International Journal of scientific research and management (IJSRM)
||Volume|| 1 ||Issue|| 4 ||Pages|| 251-254||2013||
Website: www.ijsrm.in ISSN (e): 2321-3418

Topic - Review of Cloud Computing in Fault Tolerant Environment

With Efficient Energy Consumption

Princy Gupta[1] Sugandha Banga[2]

Abstract- In this research paper, the exhaustive study has been done on fault tolerance in cloud, fault tolerance's types, limitations are introduced. Further, if fault tolerance is implemented then how much dependable it would be for the real time applications. When multiple instances of an application are running on several machines and one of the machine goes down, there is a need to implement an autonomic fault tolerance technique that can handle such types of faults. The Median Absolute Deviation (MAD) is a measure of statistical dispersion and is more robust estimator of scale than the sample variance or standard deviation, as it behaves better with distributions without a mean or variance and being more resilient to outliers in a data set than the standard deviation.There were various tools like Hadoop, Azzure, HAproxy etc via which fault tolerance can be applied. Scheduling is one of the very important components which can be used to detect performance on the cloud.

Keywords*: cloud computing, fault tolerance, data replication, CloudSim*

1. Introduction

Fault tolerance – building-up of computer system which can perform under the faults and faults are the result of errors which lead to the failure of a system. In order to minimize the failure impact on the system and application execution, failure should be anticipated and proactively handled. Various fault tolerant algorithms and techniques can be used initially to predict the type of failure and then what immediate action should be taken? Failure can be Transient, intermittent, or permanent and design fault or operational errors. Here, we'll deal with the software failure and the most difficult task in designing the software system fault tolerant is to verify that it will meet its reliability requirements. Formation of such system requires a number of models and testing in the fault environment.Overloading on VM can be detected, if the CPU utilization of a host falls below the lower threshold, all VMs have to be migrated from this host and the host has to be switched to the sleep mode in order to eliminate the idle power consumption and If the utilization exceeds the upper threshold, some VMs have to be migrated from the host to reduce the utilization in order to prevent a potential SLA violation. One of the overloading detection is Median absolute Deviation which is quite robust in nature and much sensitive [6]. Fault tolerance can be achieved by applying a set of analysis, implementing algorithms and improvement in design technique which can improve dependability. Dependability of any software system comes with *reliability* (continuity of the service), *availability* (readiness for usage), *safety* (avoidance of catastrophic consequences on

the environment) and *maintainability* (how easily a failed system can be repaired).

2. Limitations of fault Tolerance Algorithm

The cloud computing environment still cannot deliver the quality, robustness and reliability that are needed for the execution of various workflows [1]. Providing fault tolerance requires careful consideration and analysis because of their complexity, inter-dependability. Some are below:

1) Fault tolerance technique doesn't work when multiple instance are running on different VMs [5].
2) All different technologies from vendors of cloud environment need to be unified/ integrated to make dependable system [2].
3) There should be some new technique/ approach to integrate fault tolerance algorithms with existing ones [3].
4) For evaluating the performances of fault tolerance component in comparison with similar ones, a benchmark based method should be developed to ensure high reliability and availability [4].

3. Types of Fault tolerance

Reactive Fault Tolerance- Reactive fault tolerance policies reduce the effect of failures on application execution when the failure effectively occurs. There are various techniques based on these policies like Checkpoint/Restart, Replay-retry, task resubmission, recue workflow, user defined exception handling, retry, Sguard, job migration etc. [1].

Proactive fault tolerance: - The principle of proactive fault tolerance policies is to avoid recovery from faults, errors and failures by predicting them and proactively replacing the suspected components with other working components. Some of the techniques which are based on these policies are Preemptive migration, Software Rejuvenation and using self-healing [1].

4. Requirement of Fault Tolerance 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. It is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. When a fault occurs, there are few techniques which provide mechanisms to the software system to prevent system failure occurrence. The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics etc. When multiple instances of an application are running on several virtual machines and one of the server goes down, there is a need to implement an autonomic fault tolerance technique that can handle these types of faults. To address this issue, cloud virtualized system architecture has been proposed which is implemented using HAProxy and various algorithm can also be implemented for the same. [1]

5. Performance measurement of fault tolerance algorithm

As for cloud workflow systems, similar to many other grid and distributed workflow systems, *scheduling* is a very important component which determines the performance of a whole system. These workflow scheduling systems should be fault tolerant for the failures that occur in the computing environment. Fault tolerant scheduling algorithms can be categorized based on check pointing, traces of data, replication of tasks and resubmission of tasks. Each category has its own advantages and disadvantages. Fault tolerant workflow scheduling is provided by making use of failure probability which is considered in algorithms. Zhang et al [9] described an approach for combined fault tolerance and schedulingworkflow applications in computational grids. Kandaswamy et al. [10] described a mechanism for fault tolerant workflow

by considering check pointing, migration, and over-provisioning. Liang et al. [11] developed a failure prediction model based on failure analysis of BlueGene/L system. Analysis of failure probability requires traces of failure data about each resource in the environment but often cloud providers do not reveal about their infrastructure and most of the time, this information is hidden from the user. Methods like replication and resubmission of tasks do not require any history of information. Few techniques provide fault tolerance by making use of replication. All tasks are replicated to their maximum count which provides very good fault tolerance but uses lot of resources. The tasks which can be executed on highly reliable resources are also replicated and hence the resources are wasted. If there is enough number of resources available then this method will give good fault tolerance power to the workflow scheduling system. Most of the times, number of tasks will be very high when compared to the number of resources available and hence this method may lead to task serialization instead of parallel execution [12].

6. Conclusion

Fault tolerance is about tolerating, avoiding or automatically improving the faults/ errors remained in the software system after its development. This paper discussed the fault tolerance techniques covering its research challenges, tools used for implementing fault tolerance techniques in cloud computing. This paper explains how and where fault tolerance is applicable and how we can calculate the performance of fault tolerance algorithm. To check the level of accuracy, software reliability metrics are also covered.

7. Future Scope :

To maximize their ROI Cloud providers have to apply energy-efficient resource managementstrategies, such as dynamic consolidation of VMs and switching idle servers to power-savingmodes. However, such consolidation is not trivial, as it can result in violations of the SLAnegotiated with customers. It is necessary to develop randomized or adaptive algorithms to improve upon the performance of optimal deterministic algorithms. According to the results of the analysis, we have proposed novel adaptive heuristics that are based on an analysis of historical data on the resource usage for energy and performance efficient dynamic consolidation of VMs. In order to evaluate the proposed system in a real Cloud infrastructure, we plan to implement it by extending a real-world Cloud platform, such as OpenStack. Another direction for future research is the investigation of more complex workload models, e.g. models based on Markov chains, and development of algorithms that will leverage these workload models. Besides the reduction in infrastructure and on-going operating costs, this work also has social significance as it decreases carbon dioxide footprints and energy consumption by modern IT infrastructures.

References

[1] Anju Bala, Inderveer Chana, “Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing”, IJCSI international Journal of Computer Science Issues, Vol.9, Issue 1, No 1, January 2012

[2] Imad M. Abbadi, “Self-Managed Services Conceptual Model in Trustworthy Clouds' Infrastructure”, 2010.

[3] Yang Zhang1, Anirban Mandal2, Charles Koelbel1 and Keith Cooper,” Combined Fault Tolerance and Scheduling Techniques for Workflow Applications on Computational Grids “in 9th IEEE/ACM international symposium on clustering and grid, 2010.

[4] S. Hwang, C. Kesselman, “Grid Workflow: A Flexible Failure Handling Framework for the Grid”,12th IEEE international Symposium on Zigh Performance Distributed Computing (HPDC’03),

Seattle, Washington,USA., IEEE CS Press, Los Alamitos, CA, USA, June 22 - 24, 2003.

[5] Gang Chen, Hai Jin, Deqing Zou, Bing Bing Zhou, Weizhong Qiang, Gang Hu, "SHelp: Automatic Selfhealing for Multiple Application Instances in a Virtual Machine Environment", IEEE International Conference on Cluster Computing, 2010.

[6] Anton Beloglazov, Rajkumar Buyya, " Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers", Wiley interScience, Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia, September 2011.

[7] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems 011; doi:10.1016/j.future.2011.04.017.

[8] Jaya Divya SK, Jaya Nirmala S, Mary Saira Bhanu S, "Fault Tolerant workflow scheduling based on replication and resubmission of tasks in cloud computing", International Journal on Computer Science and Engineering (IJCSE) June, 2012

[9] Y. Zhang, A. Mandal, and K. Cooper, "Combined fault tolerance and scheduling techiques for workflow applications on computational grids," in International Symposium on Cluster Computing and the Grid. IEEE Computer Sociery, 2009, pp. 244-251.

[10] G. Kandaswamy, A. Mandal, and D. A. Reed, "Fault tolerance and recovery of scientific workflows on computational grids," Cluster Computing and the Grid, IEEE International Symposium on, vol. 0, pp. 777-782, 2008.

[11] Y. Liang, A. Sivasubramaniam, and J. Moreira, "Filtering failure logs for a bluegene/l prototype," in Proceeding of 2005 International Conference on Dependable Systems and Networks (DSN'05). Washington, DC, USA: IEEE Computer Society, 2005, pp. 476-485.

[12] R. Prodan and T. Fahringer. Overhead analysis of scientific workflows in Grid environments. IEEE Transactions on Parallel and Distributed Systems, 19(3):378-393, mar 2008.

1 Princy Gupta received the B.E Degree in Information and technology with specialization in software from Staffordshire University, the M.E Degree in Computer Science from Punjab Technical University. Her research interests include OS, Artificial Intelligence, Distributed Computing, Grid Computing and Cloud Computing.

2