first along with misses per instruction on large method with fault tolerance property. Their proposed algorithm worked for reactive fault tolerance among servers, reallocating the faulty servers task to new server with minimum load at the instant of the fault cloud infrastructure recorded. Another approach for high availability was discussed by (Singh and Kinger,2013) for client requests by proposing failover strategies for cloud computing using integrated check pointing algorithms.

For overcoming the server response failures albeit so many students try to access cloud servers, (Chen et al, 2016), have proposed a dynamic Cloud load balancing (CLB) architecture. For doing so, they took into consideration both server processing power and computer loading. As a result, making it less lightly that a server will not be able to carry out in ordinate computational requirements. They also designed a cloud load balancing algorithm, with the ability to apply to both virtual web servers and physical servers in order to get each server loading, computing power, and the priority service value their results showed server performance relied on proposed architecture can balance the loading with highly scalable performance. Notwithstanding, the server response time increased with the number of connections for both physical and virtual servers.
To improve the performance of the load balancing algorithm (Daraghni and Yuan 2015), proposed and approached by considering both the load balancing technical factors and structure of the network to perform the algorithm .They have designed and improved load balancing algorithm to perform effectively they constructed overlay network , viz , the functional word (FSW). The proposed approach with an attempt to balance loads of nodes, increased throughput, decreased both the response time and communication overload.

To overcome the problem static load balancing or server response for evaluating load balancing capacity (Liang et al, 2016) Introduced a new method of a cloud balancing (CLB) this method takes into account both servers processing power and computer loading . Therefore, a less likely server will be unable to handle excessive computational requirements. Two algorithms in CLB to prove the proposed approach are also addressed with the requirement. The proposed method aimed at calculating server process power and also obtaining PS values. Their findings have shown that CLB can be applied to application in the cloud, hence allowing smoother system operation. Their results also showed that cloud server performance based on the architecture proposed in their study can balance the loading performance when users logged in at the same time.

To create the capacity of fault tolerance in a cloud computing, methods of creating have been eluded by (Chereghlou, et al. 2015). They proposed policies of implementation of these methods. Next, they offered and architecture for the production of such capacity in Cloud computing by comparing those architectures in terms of the type of the policy performed in the architecture and method of fault tolerance and fault recovery. Their results showed that the discussed models took advantage of fault tolerance techniques in different ways.

To build robust fault tolerance (FT) study have been undertaken by (Ganesh et al 2014).They developed various fault tolerance , algorithms , frameworks to better understand different FT policies like reactive FT policy and proactive FT and FT techniques used on different forms of faults. They expressed some parameters to measure capability of fault tolerance in a cloud computing namely

are proactive (PRO) and reactive (RTO). RTO is the amount of data which is disappeared during a fault. Whereas, RTO is determining the minimum down time for recovery. The results showed that models which are used in combinational forms proactive and reactive provide a higher level of reliability. At the same time, they increased the response time. Rather they considerably increased the network performance. Having compared the two techniques called as reactive and proactive methods, they came to conclude that since availability techniques reduce the system, therefore these techniques are not suited for the systems that need for more availability.

3. Methodology

Currently, a few techniques exist for autonomic fault tolerance in cloud environment. For example, Shelp can survive the software faults for server applications running in virtual machine environment. There is a need to implement autonomic fault tolerance in cloud environment.

In this paper, in order to evaluate fault tolerance in Armangerayan Co. We adopted two techniques known as HA Proxy and Amazon ELB. Both have been examined experimentally. Our work has been laid down on the assumption that how much the existing load balancing approaches meet the load balancing metrics. It is assumed that the both above techniques are load balancers, having high availability.

In the first instance, we started examining cloud virtualized system architecture (HA Proxy) as shown in Figure1. The availability of the servers is continuously monitored by HA Proxy statistics tool on a fault tolerant server. HA Proxy is running on web server to handle requests. When one of the servers goes down unexpectedly, connection will automatically be redirected to another server. In this study we have compared the above techniques with two different scenarios. Scenario 1: "Fig.1": displays HA proxy statistics when the both server are up, in this case, HA proxy can use second backend server to handle user requests. Scenarios 2: "Fig.2" depicts when server 2 breaks down, HA proxy stops sending request to first server. This is usually shown in red color.

Next, we have examined Amazon ELB using two different servers. We have also explored the most important features of Amazon ELB performance metrics. Load balancing hardware, or other resources have been adopted in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid over load. For the purpose of evaluating Amazon ELB, we have attempted to make use of some relevant metrics, such as average latency, and sum of Request Count to show high availability and reliability. The average latency is means by which Amazon ELB can be analyzed.

Latency is round-trip request processing time between load balancer and backend. Tracking backend latency gives us good insight on our application performance. If it is high, requests might be dropped due to timeouts, which lead to unsatisfied users. We have ran a Linux curl command to measure the first byte response to determine if one or more backend web application servers are experiencing high latency

According to the first metric, the server 1 in our case study by examining Amazon ELB has experienced 1 milliseconds which indicates a problem with the backend server(s) rather than a problem with load balancer. While server 2 experiencing average value less than 1 milliseconds which indicate that there is no problem with backend server. This has been illustrated in "Fig. 3"

The other metric for evaluating Amazon ELB is the sum of request count. This metric measures the amount of traffic that our load balancer is handling. While load balancer is keeping an eye on peaks and drops, allows us to alert a drastic change which might indicate a problem with AWS or upstream users like DNS. The number of requests completed or connections made during the specified interval of 1 or 5 minutes with 2 instances, in the first instance there is sum of 500 requests in us-west-2a and 2000 requests in us-west-2b with load balancer name dimension. Hence, this shows that there have been 2500 requests in total.

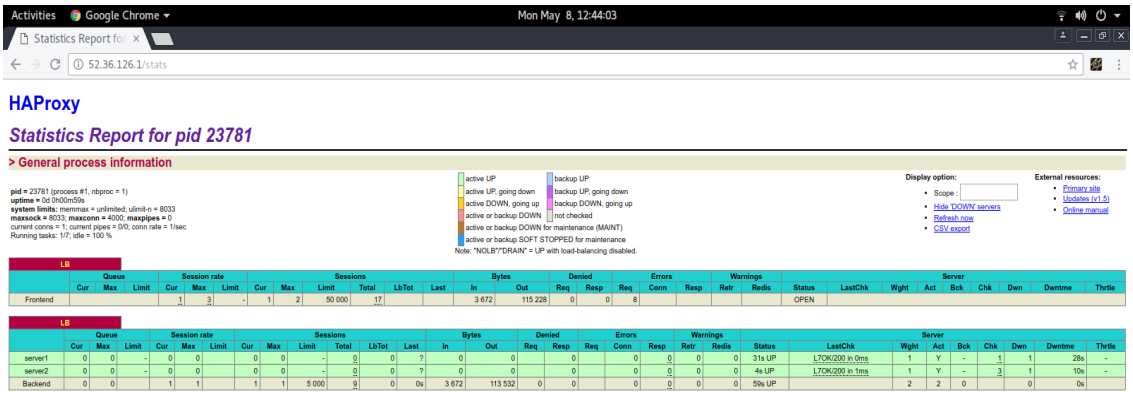


Fig.1 : displays HA proxy statistics when the both server are up

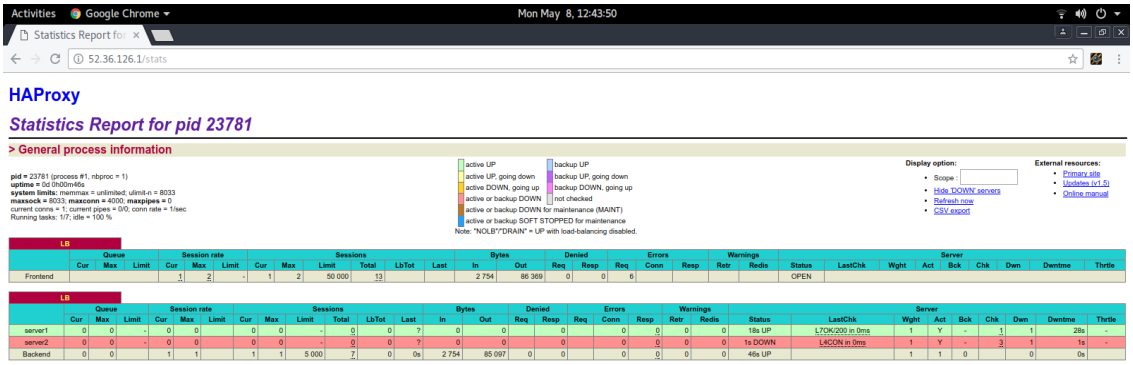


Fig. 2: when server2 breaks down

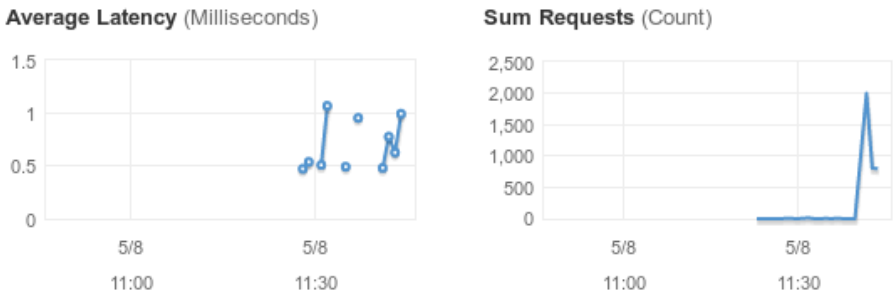


Fig. 3: Average latency and Sum request in Amazon ELB

3. Experimental Results

We have continued our case study by comparing two techniques with the help of some metrics which are displayed in Table 2 and Fig 4. We have used the command for obtaining parameters: "ab-n 100 –c 50/IP "to attain the results presented in Table2. Looking at Table.2 shows that the number of failed requests for HA Proxy has been 52 and for that of Amazon ELB came to be 50.Likewise, maximum and minimum connecting time in terms of milliseconds for HA Proxy has been 351 and 427 ms respectively. Moreover in the case of Amazon ELB this has been respectively 345 and 425 ms. In regard with maximum waiting time in HA Proxy this has been 694 ms. While, maximum processing time in HA Proxy has been 699 ms and that of Amazon ELB has been 660 ms. On the other hand, maximum processing time in HA Proxy has been 699 ms and that of Amazon ELB has been 660 ms. This comparison clearly indicates that, Amazon ELB is more reliable than HA Proxy.

Table 2:. Contrasting HA Proxy vs. Amazon ELB metrics

	Minimum Connecting Time (ms)	Maximum Connecting Time(ms)	Minimum Waiting Time(ms)	Maximum Waiting Time(ms)	Minimum Processing Time(ms)	Maximum Processing Time(ms)	Failed Requests
Ha Proxy	351	427	355	694	358	699	52
Amazon ELB	348	425	352	660	356	660	50

Fig.4 illustrates comparative percentage of the requests served within a certain time, Amazon ELB vs. HA Proxy. As it has been shown, in the range of 50%-80% HA Proxy has priority over Amazon ELB but thereafter, in the range of 90%-100% Amazon ELB taking the lead in terms of number of requests.

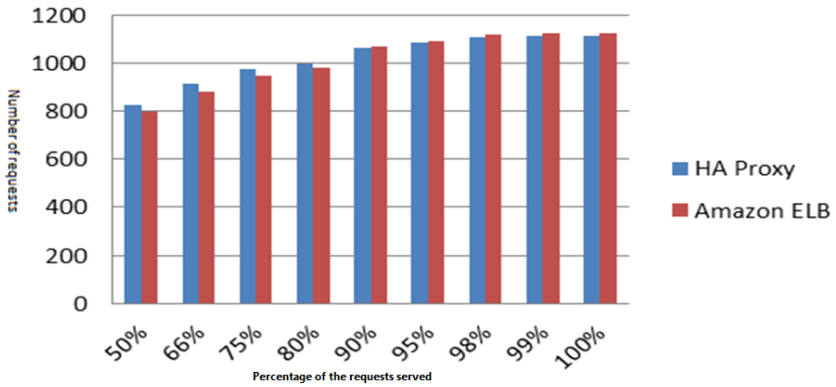


Fig 4:. Comparing percentage of the requests served within a certain time, Amazon ELB VS. HA Proxy (ms)

By looking at Fig.3 it is clearly understood that Amazon ELB having a low latency. Therefore, we can arrive at the conclusion that Amazon ELB as a fault tolerance architecture fall within the category featuring low latency. Again, based on the data it is found to be featuring both proactive and reactive policies. While reactive fault tolerance policies reduces the effect of failures on application execution when the failure effectively occurs; proactive fault tolerance policies keeps applications alive by avoiding failures through preventative measures as it is depicted in Table.1 Therefore, it is preferably to rely more on ELB as the most desirable technique for Armangerayan Co.as our case study for tolerating the unexpected faults.

4. Conclusion

This paper has discussed the fault tolerance techniques in a cloud computing by proposing two techniques known as HA Proxy and Amazon Elastic Load Balancer. Both have implemented to deal with various software server applications in virtualize environment when a server goes down, connection automatically will be redirected to another server.

For the purpose of arriving at empirical results related to fault tolerance in cloud computing, certain metrics have been taken into account to evaluate reliability and availability. According to the results obtained in our case study, Amazon ELB has been performing better HA Proxy. In the other hand, since Amazon ELB featuring Low Latency and being proactive and reactive, it is wise to use Amazon ELB as suitable fault tolerance architecture for our company.

References

[1] Anju Bala , and Inderveer. Chana.(2012). Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing, International Journal of Computer Science Issues. Vol. 9, Issue 1, No. 1, pp.283-293, . Jul 4, 2012

[2] Amal Ganesh Dr. M.Sandhya Dr. Sharmila Shankar.(2014).A Study on Fault Tolerance methods in Cloud Computing, IEEE 2014, 978-1-4799-2572-8/14.

[3] Alain. Tchana, Laurent. Broto, and Daniel. Hagimont. (2012).Approaches to cloud computing fault tolerance. In Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on, pp. 1-6. IEEE, 2012

[4] Beniamino DI Martino,Giusepina Gretella,Antonio Esposito.(2015). Cloud Probability and Interoperability:Issues and Current Trends.BRIEFS IN COMPUTER SCIENCE. Springer Cham Heidelberg New York Dordrecht London,2015

[5] http://www.springer.com/series/10028

[6] Chen et al., CLB: A novel load balancing architecture and algorithm for cloud services,

[7] Computers and Electrical Engineering (2016).

[8] http://dx.doi.org/10.1016/j.compeleceng.2016.01.029

[9] Eman Yasser Daraghmi and Shiam. Ming. Yuan. (2015). A small world based overlay network for improving dynamic load-balancing. Journal of Systems and Software 107: 187-203.

[10] Ganesh, A., Sandhya, M., Shankar, S.(2016). A Study on Fault Tolerance methods in Cloud Computing. IEEE International Advance Computing Conference (IACC),DOI:10.1109/IAdCC.2014.6779432, 21-22 Feb.

[11] Himanshu Agarwal, Anju Sharma.(2015). A Comprehensive Survey of Fault Tolerance Techniques in Cloud Computing , 2015 Intl. Conference on Computing and Network Communications (CoCoNet'15), Dec. 16-19, 2015, Trivandrum, India.

[12] Mohammad Reza rasol roveicy, and Amir massoud Bidgoli.(2017). Migrating From Conventional E-Learning To Cloud-based E-Learning: A Case Study of Armangarayan Co. Proc. Symp. Software Engineering Trends and Techniques in Intelligent Systems, Advances in Intelligent System and Computing 575, pp.62-7, 2017. doi:10.1007/978-3-31957141-6_7

[13] Mannudeep Karla. and Swinderjeet. Singh. (2015). A review of meta heuristic scheduling techniques in cloud computing.

[14] Egyptian Informatics Journal 16(3): 275-295.

[15] Mehdi Nazari Cheraghlou, Ahmad Khadem-Zadeh and Majid Haghparast.(2016). A Survey of Fault Tolerance Architecture in Cloud Computing. Journal of Network and Computer Applications.

[16] http://dx.doi.org/10.1016/j.jnca.2015.10.004

[17] P. Das, and P. M. Khilar.(2013). VFT: A virtualization and fault tolerance approach for cloud computing." In Information & Communication Technologies (ICT), 2013 IEEE Conference on, pp. 473-478. IEEE.

[18] Shang-Liang Chen , Yun-Yao.(2016). Chen.CLB: A novel load balancing architecture and algorithm for cloud services. Computers and Electrical Engineering 0 0 0 (2016) 1–7.

[19] http://dx.doi.org/10.1016/j.compeleceng.2016.01.029

[20] Singh, G., Kinger, S.(2013. A Survey On Fault Tolerance Techniques And Methods In Cloud Computing, IJERT, Vol.2 - Issue 6 (June - 2013).

[21] Sushil Kumar, Deepak Singh Rana, Sushil Chandra Dimri.(2015).Fault Tolerance and Load Balancing algorithm in Cloud Computing: A survey, International Journal of Advanced Research in Computer and Communication Engineering. Vol. 4, Issue 7, July 2015.

[22] Vishonika Kaushal,Anju Bala.(2012) . Autonomic Fault Tolerance using HA Proxy. (IJAEST) INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES. Vol. No. 7, Issue No. 2, 222 – 227.