Issues in Designing of Reliable Cloud Applications

Vishwas Dixit
Department of Information Technology
Mahakal Institute of Technology, Ujjain, M.P., INDIA

Abhishek Sharma
Department of Information Technology
Mahakal Institute of Technology, Ujjain, M.P., INDIA

Abstract- Software fault tolerance techniques prepare mechanisms to the software system to prevent system failure from fault occurring. Superfluity of resources in cloud environment makes these techniques as a good approach for developing highly reliable cloud applications. Developing a cloud application with all the critical and non critical components taking into account is not a good practice because of cost of each component makes project costly. For saving money it is necessary for designer considering fault tolerance of significant components. However complexity of cloud application is a barrier in identification of critical components from large amount of distributed components of a cloud application. For assurance of reliability of critical components various reliability improving techniques are used by the designer. Reliable cloud application ensures the reliable delivery of cloud application. In this paper we will discuss issues related with designing of reliable cloud applications.

Keywords – Software fault tolerance, Cloud Application, Significant Components, Distributed Components, Reliability

I. INTRODUCTION

Identification of critical components and designing of fault tolerance mechanisms for the cloud application has aroused wide research interests and has been accepted by cloud application designer. Presently, there are some researches which focus on reliability of cloud applications. Many frameworks and models [1] are exploring approaches to support in the direction of reliability improvement.

II. COMPONENT ARCHITECTURAL ISSUES

Cloud applications consist of a number of components. These components are deployed on distributed cloud nodes and connected with each other through links. The architecture of component provides the basis for implementation of fault tolerance. Issues like visibility (i.e., the set of components that may be invoked directly and indirectly by a specific component [2]) and connectivity (i.e., the set of components that may be invoked directly or used by a given component [2]) should be considered. Requirements of the application should decide the selection of the fault tolerance techniques used for a cloud component. Figure 1 showing the execution of the cloud application considered as failed when one component invokes other. In Case- I No fault-tolerance technique for component Vj or Vi and in Case -II Fault-tolerance technique for component Vj or Vi then, we conclude that in case- I component Vj or Vi is failed but in case- II reason behind the failure of component Vj or Vi is only that fault-tolerance strategy fails.

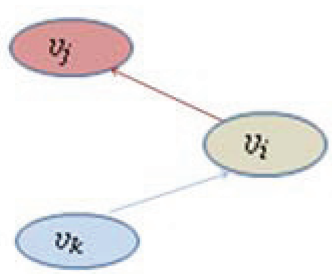

Figure 1. Cloud Component Invocation

III. COST

The cost for building component graph i.e. scale free, random graph or small world model for a cloud application greatly affects design decisions. Cloud applications are complex in nature to be seamlessly developed by integrating services and components rather than building them entirely from scratch. This mainly reduces development cost and time to market.

IV. QoS ISSUES

When cloud application represented by a component graph [3] then modified weights of edge consist of parameters like invocation latency and throughput [4] deals with the impact of the fault tolerance procedure on the end- to- end quality of service (QoS) both during failure and failure free periods.

V. ISSUES IN SELECTION OF CLOUD APPLICATION COMPONENTS

Some of the issues to consider when evaluating whether a particular application is suitable for a cloud are redundancy, the application's ability to migrate, performance, security and cost. Cloud computing strongly recommended applications those demanded low cost analytics, high volume and scenarios of disaster recovery. It is also for applications that are loosely coupled and modular that needs significantly different levels of infrastructure throughout the day, month or seasonal demand. Cloud computing suitable for applications include two or more sources and services with short life spans and scale horizontally on small servers. Some applications relay on large in-memory caches, databases, or data sets so they consume significant amounts of memory and cloud computing is not ideal for the same. Cloud computing is not a good choice in some real time web for example twitter is shifting to its own data center [5]. Cloud computing is not suitable for highly distributed applications that require high-performance file system, I/O needing high bandwidth interserver communications and Mission –critical and core business applications that depend on sensitive data normally restricted to the organization, or requiring a high level of auditabiity and accountability as these process cannot share the high importance data, processing power, and hardware with the third party. Cloud is not recommended for applications that run 24×7×365 with steady demand.

VI.CONCLUSION

In this paper, we have highlighted primary issues before the designing of fault-tolerance mechanism for cloud components. We are also focusing on using these components in the context of performance and nature of cloud components. In particular, it only considers scale free graph and takes account of crash faults and value faults. We can design a scenario on the basis of above issues and one can develop the framework considering more faults like arbitrary fault.

REFERENCES

[1] Vishwas Dixit and Abhishek Sharma, "A Survey on Reliability of Cloud Applications," in *Proc. Int'l Conf. Emerging Trends and Developments in Science, Management and Technology (ICETDSMT '13)*, pp. 476-481, 2013.

[2] Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill., 1997.

[3] Z. Zheng and M.R. Lyu "Component Recommendation for Cloud Applications,"*Proc. Int'l Workshop Recommendation Systems for Software Engineering (RSSE '10),* pp. 48-49, May 2010.

[4] Z. Zheng, C. Zhou, M. R. Lyu, and I. King, "Component Ranking for Fault-Tolerant Cloud Applications," *IEEE Trans. Service Computing*, vol. 5, no. 4, pp. 540-550,Oct-Dec. 2012.

[5] Available:http://www.datacenterknowledge.com/archives/2010/04/16/twiter-will-get-its-own-data-center/.