SIGNIFICANCE OF SOFTWARE RELIABILITY ASSESSMENT IN MULTI CLOUD COMPUTING SYSTEMS A REVIEW

B.J.D Kalyani
Department of Computer Science & Engineering
Research Scalar of Acharya Nagarjuna University, Guntur, India

Dr. Kolasani Ramchand H Rao
Department of Computer Science & Engineering
ASN Women’s Engineering College, Tenali, India

ABSTRACT
Multi Cloud Computing Systems (MCCS) coordinate hundreds of thousands of heterogeneous tasks and aim at delivering highly reliable cloud computing services. At the same time, Software failures cause tremendous losses, thus ensuring the reliability of software becomes increasingly important. Although offering equal reliability to all users benefits everyone. This paper focuses on analyzing the multi cloud computing systems and their challenges, significance of software reliability assessment in multi cloud computing systems through software reliability assessment life cycle and proposes recommendations for improving software reliability in multi cloud computing systems.

Keywords: Multi Cloud Computing Systems, MCCS Challenges, Cloud Software reliability, Multi Cloud Software reliability assessment life cycle.


1. INTRODUCTION
These days’ organizations tend to rely on more than one cloud for services. The clouds could be public clouds, private clouds as well as hybrid clouds [1]. Multi-cloud environments integrate resources from multiple cloud infrastructures [2] as
shown in fig 1. These deployments allow users to take advantage of differences in various clouds, including price, performance, capability, and availability differences.

![Figure 1](image)

**Figure 1** Interoperability in a multi cloud environment: Services offered by different providers interacting with each other.

Also trusting a single cloud is risky as there could be some malicious user or software who is spying on the data being exchanged. So to deal with these issues multi cloud environments have gained importance. The term multi cloud is defined as Multi-cloud strategy [3] is the concomitant use of two or more cloud services to minimize the risk of widespread data loss or downtime due to a localized component failure in a cloud computing environment. As an example, a user may leverage multiple clouds, including community clouds, such as Future Grid [4], and for-pay public clouds, such as Amazon EC2 [5].

The software reliability of the cloud computing is very critical but hard to analyze due to its characteristics of massive-scale service [6] sharing, wide-area network, heterogeneous software/hardware components and complicated interactions among them. Reasons to practice Multi Cloud Strategy are:

- More Architectural Flexibility
- More Technical Control
- Better Security
- Peace Of Mind
- Technical predictability
- Fostering Innovation

2. IMPACT OF SOFTWARE RELIABILITY

Software Reliability is very difficult to achieve due to the complexity of software. The effect of software is getting more significant as an increasing number of digital devices are putting into use. The scale and complexity in software also gives rise to the increase of failure number. The increasing number of faults makes fault location [9] more difficult and the repair cost rise radically.

For safety critical operations like the software used to control cars, turbines and nuclear reactors etc software-controlled systems are essential. Defects in safety-critical software [10] can lead to serious injury or death. For business-critical operations like software that runs in mobile phones, powers web servers and manages
data centers software reliability is also important. Defects in this type of software can lead to significant financial losses. This foundational role means that the reliability of the software is of primary important.

3. SOFTWARE RELIABILITY ASSESSMENT LIFECYCLE IN MCCS

It’s vital that organizations think about how their service will and should operate when a known failure condition [11] occurs. For example, what should the service do when another cloud service on which it depends is not available? What should the service do when it can’t connect to its primary database? To help organizations we recommend software reliability assessment life cycle consists of the following phases as shown in fig 2. Designing a reliable multi cloud computing service [12] and implementing robust revival mechanisms is an iterative process.

3.1 Failure mode and effects analysis

Failure mode and effects analysis is a key step in the design process for any online service. Identifying the important interaction points and dependencies of a service enables the engineering team to pinpoint changes that are required to ensure the service can be monitored effectively for rapid detection of issues. This approach enables the engineering team to develop ways for the service to withstand, or mitigate, faults.

3.2 Design Coping Strategies

Fault-handling mechanisms are also called coping strategies. In the design stage, architects define what the coping strategies will be so that the software will do something reasonable when a failure occurs. They should also define the types of instrumentation that engineers should include in the service specification to enable monitors that can detect when a particular type of failure occurs.

Figure 2 Software Reliability Assessment Life Cycle
3.3 Use Fault Injection
Fault injection [13] is often used with stress testing and is widely considered an important part of developing robust software.

3.4 Monitor the Live Site
Accurate monitoring information can be used by teams to improve services in several ways. Organizations can analyze failure and fault data that instrumentation and monitoring tools [14] capture in the production environment to better understand how the service operates and to determine what monitoring improvements and new coping strategies they require.

3.5 Capture unexpected Faults
When using fault injection on a service that is already deployed, organizations target locations where coping strategies have been put in place so they can validate those strategies. In addition, cloud providers can discover unexpected results that are generated by the service and that can be used to appropriately harden the production environment.

4. SOFTWARE RELIABILITY IN MULTI CLOUD COMPUTING SYSTEMS
All cloud service providers strive to deliver a reliable experience for their customers. In essence, a reliable cloud service [15] is one that functions as the designer intended, when the customer expects it to function, and wherever the connected customer is located. Cloud Services should be designed to:

- Minimize the impact a failure has on any given customer. For example, the service should degrade gracefully, which means that non-critical components of the service may fail but critical functions still work.
- Minimize the number of customers affected by a failure. For example, the service should be designed so that faults can be isolated to a subset of customers.
- Reduce the number of minutes that a customer (or customers) cannot use the service in its entirety. For example, the service should be able to transfer customer requests from one data centre to another if a major failure occurs.

4.1 Recommendations for improving the software reliability in MCS:
- IT management and departments need to realize that cloud providers are an extension of their internal IT department. The key difference is that with internal departments, it can be easier to validate, enforce and administer controls to manage risk.
- Choosing a cloud provider who can demonstrate validation of controls. Some of these controls include audit, data, accessibility, datacenter security controls and data encryption.
- Opt for a private cloud, or a virtual private cloud, where systems are virtually separated from each other through an encrypted environment inside a public cloud.
- Analyze which legacy applications are appropriate for the cloud.
Significance of Software Reliability Assessment In Multi Cloud Computing Systems A Review

- Ask the cloud provider for a definitive disaster recovery plan [16].
- Maintain regular backups of critical cloud-based assets, and protect data through strong encryption.
- Ask for regular security-event alerts from cloud vendors, and ask them to flag specific mission-critical assets [17].
- Seek independent audit reports from service providers for greater transparency.
- Add additional security measures [18] to the cloud such as single sign-on access to multiple cloud applications.

5. CONCLUSION

Software consistency cannot be directly measured, so other related factors are measured to estimate software reliability and equate it among products. This paper describes fundamental reliability concepts and a reliability design-time process for organizations that create, deploy, and/or consume cloud services. It is designed to help decision makers to understand the factors and processes that make cloud services more reliable.

6. ACKNOWLEDGEMENTS

Our thanks to Department of Computer Science and Engineering; Acharya Nagarjuna University, Director, Secretary, Management, Principal & Department of Computer Science and Engineering, Swarna Bharathi Institute of Science And Technology, Khammam for providing necessary facilities to carry out the research work.

7. REFERENCES


