JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Techniques to Enhance Performance of Cloud Computing

Dr. Kamal Kishore
Associate Professor
Modern Group of Colleges, Pandori Bhagat Mukerian.

Abstract
Cloud computing provide solution to handling jobs having high resource requirements on paper use basis. The problem arises as virtual machines deteriorate over the period of time. To this end this paper provides in-depth study of fault handling strategies along with future enhancements. Fault handling provides the way through which cloud system can respond to hardware and software failure that occurred unexpectedly. There are mainly two types of techniques for fault handling in cloud systems that is write back fault handling and write back fault handling. In this literature we are going to analysis various fault handling techniques and make comparative analysis of their advantages/ disadvantages. Also it gives brief study of fault, error, failure that occurred in cloud and a comparison based on parametric evaluation of various techniques is presented.
Keywords-faults; errors; failures; fault handling;

I. INTRODUCTION

Advance Computing is a technology that consists of dynamic environment which emerge distributed computing resources like virtualization, networking, web services etc. It provides end user flexibility and reduced the cost as it provides on –demand services. Today cloud computing become popular over the existing computing technology as it provides many advantages like flexibility, scalability , quality of services and cost saving. It uses shared pool of resources and has capacity to use these resources over the internet and make them remotely available so that it could be used as pay per use basis. Cloud systems provide unlimited computing and storage virtually, also provides abstraction that hides complexity from users.[1] The services are provided by cloud over the internet by public, private or hybrid cloud. On the basis of services it provides there are mainly three kinds of cloud services also known as deployment models:

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

A) Problems in cloud computing

In cloud computing there is many challenges related to data and information handling. These challenges are as given below:

- Security and Privacy: the biggest challenge in cloud computing is security and privacy of information. The data that are uploaded over the cloud have to be secured and this could be done by using encryption, securing hardware and application.
- Portability: The cloud data should easily portable from one service provider to another so it is big challenge. It is not possible to transfer data from one cloud to another because of vendor lock. Also no cloud provider use different languages for their platforms that is an issue for migration.
- Interoperability: For incorporate services from one platform to other platforms is very difficult as developing this type of services is complex.
- Computing performance: As data over the cloud increase so it needs high bandwidth that results in high cost. Thus the computing performance of cloud is affected by bandwidth as if it is low then performance is also low.
- Reliability and availability: Cloud is emerging industry so it is necessary for cloud to robust and available. As cloud services are provided by third party is reliability is a major issue.

B) Causes of Issues in cloud computing

As the Cloud computing services keep on growing and multifaceted nature, so there is a need to guarantee the security, accessibility, and reliability in such services. The fluctuating execution situations, expansion and expulsion of framework parts, frequent updates, online repairs, escalated workload on servers, are the essential reasons that can incite failures and faults in cloud computing services. These reasons make it more important to use fault handling techniques in order to get reliability for real time computing in cloud.[2]

Rest of the paper is organized as follows:

Section 2 presents the literature survey of proactive and reactive fault handling mechanism along with suggested improvement Section 3 gives the comparison of the techniques that described in literature survey. Section 4 gives our contribution. Section 5 gives conclusion and future scope. Section 6 gives references.

II. LITERATURE SURVEY

This section gives in depth study of fault handling mechanism along with problems which are present in the discussed techniques. The classification of fault handling mechanism is as given below:

A) Proactive Fault handling

The main focus of proactive fault handling is on predicting the failures, faults and errors before their occurrence to avoid recovery. It proactively predicts the component that is suspected and replaces it with the working components. It includes various techniques as given below that handle faults proactively:

- Software Rejuvenation
- Self-Healing
- Preemptive migration

- *Software Rejuvenation:*

It is the mechanism in which the overall performance degradation is prevented using periodic reboots. In this after a certain number of migrations in cloud it restarts the system from the initial clean up state. In this technique it accumulates the error conditions in migration and then frees up the resources that utilized in cloud, restart the whole migration process again. As it starts the whole process from the scratch so it needs more effort which can be minimized if progress is saved at the backup server. In case of failure the content matter can be recovered from backup servers.

- *Self-Healing:*

It is the mechanism in which a large task is sub divided into parts and these multiple instances of application are executed over various servers. If from these instances if any failure occurs then it is handled automatically. It also discovers and reacts to the failures that occurred in the system. This technique maximizes the availability, reliability and survivability of the system. The system that utilized self-healing is able to recover from failure during the migration process. As for self-healing another server is maintained for recovering from failure so more cost is encountered.

- *Preemptive Migration:*

It is the mechanism in which feedback loop control is utilized in which application are monitored and analyzed constantly. It allocates the resources to those tasks that are reaching to its deadline and also monitoring is to be done. In this application interoperability is a big challenge as it focus on the tasks that are about to dead and new task will wait until all the previous tasks is fulfilled. In this a control system is used that surveillance all the task and predict if a task is about to fail. Also it shifts this task to another VM when the VM that executing is about to fail. The main disadvantage of this technique is that waiting time is more as another virtual machine is called up for execution.

B) Reactive Fault handling:

The main focus of this technique is to reduce the effect of failure that occurred during the execution of application. Recovering using reactive fault handling is expensive procedure. There are following methodologies that are utilized reactive fault handling:

- Checkpoint/ Restart
- Task resubmission
- Job migration
- Replication

Checkpoint / Restart: It is the technique that mainly focuses on the task level fault handling in applications that execute task for long time. In advance computing for large task the execution time is significantly large and delay/ losses are not allowed during their execution. So for this check pointing is utilized as efficient technique for tolerating fault. The check point is inserted in the task at the various locations where fault occurrence is accepted. After that check pointing is done and if a failure occurs rather than the starting point than task would restart from the last check pointed state. There are following techniques based on check point:

- System level check point
- Application level check point

System level checkpoint: In this operating system level fault handling has to be done. In this checkpoint is maintained by system itself no application level checkpoint is to be done. The developer has to be aware about the checkpoint that is enforced by system. Applications are not considered for check pointing.

Application Level Checkpoint: The checkpoints are inserted in the software as the vulnerable areas are known to the developer. There is no burden of inserting checkpoint over the operating system. Also the capacity of the checkpoint reduced but it is difficult to insert checkpoint in legacy applications.

Job Migration: In this a job is migrated from one machine to another as it may not run on particular machine due to failure that occurred. As complete job is assigned to the server and then server perform the task if server is free. If the current server is busy then job is migrated to another server. If next server is also busy then jobs is put into the queue and when server gets free than task from queue is executed.

Replication: It means copy. In advance computing tasks are duplicated and executed on various resources. It involves sharing of information along with redundant resources that provides consistency. It makes copies of jobs and synchronizes the changes that are done in the copies to main jobs.

Task Resubmission: In this if a task is failed than it would be resubmitted to the same resources or some another resources. The resubmission of the task is done on runtime and if failed job is detected resources are given to the job again.

C) Comparison of various techniques:

Ref.no.	***Technique Used***	***Advantages***	***Disadvantages***	***Future enhancements***
[3]	Proactive fault handling Privacy assured protocol	Bandwidth efficiency Error reduction	Communication failure may occur	Resource provisioning can be applied for handling communication failure
[4]	Proactive fault handling as a service	Cost effective and flexible High availability	Resource availability is less	Provide an algorithm the enhance availability of resources
[5]	PSO based optimization selection	Techniques associated with fault handling are described that can be optimized for future use	No energy efficiency mechanism is considered	May consider mechanism that enhance energy efficiency
[6]	Greedy scheduling algorithm	technique for enhancing the fault handling is considered	Resource allocation is time consuming	Provide efficient resource allocation strategy
[7]	Fault tolerant technique	Server end fault handling is described that could provide sufficient time for migration of resources before VM deteriorate	Energy efficiency criteria's are missing	Enhanced for considering client end fault tolerant
[8]	Reliability based algorithm	Fault handling capabilities are incorporated within Virtual machines for enhancing reliability	Fault handling mechanism utilizes resources extensively hence energy consumption is high	Provide fault tolerant mechanism that is more energy efficient
[9]	Handle faults in real time computing on the cloud infrastructure.	Reliability is enhanced through the proposed approach	Real time environment is considered but it is not suitable for static environment	May enhanced to use real time environment
[10]	Check point technique is used for managing faults and perform recovery	Fault handling strategy for optimizing performance is considered	Efficient use of energy is not done	Server selection can be done using dynamic placement
[11]	Manage faults in hard real time systems using optimistic fault handling algorithms	Scheduling policy for energy efficiency and fault handling is considered hence achieve both energy and fault handling efficiency	Reliability and quality of service could be an issue	Enhanced for efficient faults tolerance technique
[12]	Handle faults in Virtual Data Centres using the address handling techniques at server end.	Virtualization is considered for optimization that enhance performance of allocation resources	Allocation of resource is static	May enhanced for dynamic allocation
[13]	Handle faults in mobile devices in computing environment and use CAN structure for fault management	Reliability is achieved within mobile advance computing through the proposed approach	Execution time and space requirement is high	Provide enhancement in algorithm to enhance reliability
[14]	Manage the faults that occurred in mesh topology during the connectivity and also gives the mechanism	Parameters enhanced considerably involving area, topology and distance in Wireless network	No energy and fault handling mechanisms are considered	Gives a better fault tolerant method that is energy efficient.

Table 1: Comparison of various Techniques

The plot describes the papers the quantity of papers used for comparative study:

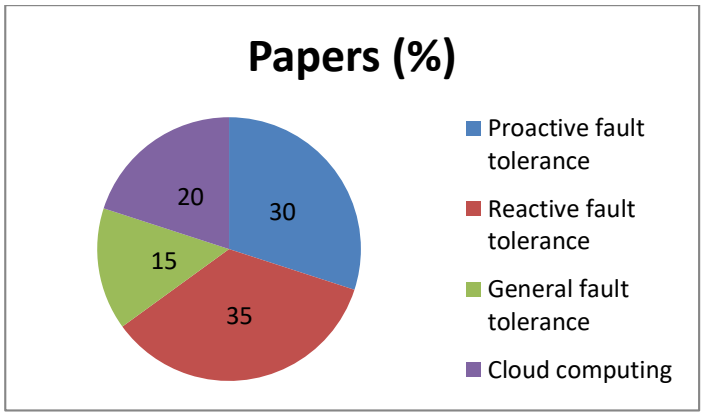

Figure 1: Describing literature used in comparative study

III. COMPARATIVE SIMULATION RESULTS

Various fault tolerant mechanism has been described in the previous section. The results from the most significant literature are extracted and mentioned in this section. The parameters that utilized are execution time and accuracy. Execution time describes total time taken for executing VMs over cloud with fault tolerant mechanism. Accuracy describes correctness of handling faults in system.

No. Of jobs	*Reliability based Algorithm[8]*	*PSO based optimization selection[5]*	*Greedy scheduling algorithm[6]*
100	30.266	26.33	22.11
200	60.091	55.222	44.031
300	90	75.55	66.447
400	119.913	100.445	99.344

Table 2: Comparison based on execution time

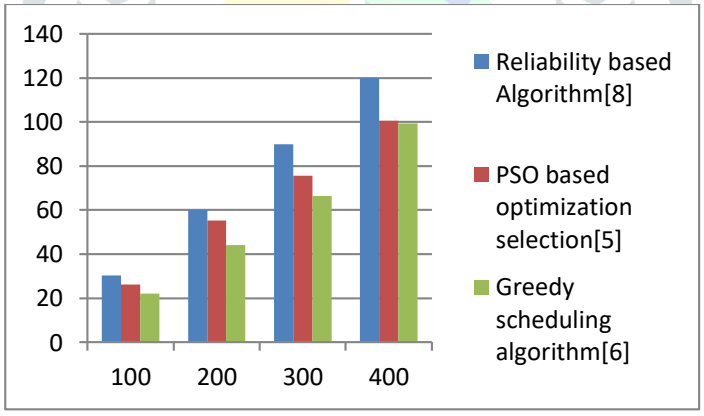

Figure 2: Comparison based on execution time

The result of greedy scheduling algorithm is better than another one but it must be improved by utilizing power aware Vm allocation mechanism.

No. Of jobs	*Reliability based Algorithm[8]*	*PSO based optimization selection[5]*	*Greedy scheduling algorithm[6]*
100	77	68	75
200	79	76	80
300	82	81	85
400	84	87	90

Table 3: Comparison based on accuracy

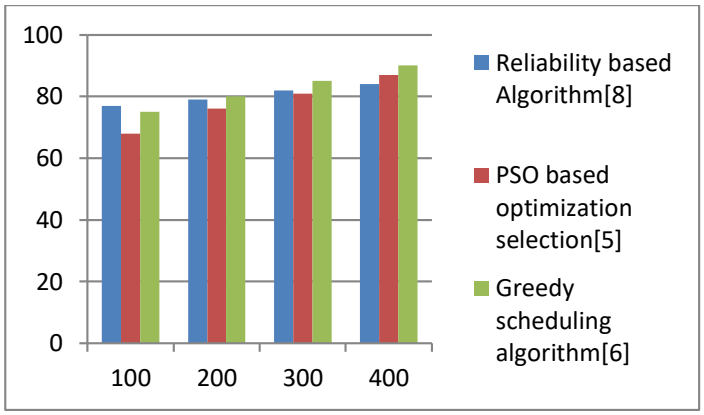

Figure 3: Comparison based on accuracy

The accuracy of greedy approach better but it must be enhanced further.

IV. CONCLUSION

This paper presents the comprehensive analysis of the fault tolerant strategies and their applications. The advantages as well as disadvantage associated with each technique are elaborated through the comparison table. The enhancement that can be made into the existing technique is also highlighted. From the comparative study it is also concluded that least of work has been done towards dynamic placement and resource provisioning mechanism using distance based approach. The brief introduction of cloud is also presented to get in touch with the terminology of advance computing. By considering this work future enhancement to proactive and reactive fault handling can be made to improve efficiency of cloud.

REFERENCES

[1] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1206–1216, May 2015.

[2] Y. Chang and J. Chen, "A Hybrid Cloud for Effective Retrieval from Public Cloud Services," *IEEE*, pp. 61–69.

[3] J. Won, C. Y. T. Ma, D. K. Y. Yau, and N. S. V Rao, "Privacy-Assured Aggregation Protocol for Smart Metering : A Proactive Fault-Tolerant Approach," pp. 1–14, 2015.

[4] J. Liu and J. Zhao, "Providing Proactive Fault handling as a Service for Cloud Applications," pp. 1–2, 2016.

[5] and R. B. Liu, Jialei, S. Wang, Ao Zhou, S.A P Kumar, "Using Proactive Fault - Tolerance Approach to Enhance Cloud Service Reliability," *IEEE Trans. Cloud Comput.*, pp. 1–13, 2016.

[6] P. Stahl, J. Broberg, and B. Landfeldt, "Dynamic Fault-Tolerance and Mobility Provisioning for Services on Mobile Cloud Platforms," *2017 5th IEEE Int. Conf. Mob. Cloud Comput. Serv. Eng.*, pp. 131–138, 2017.

[7] S. K. Suman, "Effective Fault handling Management in Cloud Computing," vol. 6, no. 3, pp. 192–198, 2016.

[8] A. M. Sampaio and J. G. Barbosa, "Towards high-available and energy-efficient virtual computing environments in the cloud," *Futur. Gener. Comput. Syst.*, vol. 40, pp. 30–43, 2014.

[9] R. Jhawar, V. Piuri, and M. Santambrogio, "A Comprehensive Conceptual System-Level Approach to Fault handling in Cloud Computing," pp. 0–4, 2012.

[10] Y. L. Devi, "FAULT TOLEREANE TECHNIQUES AND ALGORITHMS IN CLOUD Cloud system," vol. 4, no. 1, pp. 1–8.

[11] J. Liu *et al.*, "Using Proactive Fault - Tolerance Approach to Enhance Cloud Service Reliability," pp. 1–13, 2016.

[12] O. Hannache, "Probabilistic Model for Evaluating a Proactive Fault handling Approach in the Cloud," pp. 94–99, 2015.

[13] C. Chen, M. Won, R. Stoleru, and G. G. X. Member, "Energy-Efficient Fault-Tolerant Data Storage & Processing in Mobile Cloud," vol. 3, no. 1, pp. 1–14, 2014.

[14] P. M. K. P. Das, "VFT: A Virtualization and Fault handling Approach for Cloud Computing," *Proc. 2013 IEEE Conf. Inf. Commun. Technol. 2013)*, pp. 473–478, 2013.

[15] T. Mahajan. "Tools for Cloud for Resource Management: NOVA and SNMP". *SN Computer Science*, vol. *2* no. 1, pp.1-5,2021,

[16] Alzakholi, O., Shukur, H., Zebari, R., Abas, S., & Sadeeq, M.. Comparison among cloud technologies and cloud performance. *Journal of Applied Science and Technology Trends*, *1*(2), 40-47, 2020.

[17] Arunarani, A. R., Manjula, D., & Sugumaran, V.. Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, *91*, 407-415, 2019.