SERVICE LEVEL AGREEMENT BASED FAULT TOLERANT WORKLOAD SCHEDULING IN CLOUD COMPUTING ENVIRONMENT

Manpreet Singh Gill[1] and Dr. R. K. Bawa[2]

[1]Research Scholar, Department of Computer SciencePunjabi University, Patiala, Punjab, India
[2]Professor, Department of Computer Science, Punjabi University, Patiala, Punjab, India

ABSTRACT

Cloud computing is a concept of providing user and application oriented services in a virtual environment. Users can use the various cloud services as per their requirements dynamically. Different users have different requirements in terms of application reliability, performance and fault tolerance. Static and rigid fault tolerance policies provide a consistent degree of fault tolerance as well as overhead. In this research work we have proposed a method to implement dynamic fault tolerance considering customer requirements. The cloud users have been classified in to sub classes as per the fault tolerance requirements. Their jobs have also been classified into compute intensive and data intensive categories. The varying degree of fault tolerance has been applied consisting of replication and input buffer. From the simulation based experiments we have found that the proposed dynamic method performs better than the existing methods.

KEYWORDS:

Fault Tolerance, User Classification, Job Classification, Replication, Buffering, Input Buffer,

1. INTRODUCTION

Cloud Computing is a model for supporting universal, appropriate, on-demand access of network to a shared group of configurable resources of computing which can be quickly provisioned and released with minimum effort of management. The cloud computing is a new computing model which comes from distributed computing, grid computing, parallel computing, utility computing, virtualization technology, and other computer technologies. Now days everywhere cloud computing is highly accepted, because every organization wants to get rid of large storage devices. In cloud computing, data is stored permanently not on your personal PC/SERVER but rather on a remote server, which is connected to the Internet . Cloud environment provides the following services to the cloud users.

- Infrastructure as a Service (IaaS)
- Software as a Service (SaaS)
- Platform as a Service (PaaS)

Due to scale and complexity cloud computing environment is very prone to various types of faults. Such faults and failures lead to cloud job failure which can decrease the cloud performance

DOI:10.5121/ijgca.2016.7401

in terms of efficiency and throughput. At the same time, it may violate the Service Level Agreement (SLA) which was decided between the cloud service user and provider to ensure the Quality of Service (QoS).

2. FAULT TOLERANCE

Cloud environment uses the various fault tolerance strategies and counter measures to deal with these issues. Fault tolerance is the property of a system which enables it to continue operating in proper manner in the occurrence of the failure of some of systems components.

In a life-critical system, the existence of a fault-tolerant control system is really important. One of its important functions is to steer the procedure to a safe state whenever unwanted events like faults occur. To achieve this role availability, the reliably of the fault-tolerant control system has to be high. To attain a high degree of availability alongside random failures, one has to recourse to redundancy. Moreover, to avoid common failures, there are distinct necessities on the redundancy, such as independence, reliability, diversity and separation.

3. CLOUD FAULT TOLERANCE

In cloud environment the faults and failures can be at the various levels which include Virtual Machine failure, host failure and network failure etc. These failures can be due to hardware and software failures. Following are the techniques which are used for fault tolerance in Cloud computing environment.

a) Reactive Fault Tolerance

Reactive fault tolerance strategies lessen the influence of failures on application execution when the failure happens efficiently. Following are the various approaches which fall under the reactive fault tolerance category.

- **Replay**: Replay is a fault-tolerant strategy in which the execution of the field restarted once again on the same machine or on a different machine. This pretty execution is initiated by the cloud service not the cloud user.
- **Retry:** in this approach resubmitted by the cloud user for execution. This is the simplest and most widely used method among the public cloud.
- **Job Migration:** If a job is failed during execution, then it is moved on a different machine and its execution is restarted.
- **User Defined Exception Handling:** In case of failure, the procedure defined by the user to handle exceptions is initiated to either recover the execution or to fix the workflow to avoid another instance of failure.

b) Proactive Fault Tolerance

The principle of proactive FT strategies is to avoid the errors, faults and failures by calculating and proactively replacing them with the doubted components of other working components. Some of the methods that are established on proactive fault tolerance strategies are using Software Rejuvenation, migration and Load Balancing, etc.

- **Check pointing:** Check pointing is a mechanism in which the partial results of a job execution are stored from time to time on a stable storage. These partial job results can be used to resume the execution of field job. Checkpoints can be taken and regular time intervals called periodic checkpoints or these can be at regular time intervals called aperiodic checkpoints. Check pointing is the most widely used fault tolerance approach in distributed environment
- **Replication:**In this approach, multiple copies of the same job are executed in parallel. In case one of the Virtual Machine (VM) fails still the job can be executed by the alternate VM. Replication provides more reliability but at the cost of added cost of replicated job.

4. PROPOSED METHODOLOGY

As discussed earlier, cloud computing environment is based on distributed architecture to provide the users with robust and scalable service. Using the dynamic design, the cloud environment is flexible enough to incorporate the changing user requirements from time to time. The fault-tolerant solutions discussed in the literature review addresses a specific kind of failure in a specific predefined way to make cloud environment more reliable. These methods they are very stringent in accommodating the varying or changing requirements of the cloud users. The above discussed solutions, treat the user workload in a static manner. The solutions they provide a varying degree of fault tolerance and performance from one user to another but they are not able to implement this approach dynamic within the cloud user workload. In the following sections we have discussed the design and implementation of the new proposed method which can provide a dynamic fault-tolerant solution to the different kind of jobs within the workload of one specific user.

Research Gap

The various fault tolerance methods studied in this research are based on coarse granularity which makes these methods rigid and unable to support a flexible fault tolerance as per the user requirements. When we talk about cloud computing environment and the various users, who have hosted their business logic applications on Cloud environment, we cannot say that the requirements of every cloud user are same. The priority of these users may vary according to their business requirements. For example, the fault tolerance and reliability requirement of a banking application would be more as compared to an online shopping website.

Cloud computing environment may be hosting all these kind of businesses and their corresponding applications. A very strict cloud fault tolerant job scheduling policy can no doubt prevent and recover from any failures but this will also lead to unnecessary overhead on the cloud resources, which will ultimately decrease the resource utilization. On the other hand, a passive fault tolerant scheduling policy will not be able to deal with the faults and failures since it will lead to increased number of job failures and delayed job execution. In this case, the service level agreement between the service provider and the service user will be violated. It effects the reputation of the cloud service provider in future and it also leads to financial penalties also.

The motivation behind this research is to find an intermediate solution which can handle the cloud applications with adaptive fault tolerance approach. The proposed method should be able to switch between the various fault tolerant solutions as per the user classification. To make the proposed method even more fine granular, it should be able to provide an adaptive fault tolerance to the jobs within the cloud user application.

5. OBJECTIVES

Following are the objectives of the proposed SLA based fault tolerant scheduling approach:

- To propose an adaptive fault tolerance method for cloud job scheduling, which can adjust the degree of fault tolerance for a particular user and workload based on the user and workload classification method. This will help in providing more reliable service to the high-end customers while keeping the fault tolerance overhead to a minimum for low-end services.
- To increase the probability of cloud job getting executed within the set SLA guidelines, while keeping the cost and turnaround time to a minimum. The primary goal is to decrease or eradicate the SLA violations for cloud job execution.
- For quantifying the performance, the proposed method will be compared with the existing fault tolerance solutions based on cost and SLA matrix.

6. ASSUMPTIONS AND CONSIDERATIONS FOR THE PROPOSED METHOD

Following are the research assumptions which have been considered for the proposed fault tolerant solution for SLA based scheduling.

- *Infrastructure Provider:* is an entity which provides the basic infrastructure required for the installation and deployment of cloud resources for providing various cloud services.
- *Cloud User:* Cloud user is the one who uses the services provided by the infrastructure provider would in order to deploy a his/her customised business applications.
- *Fault Tolerant Service:* fault tolerant service provides the pages fault tolerant solutions the cloud user by the various application programming interfaces and programming extensions.
- *Service Provider Roles:* we have assumed that the cloud service provider has a consistent and accurate access to the availability and failure state of the cloud resources. The availability and failure information can be accessed uniformly throughout the cloud infrastructure.
- *Fault Model:* the fault model defines and sets the boundaries for the design and operation of the fault tolerant solution. This consists of the various mechanisms which are to be applied pre or post failure in order to provide a satisfactory service to the cloud user.
- *Fault Tolerance Overhead:* failure overhead is measured according to cost and amount of resources consumed for implementing a particular fault tolerant approach. The failure overhead increases with the degree or extent of fault tolerance.
- *Fault tolerance performance:* This is used to measure the effectiveness of the applied fault-tolerant solution. This is evaluated in terms of successful versus failed job executions. The job waiting time and the overall turnaround time is also a part of the fault tolerance performance metrics.
- *Resource Manager:* Resource manager service keeps track of the resource states. The resources State may vary from ideal to busy. This service also keeps track of the resources which are being used for implementing the replication service. It also keeps track of the machine attributes in terms of wood serial number, hardware architecture in terms of processor speed, the storage capacity and type along with the main memory details. The information related to the busy/ideal state of the machine processor cores is also maintained in the database of the resource manager service.

- *Replica Manager:* Replica manager service keeps track of the details related to the number of active replicas of client job along with their physical location and synchronisation in case of interactive workload.
- *Fault Detection Service:* this service keeps track of the availability of cloud VMs by using the concepts of heartbeats.
- *Buffer Manager:* the buffer manager service manages the process or job-related data buffer to facilitate job processing. The data stored in this service can be used to migrate or resume job execution.
- *Recovery Manager:* This service is responsible for recovering the processing of the failed job by using the recovery methods decided in SLA.
- *Input Buffer:* Input buffer service provides buffering capability to store the job states and input data. This data can be used to restart job execution upon VM failure without transferring the input data once again. This is less costly as compared to job replication.
- We have assumed that the cloud resources are of homogeneous nature in terms of operating system, hardware and network resources and the job migration from one cloud resource to another cloud resource does not lead to any compatibility issues.

7. CLOUD USER CLASSIFICATION

The cloud user classification is one of the basic foundation of the proposed fault tolerant solution. As already discussed, different users may have different requirements for their business applications. We cannot treat all the cloud users with a single fault tolerant or job scheduling policy, and expect the quality of service and user satisfaction. To achieve this, the fault-tolerant solution should be able to cater the needs of different users. To invoke the fault-tolerant solution for a particular class of users, first of all these users should be classified into various classes as per the fault-tolerant requirements. A particular user class will be assigned a particular priority. This priority will be used to map the user would with the corresponding SLA for fault tolerance. For the proposed method, we have considered the following user classes.

- **Gold Users:**Premium user class is the class with the highest fault tolerance priority over the other user classes. During the race for fault tolerance, the premium user's jobs will be given priority would over the other user jobs for handling the various requests.
- **Silver Users:**This user class is given less priority as compared to the premium user class. The workload jobs of this fault tolerant user class will be delayed in case of clash with the premium user class jobs.
- **Bronze Users:**This user class is the class of normal users. This class is provided with all the basic services. This user class as the least priority in terms of fault tolerance implementations.

8. CLOUD JOB CLASSIFICATION AND SLA SPECIFICATION

Similar to the cloud user classification, the cloud jobs of a particular user have further been classified into the following two classes.

- **Compute Intensive:**Compute intensive jobs which require a lot of CPU capacity for processing. For a smaller size of input data, a lot of analytical or predictive steps are performed. Scientific experiments and DNA sequencing are two examples of compute intensive jobs.

- **Data Intensive Jobs:**Data intensive jobs spend most of their time in Input/Output rather than processing. These jobs require a lot of data (terabytes) to be transferred for processing. The fault tolerance requirements of data intensive jobs in different from compute intensive jobs.
- **SLA Specification:**The SLA for cloud users have been considered in terms of number of job failures and job turnaround time.

The users who are processing the workload which is not a real time based application, can afford slight delay in the execution and also some degree of job failures. This was the basic idea behind the proposed technique to provide requirement based fault tolerance while minimizing the fault tolerance overhead in terms of replication and input storage buffer.

9. SLA AND CLOUD USER MAPPING FOR FAULT TOLERANCE IMPLEMENTATION

A rigid fault tolerance policy induces a static overhead of fault tolerance on all the workload. It leadsto increased turnaround time and execution cost. To avoid this, the proposed adaptive method can optimize the execution cost, turnaround time vs. the fault tolerance implementation requirements. The fault tolerance implementation for the cloud users has been proposed as follows.

- Gold Users: Gold users have been considered as high end cloud users, who have no restriction in terms of cost, fault tolerance is the utmost requirement for these users. So for this user class, the SLA policy is oriented towards minimizing the number of job failures. To do that, the proposed policy replicates all the jobs of Gold class users. This adds up to the cost but it provides a high degree of fault tolerance to the user jobs. Due to replication, if one VM fails, its corresponding failed job can be completed on the other VM which was executing the redundant copy.
- Silver Users: For this user class, the job sub class has also been considered. As already discussed the sub-classes are of two types: computer intensive, data intensive. For silver users, the fault tolerance requirement for compute intensive job is different from data intensive job. Computer intensive jobs have been replicated but the data intensive jobs have been backed up by input buffer. If a compute intensive job fails, in order to recover, the parallel running copy of the same job is accepted as the final result. In case of data intensive job, to avoid re-transfer of data, input buffer is used to resume the job execution.
- Bronze Users: In this case job subclass has not been considered. All the jobs submitted by this user class use the input buffer service for fault tolerance.

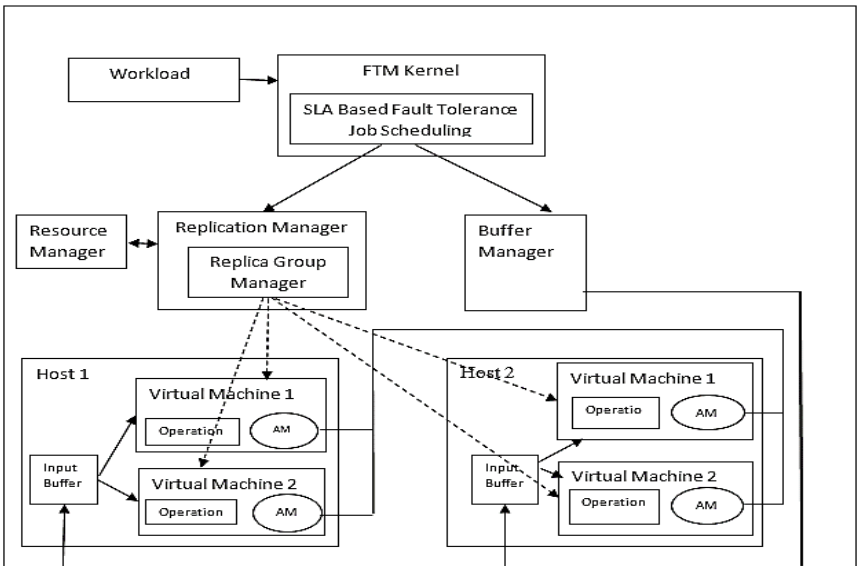

Figure 1: Proposed Architecture

10. CONCLUSION

The proposed issue Fault tolerance is one of the most crucial issue which is faced by the cloud users and cloud service providers. If poorly handled, it can lead to increased waiting time, increased job turnaround time and in worst case increased job failures. Strict fault tolerance policies provide a static fault tolerance but induce additional overhead. Considering this we to have propose an adaptive fault tolerant job scheduling method which should vary the degree of Fault tolerance as per the user requirements. The cloud users have been classified into various classes based on the application requirements along with the job classification.

REFERENCES

[1] P. Kumar, G. Raj, and A. K. Rai, "A novel high adaptive fault tolerance model in real time cloud computing," in 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), 2014, pp. 138–143.

[2] S. Limam and G. Belalem, "A Migration Approach for Fault Tolerance in Cloud Computing," Int. J. Grid High Perform. Comput., vol. 6, no. 2, pp. 24–37, 2014.

[3] A. Ganesh, M. Sandhya, and S. Shankar, "A study on fault tolerance methods in Cloud Computing," in 2014 IEEE International Advance Computing Conference (IACC), 2014, pp. 844–849.

[4] I. P. Egwutuoha, S. Chen, D. Levy, and B. Selic, "A fault tolerance framework for high performance computing in cloud," in Proceedings - 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012, 2012, pp. 709–710.

[5] D. Sun, G. Chang, C. Miao, and X. Wang, "Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments," J. Supercomput., vol. 66, no. 1, pp. 193–228, 2013.

[6] P. Das and P. M. Khilar, "VFT: A virtualization and fault tolerance approach for cloud computing," in 2013 IEEE Conference on Information and Communication Technologies, ICT 2013, 2013, pp. 473–478.

[7] M. Armbrust, A. Fox, R. Griffith, A. Joseph, and RH, "Above the clouds: A Berkeley view of cloud computing," Univ. California, Berkeley, Tech. Rep. UCB , pp. 07–013, 2009.

[8] R. Rajavel and T. Mala, "Achieving service level agreement in cloud environment using job prioritization in hierarchical scheduling," in Advances in Intelligent and Soft Computing, 2012, vol. 132 AISC, pp. 547–554.

[9] S. Fu, "Failure-aware resource management for high-availability computing clusters with distributed virtual machines," J. Parallel Distrib. Comput., vol. 70, no. 4, pp. 384–393, 2010.

[10] K. Lu, R. Yahyapour, P. Wieder, C. Kotsokalis, E. Yaqub, and A. I. Jehangiri, "QoS-aware VM placement in multi-domain service level agreements scenarios," in IEEE International Conference on Cloud Computing, CLOUD, 2013, pp. 661–668.

[11] L. Wu, S. K. Garg, and R. Buyya, "SLA-Based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments," 2011 11th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput., pp. 195–204, 2011.

[12] M. M. Hassan, B. Song, M. S. Hossain, and A. Alamri, "QoS-aware Resource Provisioning for Big Data Processing in Cloud Computing Environment," in Computational Science and Computational Intelligence (CSCI), 2014 International Conference on, 2014, vol. 2, pp. 107–112.

[13] S. Malik and F. Huet, "Adaptive fault tolerance in real time cloud computing," in Proceedings - 2011 IEEE World Congress on Services, SERVICES 2011, 2011, pp. 280–287.

AUTHORS

1. Manpreet Singh Gill

2. Dr. R. K. Bawa