Evaluating Major Issues Regarding Reliability Management for Cloud-based Applications

Babak Bashari Rad, Pouya Ataei

Asia Pacific University of Technology & Innovation (APU),
Technology Park Malaysia, Kuala Lumpur, Malaysia

Summary
Cloud computing, as a prominent and promising solution to many existing business needs, is expanding at thrilling speed. Thereupon, Cloud providers are continually progressing to provide with more complex services, and request handling approaches. This was due to the growing need in the market, as there is a movement from traditional monolithic applications to an increased composition of micro-services. Increased complexity on the grounds of deployment and runtime management has resulted in inevitable occurrences of failure, which has paid off with declining reliability over the Cloud. Along these lines, it becomes necessary to evaluate major issues regarding reliability management on the Cloud. These issues will guide on how to improve reliability management for cloud applications. On the grounds of the Cloud, an inherently unreliable environment, it is always challenging to ensure a continuous reliability. In this paper, major issues regarding reliability over the cloud, root causes, taxonomies of major failure subjects, with provided recommendation on further research and solution to address such problems will be discussed.
Key words:
Cloud Computing, Reliability Management, Orchestration, Cloud Service, Fault Tolerance, Network Resource

1. Introduction

Cloud concepts have existed for many years, however, the emergence of cloud technologies began in distinct number of phases, going from mainframe computing in the 1950s, to virtual machines in the 1970s, and eventually to grid computing, utility computing, Software as a Service (SaaS), and Cloud computing in the 1990s [4]. Nowadays, with the advent of service based engineering and architectures for providing variety of different services for business needs, like infrastructure as a service (IaaS), data as a service (DaaS), load balancing as a service (LBaaS) and many more, Cloud computing has become at the peak of hype circle of emerging technologies. Being supported by scholarly research predictions, the Cloud computing environment, thanks to its demand access orientation, is dramatically expanding. According to [1], Cloud storage market value of 65 billion USD is estimated by the year 2020, which brings considerable amount of attention to business holders to invest their applications in the Cloud. By the virtue of continues advancements in the paradigms of Cloud computing, Cloud based applications stands in higher need for reliability, as demonstrated in [5]. Besides, these applications are remarkably diverse, some are developed Cloud-specific that utilizes Cloud exclusive features, while some other are more traditional applications that are static and ported to virtual servers running on the Cloud's infrastructures. Moreover, in the interest of compatibility with historical components of an application, a complex implementation of partially micro-services and partially monolithic components can be exerted. Any which way, radical demands of the industry for a reliable service that performs in line with the expected 'Quality of Services' and 'Quality of Experience' remains [1, 2].

Notwithstanding, many papers touching the concept called "reliability", the term is loose, which can arise various understandings and perspectives toward it. Within the course of this study, according to standard TL9000, the reliability of an application is deemed as "the ability of an item to perform a required function under stated conditions for a stated time period". Consequently, this reliability is measured by calculating average time between failures of an application; this is often referred to as Mean Time to Failure (MTTF). The rest of the paper is assembled as follows: section 2 elaborates cloud and its major choke points. Section 3 discusses the overall reliability on the cloud and reliability models. Section 4 delineates the major failures that occurs with supplementary information. Section 5 discusses reliability enhancement techniques over the cloud. Section 6 discusses the reliability evaluation, briefly, and section 7 contains a telecommunication example. Finally section 8 elaborates the summary of the study, and point out future directions and possible solutions.

2. Cloud and its Choke Points

Traditionally, applications were deployed to Cloud utilizing virtual machines as traditional server. These servers then, had to be configured manually using scripts and commands. As demonstrated in [3], this method, while delivering the requirement, was inefficient, thus

Manuscript received July 5, 2017
Manuscript revised July 20, 2017

automation came as the next generation to response to this bottleneck in the deployment of Cloud applications. Automation, despite being far more effective than manual configuration had it limitations too, associations among different components of an application were scant, and concentration was completely with atomic servers, [8]. Thereupon, orchestration frameworks came to the play, being the superior solution, the framework were designed in a way that major operations such as deployments, maintenance, and overall management of application's lifecycle were assigned to services. Besides, the framework was capable of deploying an entire service comprised of various components, taking interdependencies into account. Despite being the ultimate solution, orchestration frameworks, are not free of limitations too. According to [8] and [9], the disk failures are inexorable on the ground of distributed data stores, and in the other hand, VM failures has resulted in declining reliability assurance of Cloud services. In order to address these issues, numerous solutions have been proposed. In what follows, failure analysis, service reliability and solutions will be elaborated from various perspectives in order to point out strengths, limitations, applicability, reliability impact, and recommended future improvements.

3. Reliability over the Cloud

Cloud computing is an extensive platform that embodies assorted technologies like utility computing, grid computing, distributed computing, and autonomic computing, taxonomized under 3 major service models which are infrastructure as a service (Iaas), platform as a service (Paas), and software as a service (Saas) [8]. On the grounds of colossal architecture like Cloud, providing services reliably, underlying countless conditions and scenarios, is a very sophisticated matter. There's always probability for diverse type of failures in the Cloud environment such as resource missing failure, network failure, database failure, overflow failure, timeout failure, software failure and hardware failure [1], [2]-[5],[9]. On the account of reliability analysis, Cloud is constituent of diverse elements and factors, that are each equally critical. These factors such as wide area networks, and heterogeneous software/hardware components, presents practitioners with considerable amount of sophisticated interactions among various sections of the Cloud. Henceforward, the current models are not matured enough to meet the market demands and to ensure the pure reliability of this nexus of computer technologies [3].

4. Failure Analysis of Cloud Services

As demonstrated in [6], [7-9] failures may occurs on various sites of Cloud architecture, such as overflow failures, data resource failure, computing resource failure, virtual machine failure, database failure, disk failure, hardware failure and network failures. In what follows these various types of failures will be elaborated.

- **Overflow**

Under the conditions that the job request exceeds the maximum allowed request set in the request queue, overflow failure transpire, [6]. Upon reaching the maximum number of requests, all the proceeding requests will be obstructed and the users will be experiencing erroneous services or lack of access to that particular service that is where the overflow failures materialize. Moreover, according to [10], if new requests have to hold-up for a longer time, more timeout failure will be apparent, causing even more trouble for both the user and the service provider.

- **Timeout**

Timeout is the predefined estimated amount of time by the service provider for the execution time required to finish the intended job or functionality. If the waiting time for that specific job/request exceeds the predefined estimation, then timeout failure materializes.

- **Data Resource Missing:**

In a typical CMS (Cloud management system), data resource manager (DRM) registers data sources. Thus, if the assigned data resource to DRM are removed but the DRM fails to update and notify about it for certain job/request, data resource missing failure arises, [6], [10].

- **Computing Resource Missing:**

The computer resource missing either physical or virtual, just like data resource missing, may suffer from inevitable chances of failure. A node, or workstation, may turn off without effectively notifying the CMS.

- **Software Failure:**

Just like any computing architecture, Cloud, is suffering from unanticipated events in applications operating on various computing resources. One prominent instances of software failure, is VM (virtual machine) failure, which has attained enormous attention, [2].

- **Database Failure:**

Databases, as shown by Wang et al. [6], despite being designed as reliable as possible, may also be inflicted with runtime and execution failures. Disruptions toward addressing requests, load balancing, clustering, and data flows may fail to deliver the required specifications and results in data access failure.

- **Hardware Failure:**

Cloud computing demands a gargantuan and efficient data centre oriented, hardware structure, which must be reliable, responsive, and compatible with SLA (service level agreement) based applications, [5]. On that account, enterprise level Cloud architectures stand in need of complex networked hardware consisting of a web switches, routers, and access points. As such architectures involves with great complexity, it is currently impossible to develop a hardware facility that provides with zero failure.

Considering all the aforementioned failure analysis, the failure taxonomies falls under groups:

Service Failure: (a) Database Failure, (b) Computing Resource Missing, (c) Data Resource Missing, (d) Timeout, e) Overflow

Software Failure

Hardware Failure

These taxonomies, despite being separated, are highly interconnected through the Cloud architecture; hence, failure in each prompts failures in another. Fig 1, provides an overview of various failure occurrences and their correspondence to each other.

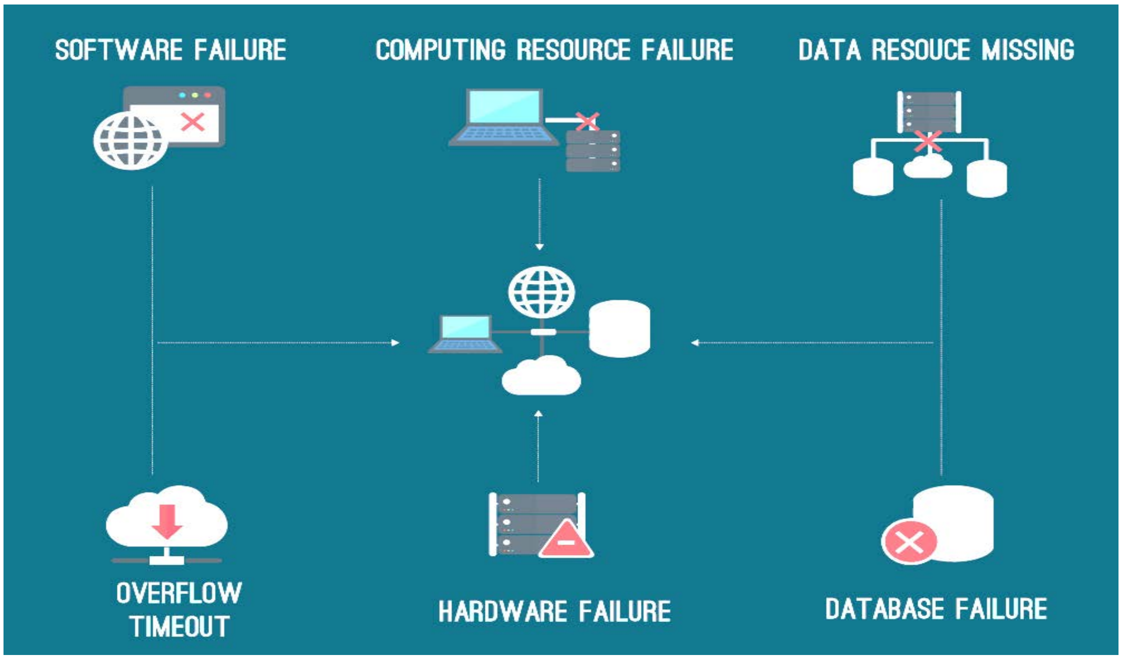


Fig. 1 Failure Analysis of Cloud Services

5. Reliability Enhancement Techniques on the Cloud

In contemplation of alleviating Cloud extra storage and networking resources consumption while meeting the data reliability requirement, many solutions have been proposed. According to [12], fault tolerance, fault prevention, fault forecasting and fault removal are four substantial reliability improvement approaches. The last three approaches pursue an identification and removal of faults that may inflict the system. These approaches are exerted underlying the goal of preventing faults with considerable impacts. Nevertheless, in consideration of the convoluted architecture of Cloud, in which VM failure is inexorable, this goal is impractical [1]. Mechanisms of fault tolerance, oriented toward continual service insurance in the event of failure, are complementary mechanisms with a radically distinct reliability enhancement approach. Throughout the years, many fault tolerance mechanisms have been proposed. These mechanisms are divided in two large groups; ***Reactive Fault Tolerance*** and ***Proactive Fault Tolerance*** [8].

5.1 Reactive Fault Tolerance

Checkpointing is one of the standard fault-mechanisms on the Cloud. Checkpointing operates by capturing the execution states of running tasks (e.g., VM execution units), and have the system reverted back to the latest stable state, in the case of failure occurrences. According to Yuan et al. [10], this method, despite being effective, is quite expensive, considering the resource utilization of periodically capturing images, and using them to resume the failed service. Additionally, in the event of small scale tasks, Checkpointing is definitely an overkill, and a performance overhead. Another prevalent fault tolerance mechanism is replication. Replication is a mechanism that exploits redundant deployments of computer resources. This approach operates by making a backup of available resources like virtual machines. As demonstrated in [1], the resources are grouped into two basic categories: primary and backup. For instance, if the primary VM fails to execute the request, the data will be migrated to the backup VM to handle and finalize the request. Conspicuous approaches have been developed to reduce the costs of implementation by exploiting the degree of redundancy such as 'k-fault tolerance'. Notwithstanding various rationalized models, replication approach is still a very expensive process. Supposing that a task goes incomplete due to failed VM, a backup VM will be assigned to the task. This backup VM, which depend upon data to be retrieved and processed over, again from the central storage servers, brings expensive implications in terms of network resources and time.

5.2 Proactive Fault Tolerance

Software Rejuvenation is a proactive fault tolerance mechanism designed to clean up system internal state so as to inhibit chances of future crash failures. According to [7], this mechanism involves in occasional termination of application in order to clean its internal state. After the cleaning, the application is restarted. This approach is however considerably expensive, considering Cloud environment that offers protection not only against hardware failures, but against performance degradation as well.

Self-Healing is another proactive fault tolerance mechanism that functions in a manner that a system or device that is not operating sufficiently, without human intervention, make the necessary adjustments to restore itself to a more stable state, [8]. This process however does not guaranty an absolute prevention of failures, and despite being proactively healing, some devices may still fail to deliver the functionality.

Pre-emptive Migration is a concept in Cloud computing that inhibits compute node failures from affecting applications running in parallel. This concept prevents failures by pre-emptively migrating application components away from nodes that are about to fail [6]. This technique is also limited to the current state, as failure occurrences still occurs over the Cloud, even while pre-emptive migration has been exerted

6. Reliability Evaluation

Evidently, Cloud as an unreliable environment does not insure uninterrupted resource availability, which proposes the need for an enhanced software solution. A supporting development environment, to utterly overcome the aforementioned issues and constraints has not been developed, yet. However, there were several approaches to address this problem, such as virtual machine placement optimization for Cloud service reliability enhancement by Zhou et al [2], and proactive replica checking by Li, Yang, and Yuan [1], that concentrates on virtual machine failure prevention and storage consolidation. Withal, these nascent approaches are yet in infant stage, and take several years to mature and come to practice. Over and above the current researches, an ultimate orchestration framework would be a near future necessity to administer and grant an enhanced orchestration, allowing both clients and resources to progress with ease and efficiency. Such a framework will have to be connected with an additional monitoring layer that overture a noble approach to service handling. A proper alarm setting, context related callbacks, advanced control loop, and scaling mechanisms are the major role players that has to be emphasized on. Furthermore, as a management framework, the orchestrators has to be capable of providing reliable and continuous delivery of services, by exerting control on the request routing path and a complete knowledge on what application runs under what software version. Another issue to be addressed by this management framework is the rolling upgrade. One of the prominent problems being repeatedly announced in the Cloud is the fact that resources may need to be shut down or manually altered, in order to be updated. This may result in unavailability of a service, which results in a decreased reliability of the overall architecture. On that account, the orchestration framework has to meet the reliability measures necessary to update without needing to shut down or manual interference.

7. The Telecommunication Example

In the field of telecommunication, reliability is a rigid matter, enforcing practitioners to ensure 'five nines' or 99.999%. This implies that the downtime must be 5 minutes at maximum annually, and critical services have to be available in accordance to the stated quality of service. Therefore, in the distributed, complex, and inherently unreliable context of Cloud, approaches and mechanisms of increasing reliability of the applications are essential to the utilization of virtualized services within the practicing companies. Maxis Berhad is a communications service provider in Malaysia that provides its clients with 3G services. Maxis utilized Mobile Cloud Networking (MCN) to achieve this goal. One of the necessary consideration to achieve the goal, was the utilization of Cloud computing in order to virtualize formerly hardware oriented telecommunication software stack such as Evolved Packet Core, Radio Access Network, and IP Multimedia System (IMS). In addition, the company aims to exert an organized management framework of the virtualized

elements in order to increase efficiency and efficacy.
Along these lines of complexity, the company has faced three major reliability issues:

- **Scaling**

Indeed, in order to respond to a varying amount of load, an orchestration framework needs to scale. While deploying the IMS, the service orchestrator does not notify the monitoring systems effectively regarding the volume of requests per second by the IMS instances. In addition, that resulted in performance degradation and inability of scaling-in or –out in line with requests.

- **Failure Handling**

IMS, despite having some mechanisms in the case of failure, does not effectively handle the issue thoroughly. Virtual resources may fail to send periodical ping messages to the monitoring message bus to signal that they are still operating, which leads to orchestrator framework continuity in processing request, even while some of the VMs have failed. This will result in a sub-optimal failure handling.

- **Ineffective Resource Utilization**

The orchestration framework for the management of applications consumes colossal resources for monitoring, error reporting, and fault-tolerance that is otherwise could be utilized for effective mechanisms such as scaling. It influences the overall reliability of the system, as the volume of requests grows.

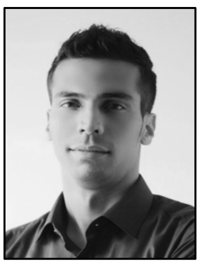
8. Conclusion and Recommendations

Through this paper, an overview of the Cloud, its reliability issues, failure analysis, and enhancement techniques has been identified and discussed. Taking all the aforementioned in consideration, it can be deduced that Cloud, indeed as an unreliable environment does not ensure continuous resource availability. This arises from the amount of complexity involved in the nexus of technologies being configured and setup to orchestrate for a common goal, which is to provide on-demand, off-premise access to software, infrastructures, and platforms. Numerous state-of-the-art approaches overlook the enormous network and storage consumption issues that can materialize while the service is taking, both the proactive or reactive failure recovery mechanisms. These expensive approaches may result in reliability issues themselves, and create another chain of problems for Cloud providers. In the forthcoming, this research can be extended in two directions. First, a further rationalized fault tolerance mechanism that consumes curtailed amount of resources. Second, new mechanisms proposed based on existing ones, with the aim of providing Cloud computing with increased reliability and data access performance.
Based on this, a recommendation would be the, first of all, development of an ultimate orchestration framework that addresses all the existing reliability and availability limitations of the Cloud, and, second, a novel network topology that mitigates the consumption of networking resources. Along these lines, optimal VM placement, host server selection, recovery strategy decision, Cloud storage space consumption, location of replicas, and improvement of data access performance are the major role players that has to extensively accentuated and enhanced.

References

[1] W. Li, Y. Yang, and D. Yuan, "Ensuring Cloud data reliability with minimum replication by Proactive replica checking," IEEE Transactions on Computers, vol. 65, no. 5, pp. 1494–1506, May 2016.
[2] Zhou et al., "Cloud service reliability enhancement via virtual machine placement optimization," IEEE Transactions on Services Computing, pp. 1–1, 2016.
[3] Dudouet, F., Edmonds, A. and Erne, M., “Reliable Cloud-applications: an implementation through service orchestration”. In Proceedings of the 1st International Workshop on Automated Incident Management in Cloud (pp. 1-6). ACM, April 2015.
[4] J. P. Martin-Flatin, "Challenges in Cloud management," IEEE Cloud Computing, vol. 1, no. 1, pp. 66–70, May 2014.
[5] Melo, Matheus, et al. "Availability study on Cloud computing environments: Live migration as a rejuvenation mechanism”, 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2013.
[6] Wang, Q., Kanemasa, Y., Li, J., Jayasinghe, D., Kawaba, M. and Pu, C., “Response time reliability in Cloud environments: an empirical study of n-tier applications at high resource utilization,” In Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on (pp. 378-383), October 2012.
[7] D. Bruneo, S. Distefano, F. Longo, A. Puliafito, and M. Scarpa, "Workload-based software rejuvenation in Cloud systems," IEEE Transactions on Computers, vol. 62, no. 6, pp. 1072–1085, Jun. 2013.
[8] Z. Amin, H. Singh, and N. Sethi, "Review on fault tolerance techniques in Cloud computing,"International Journal of Computer Applications, vol. 116, no. 18, pp. 11–17, Apr. 2015.
[9] B. Balasubramanian and V. K. Garg, "Fault tolerance in distributed systems using fused data structures," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 4, pp. 701–715, Apr. 2013.
[10] D. Yuan et al., "A highly practical approach toward achieving minimum data sets storage cost in the Cloud," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1234–1244, Jun. 2013.
[11] Bibliography: [1] G. A. Gibson and D. A. Patterson, "Designing disk arrays for high data reliability," Journal of Parallel and Distributed Computing, vol. 17, no. 1-2, pp. 4–27, Jan. 1993.
[12] Lyu, M.R., 1996. Handbook of software reliability engineering (Vol. 222). CA: IEEE computer society press.

Dr Babak Bashari Rad received his B.Sc. of Computer Engineering (Software) in 1996 and M.Sc. of Computer Engineering (Artificial Intelligence and Robotics) in 2002 from University of Shiraz; and Ph.D. of Computer Science (Information Security) in 2013 from University Technology of Malaysia. Currently, he is the programme leader of postgraduate studies and senior lecturer in the School of Computing, Asia Pacific University of Technology and Innovation (APU), Kuala Lumpur Malaysia. His main research interest covers a broad range of various areas in computer science and information technology including Information Security, Malware Detection, Machine Learning, Artificial Intelligence, Image Processing, Robotics, Cloud Computing, Big Data, and other related fields.



Pouya Ataei received the B.S. dual degrees in Software Engineering from Asia Pacific University and Staffordshire University in 2015. During 2015-16, he was an active researcher in the industry. His current research interests include Big Data, IOT, Cloud Computing, Software Engineering, and Security. He has completed his M.Sc. in Software Engineering from Staffordshire University.