IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol-6, Special Issue-5, Jun 2018 E-ISSN: 2347-2693

Study of Various Reactive Fault Tolerance Techniques in Cloud Computing

Atul Kumar[1*], Deepti Malhotra[2]

[1]CS and IT, Central University of Jammu, Samba, India
[2]CS and IT, Central University of Jammu, Samba, India

E-mail:kumaraatul1993@gmail.com, deepti433@yahoo.com

Available online at: www.ijcseonline.org

***Abstract*:** Cloud is large, complex and distributed in nature. It has many features like availability, Reliability and performance of the cloud. Due to its large size and complex nature it is prone to various types of fault and failure like data unavailable, data deletion or corruption etc. Cloud are made as fault tolerant system that tolerate any imminent fault or failure, but still sometime fault happens that disrupts the normal service of cloud. Many researchers gave different technique like replication, checkpointing , Retry ,resubmission etc. to tolerate the failure. In this paper, study of various reactive fault tolerance techniques has been done and after analysis conclusion is presented with some future scope.

***Keywords*:** Replication, Checkpointing, Resubmission, Reliability

I. Introduction

Cloud is resource poll that is utilized by user on demand as a service through internet. It has already emerged as the latest trend that is used almost everywhere these days like in banking sector, healthcare industries and in various government, semi-government and other private sectors. Though it broadly used everywhere but still it comes with lot of fault related issues. Fault or failure can occur any anywhere anytime that may lead to loss of data, business loss, financial loss and customers trust loss too. Fault can of many reason like due to some Human error like fault in design or imperfect design,
bugs in application or some security breaches(Hacking, DDOS attacks etc).In cloud ,failure can of various type like Hardware Failure ,Virtual Machine(VM) Failure and Application Failure[1].
Hardware failure: It can be any physical component failure in cloud and can be corrected by service provider only.
Application failure: It is failure of the application that is hosted by cloud .It is detected by customer only but solution is provided by both i.e. either by customer or by cloud service provider only.
VM failure: It is type of failure in which one or more virtual machine fails .It can be detected by both i.e. either customer or by cloud service provider but solution is provided by service provider only.

To avoid all such failure a system must be designed that is fault tolerant in nature. System can be declared as fault tolerant that is able to produce optimal result even after the failure occurs. There are two techniques used to make system fault tolerant i.e. Proactive and Reactive fault tolerance techniques. Proactive techniques(Software Rejuvenation, Self healing and Preemptive migration) are those that used to avoid any upcoming failure through prediction but reactive techniques (Replication, Retry, Sgaurd, Resubmission etc)are those techniques that are used after the fault occurs .In this paper study of various Reactive approaches are done. Section 2 explains reactive techniques and it type in detail whereas section 3 includes the comparison table. Section 4 contains work related to the topic and last section i.e. section 5 includes the conclusion.

II. Reactive Techniques

Reactive fault tolerance approach reduces the effect of failures on application execution when the failure occurs effectively. There are various techniques which are based on these approaches like Checkpointing, restart, Replay and Retry and so on.:[2]
Some Reactive types of fault tolerance techniques are:

1. Checkpointing/Restart [3]
   It can be of various types which can be further classified as following:

a. System level Checkpoints

It provides Fault tolerance to the applications that runs at OS level, but does not manage any fault at application level. This method may be appropriate for the applications that cannot maintain check points by themselves as their workload highly differs from each other or the applications that are not provided with checkpointing service like legacy application.

Advantage	Disadvantage
Developer need to worry about	Application nature of the OS is not

checkpointing	considered by checkpoints
Application unaware(not required) of fact that checkpointing is happening	

b. *Application level Checkpoints*

Developers of the application knows best about its software i.e. known what are the vulnerable areas .So he will create checkpoint for the application in order to reduce any type of loss during failure.

Advantage	Disadvantage
No burden on Operating System	Difficult to create checkpoints for legacy applications.
Capacity of checkpoints size can be reduced dramatically.	

c. Library level Checkpoints

This checkpoint is placed neither too far nor to close to the applications. This mechanism analyzes the running application to find the important information at compile time/runtime. Main difficulty faced by this method, is when to take a specific checkpoint. A situation where immense difference is created by timing of checkpoints, i.e. when process is in loop or some temporary computation is performed i.e. very memory or I/O intensive.

d. Non-incremental checkpoints

At every interval, the whole memory is saved to disk in non-incremental checkpoints. This is favorable when memory is mostly dirty in each interval.

Advantages	Disadvantages
New checkpoint is stored on disk.	On every interval entire memory needed to be written on disk
Technique is more cost effective than incremental checkpoint in case of less storage is limited or expensive.	Interval with big difference is allowed but possibility of subinterval with huge gaps is difficult.

e. Incremental checkpoints

During failure to construct the whole system, checkpoint sequences from start to the latest increment is needed to be stored. This requires higher storage that makes it also costlier than non-incremental checkpoints. But still this mechanism is considered better than non incremental like in case of read-intensive workloads.

2. Replication

Replication is one of the commonly used fault tolerance technique in cloud where failure usually occurs. It is used to enhance the availability of resources in distributed storage environment. Major challenges in replication are replica selection and placement, as the storage system is very large in size and also too complex in nature. In this each data node has their own capabilities thus assigning equal number of replicas in heterogeneous will not improve any performance but occupy unnecessary storage space. Replication based system improves resources availability by allowing access to data even when some of the replicas are unavailable.[4]

3. Retry

This techniques deal with the transient type of faults. When fault is detected a retry mechanism is applied to recover from the effect of fault. This activated mechanism makes the defective module retry its activity for a certain time period. If fault remain longer than the retry period then it will be considered as permanent fault and faulty node will be replaced. If fault disappears in between retry period then it will be considered as transient fault and system will start normal functioning after recovering from it. Retry period must be long enough to make the transient fault disappear and short enough to avoid overlapping of faults.[5]

4. SGaurd

It is based on rollback recovery. Many authors presented a technique for distributed Stream Processing Engines (SPEs) dedicated for commodity servers clusters. Due to checkpointing activities ,there is a less chance of disruption during processing of streams .Also SGaurd uses new forms of file distributed system like GFS(Google file system), HDFS(Hadoop Distribution File System) etc. As the use of checkpoint technique is expensive but SGaurd improves efficiency by best utilization of resources through new distributed file systems.[6]

5. Resubmission[7]

This technique is used when subtask failure lead to the failure of full job especially in case of workflow application in which even single task failure lead to the failure of whole job. Scheduling of jobs within deadline is a challenging task in workflow application makes it vulnerable to fault or failure. In case failure is detected

in the system, job is either resubmitted to the same resource or to some different resource. Most of researchers use replication and resubmission both techniques to improve the fault tolerance. Replicas are used for execution of jobs and in case all replicas fail then the resubmission mechanism is used to resubmit the job.

III. Comparison table of various Reactive fault tolerance technique

Table 1: Comparison of various Reactive fault tolerance techniques in table 1

Technique	Author's	Year	Proposed	Key issues	Features	Other
	N.R. Rejinpaul, L. Maria Michael Visuwasam Qingsong Wei, Bharadwaj Veeravalli,	2012	Framework/algorithm CFTCC (Checkpoint-based Fault Tolerance for Cloud Computing environment), Global Checkpoint Algorithm, Local Checkpoint Algorithm	resolved Real time Computing issue	Timeliness, Redundancy, Availability, High cost but avoid catastrophic loss	Framework used HDFS, AUFS
		2010	CDRM(Cost-effective Dynamic Replication Management Scheme)	Number of Replica	Availability, Load balance, performance	HDFS
Replication[9]	Bozhao Gong Lingfang Zeng, Dan Feng Abdallah M. Saleh, Janak H.	1988	Transient fault model	Differentiate between permanent and temporary faults, Issue related to fault overlapping	Analyze transient fault to derive optimum retry period, Online fault detection scheme	VLSI circuit
Retry[5]	Patel					
SGaurd[6]	YongChul Kwon, Magdalena Balazinska, Albert Greenberg	2008	Fault tolerance technique for distributed SPEs(Stream processing engines)	Save memory resources, Transparent checkpoint	Rollback recovery, Availability	GFS/HDFS
Resubmission[10]	G. Yao, Y. Ding, K. Hao	2017	ICFWS (Fault Tolerant workflow scheduling algorithm)	Meet the soft deadline of workflow	Timeliness, availability	--

IV. Related work

Eman AbdElfattah at al[11] proposed a fault tolerant model to improve the reliability of system that uses replication and resubmission technique. The model produced better results that are compared with another already existing model that uses same techniques i.e. replication and resubmission. The proposed algorithm reschedules the failed task to some highly reliable Virtual Machine (VM).

Mohit Kumar Gokhroo et al[12] proposed a fault detection and mitigation approach. It detect fault in VM at early stage and avoid it or fault at runtime .If fault in VM is detected then necessary action are taken like restarting job at new VM or migrating job to some other healthy VM.

Vitor Barbosa Souza et al[13] focused on the fault that occurs in Fog to Cloud Architecture that manage resources of both fog and cloud. Both proactive and reactive approach is used for fault recovery of element of network by modeling them as Multidimensional Knapsack problem (MKP).

P. Padmakumari et al[14] proposed a hybrid approach to tolerate fault in Virtual machine. Aim is to improve the reliability and availability of the VM. Initially for proactive approach two test are given for failure prediction i.e. Acceptance test and Reliability assessor and If failure occurs in both the test then reactive approach also provided to deal with the failure of VM.

Omer Subasi et al[15] provides a technique to counter the two types of common failure i.e. Fail-stop errors and Silent Data Corruptions (SDCs) in HPC(High Processing Computing).Selective
Replication based scheme is used that replicates application at runtime.

V. Conclusion

Fault occurrence is normal in cloud computing and various approaches like proactive and reactive techniques are used to deal with them. Proactive techniques are considered better than reactive techniques as these techniques predict and avoid failure before its occurrence but sometime these techniques too fails in which reactive techniques are used to overcome the limitations of proactive techniques. For better results both techniques must be used simultaneously. In this paper, it has been observed that there is a need to improve the performance metrics of the various reactive fault tolerance techniques by designing the various algorithms with lesser time complexities**.**

References

[1] S. Prathiba and S. Sowvarnica, "Survey of failures and fault tolerance in cloud," in *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, 2017, pp. 169–172.

[2] G. R. Kalanirnika and V. M. Sivagami, "Fault Tolerance in Cloud Using Reactive and Proactive Techniques," *Int. J. Comput. Sci. Eng. Commun.*, vol. 3, no. 3, pp. 1159–1164, 2015.

[3] S. Kadekodi, "A Compression in Checkpointing and Fault Tolerance Systems," *ACM Ref. Format Saurabh Kadekodi ACM Trans. Em-bedd. Comput. Syst. V, N, Artic. A*, vol. 8, 2013.

[4] A. Rajalakshmi, D. Vijayakumar, and K. G. Srinivasagan, "An improved dynamic data replica selection and placement in cloud," *2014 Int. Conf. Recent Trends Inf. Technol. ICRTIT 2014*, vol. 3, no. 3, 2014.

[5] A. M. Saleh and J. H. Patel, "Transient-Fault Analysis for Retry Techniques," *IEEE Trans. Reliab.*, vol. 37, no. 3, pp. 323–330, 1988.

[6] Y. Kwon, M. Balazinska, and A. Greenberg, "Fault-tolerant Stream Processing using a Distributed, Replicated File System."

[7] T. Nadu, "Fault tolerant workflow scheduling based on replication and resubmission of tasks in Cloud Computing," *Int. J. Comput. Sci. Eng.*, vol. 4, no. 6, pp. 996–1006, 2012.

[8] L. M. M. Visuwasam, "Checkpoint-based Intelligent Fault tolerance For Cloud Service Providers," no. 2, pp. 59–64, 2012.

[9] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster," in *2010 IEEE International Conference on Cluster Computing*, 2010, pp. 188–196.

[10] G. Yao, Y. Ding, and K. Hao, "Using Imbalance CharacteristicforFault-TolerantWorkflow

[11] Scheduling in Cloud Systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 12, pp. 3671–3683, Dec. 2017.

[12] E. AbdElfattah, M. Elkawkagy, and A. El-Sisi, "A reactive fault tolerance approach for cloud computing," in 2017 13th International Computer Engineering Conference (ICENCO), 2017, pp. 190– 194.

[13] M. K. Gokhroo, M. C. Govil, and E. S. Pilli,"Detecting and mitigating faults in cloud computing environment," 3rd IEEE Int. Conf. , 2017.

[14] V. B. Souza, X. Masip-Bruin, E. Marin-Tordera, W.Ramirez, and S. Sanchez-Lopez, "Proactive vs reactive failure recovery assessment in combined Fog-to-Cloud (F2C) systems," in 2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2017, pp. 1–5.

[15] P. Padmakumari, A. Umamakeswari, and M.Akshaya, "Hybrid Fault Tolerant Scheme to Manage VM Failure in the Cloud," Indian J. Sci. Technol. ISSN, no. 948, pp. 974–6846, 2016.

[16] O. Subasi, G. Yalcin, F. Zyulkyarov, O. Unsal, and J.Labarta, "Designing and Modelling Selective Replication for Fault-Tolerant HPC Applications," Proc. - 2017 17th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGRID 2017, pp. 452–457, 2017.

Authors Profile

Atul Kumar completed his B.E in Computer Science from Model Institute of Engineering & Technology, Jammu in 2015.He is currently pursuing M.Tech (2018) in Computer Science from Central University of Jammu, J&K.. His Research interests include Cloud Computing, Distributed Computing and Internet of Things.

Deepti Malhotra received her B.E in Computer Science from Model Institute of Engineering & Technology, Jammu in 2005. She received M.E in Computer Science from Thapar University, Patiala in 2007. She completed her PhD. Degree in computer Science from University of Jammu in 2013.She has published many research papers in various International Journals and conferences. She is currently working as Assistant Professor in Central University of Jammu in department of CS & IT. Her Research interests are Parallel & Distributed Computing, Grid Computing, Cloud omputing, and Data Mining.

