Proactive fault tolerance in cloud: A research perspective review

Mehak Sharma, Moin Hasan and Major Singh Goraya
Lovely Professional University, Phagwara Punjab-144411

Abstract— **Fault tolerance is one of the primarily focused research areas in the field of cloud computing. Most of the existing fault tolerance frameworks are based on reactive approaches. However, due to the advancement in artificial intelligence techniques, there is a wider scope of research in proactive fault tolerance. Therefore, this paper presents a comprehensive review of the proactive fault tolerance frameworks in cloud. In addition, possible research directions are also described.**

Index Terms— **Cloud computing, fault tolerance, artificial intelligence**

I.INTRODUCTION

Cloud is an Internet-based service provisioning paradigm which provides various computing services to millions of independent users as well as organizations [1]–[3]. Due to the presence of thousands of commodity servers and their complex organization, cloud datacenters are prone to various faults, e.g., hardware faults, software faults, power faults, etc. [4]–[6]. Occurrence of faults causes the interruption in the service delivery and consequently reduces the performance [7]. Therefore, fault tolerance in cloud has been admitted as the major area of research. Fault tolerance is defined as a system in which system keeps on working even if faults are present. Numerous fault tolerance techniques, algorithms, and frameworks have been evolved in the literature. Broadly, the fault tolerance techniques are classified as *proactive* and *reactive* [8]. Proactive techniques forecast the probable fault occurrence and take necessary preliminary actions to prevent the system. On the other hand, reactive techniques handle the faults after their occurrences. Due to the advancements in the field of artificial intelligence and machine learning, there is a vast scope of research towards proactive fault tolerance in cloud. Therefore, this paper presents a brief research perspective review of proactive fault tolerance in cloud. This review paper is intended for the young researchers working towards the fault tolerance through artificial intelligence in cloud.

The paper is organized as follows: Next Section describes the background of fault tolerance in cloud followed by the review of eminent proactive fault tolerance frameworks. Results and discussion are included in the subsequent Section followed by the conclusions.

II.FAULT TOLERANCE IN CLOUD

Broadly, fault tolerance in the distributed computing environments (e.g., cloud, grid, etc.) can be achieved through two basic approaches, viz. proactive approaches and reactive approaches. The fault tolerance techniques based on proactive approaches include *self healing*, *preemptive migration*, and *system rejuvenation*. The self healing technique is based on the biological process of the organisms [9]. In this technique, the system uses machine learning to memorize the previous fault occurrences and their respective handling [10]. If a fault (which has been occurred in the past) occurs, the system applies the same tolerance process which was used earlier to successfully heal from the fault. In preemptive migration, the system forecasts the machines which may observe any fault soon using probability theories and AI techniques [11]. The tasks from the fault probable machines are then preemptively migrated to other safer machines [12]. System rejuvenation technique tolerates the faults by taking periodical backups of the system followed by a system cleaning process [13], [14]. There are two categorizations of system rejuvenation, namely *full system rejuvenation* and *partial system rejuvenation*.

Other is, reactive fault tolerance techniques include *replication*, *reactive migration*, and *checkpoint restart*. Replication is considered as one of the easiest and effective fault tolerance method. In this technique, multiple resources are assigned for an application. If one of the assigned resources observes fault, the application does not fail, but remain continuous on other resources [15], [16].Replication can be done using two methods, namely *active replication* and *passive replication*. In active replication, all the resources simultaneously execute the application. In passive replication, only one resource executes the application, while the other resources remain idle and provide the execution backup. In reactive migration, when a machine observes a fault, the application being executed on it is migrated to some other fault free machine [17]. In checkpoint restart, the state of an executing application is periodically updated [18]. After a fault occurrence, the execution of the application is resumed from the last updated state. Figure 1 shows the distribution of various fault tolerance techniques in cloud.

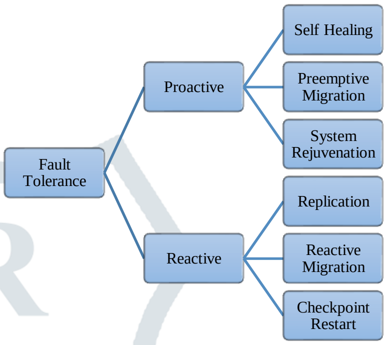

Figure 1. Fault tolerance techniques in cloud

III.PROACTIVE FAULT TOLERANCE FRAMEWORKS

This section presents the review of eminent proactive fault tolerance frameworks proposed in the cloud environment.

In 2009, Sidiroglou et al. [19] proposed self-healing technique based on proactive framework. Authors make use of rescue points, defined as the locations in the code to tolerate specific programmer-anticipated faults. On a fault occurrence, corresponding rescue point is recognized and initiated. Rescue points are recognized using a rescue trace graph on the basis of three parameters, viz. survivability, correctness, and performance.

In 2010, Chen et al. [20] improved the previous framework by assigning weights to the rescue points corresponding to their usability. On a fault occurrence, rescue points are recognized by traversing them in a decreasing order of their respective weights.

In 2012, Egwutuoha et al. [21] proposed a preemptive migration method for fault tolerance in which unhealthy nodes are predicted beforehand using sensors and load of these unhealthy nodes is migrated over newly build healthy nodes.

In 2013, Bruneo et al. [22] proposed a framework for fault tolerance which make use of rejuvenation technique in which any mismanagement and fluctuation in the system is checked periodically. If a system iis found unhealthy during checking, then, the status of all VMs corresponding to this system is saved and rejuvenation is done. Researchers also vary the examination period according to workload of the system.

In 2015, Liu et al. [23] proposed system rejuvenation based fault tolerance framework. The framework consists of a software agent dedicated to periodically examine the system performance in terms of CPU and memory. In case rejuvenation is required, the current state of the system is transferred to an interim node and rejuvenation is carried out.

In 2017, Sun et al. [24] proposed a fault tolerance framework based on preemptive technique. The framework targets to minimize the response time. Therefore, when the arrival rate fluctuates considerably, the critical applications are preemptively migrated to some safe nodes.

IV.RESULTS AND DISCUSSION

This section discusses various aspects drawn from the review and presents the results. Moreover, possible research directions are also discussed.

Table 1 shows the key features and limitations of the fault tolerance frameworks discussed in the previous Section.

Table 1. Features of reviewed frameworks

Framework	**Technique Used**	**Key Features**	**Limitations**
Sidiroglou et al. [19]	Self Healing	Independent of OS kernel alteration	Slow
Chen et al. [20]	Self Healing	Works faster and less overhead	Limited fault applicability
Egwutuoha et al. [21]	Preemptive Migration	Cost efficient	Complex architecture
Bruneo et al. [22]	System Rejuvenation	High availability	Confined to infrastructure layer
Liu et al. [23]	System Rejuvenation	Provisions cloud migration	Unoptimized overhead
Sun et al. [24]	Preemptive Migration	Less response time	Confined fault applicability

In the literature, number of fault tolerance frameworks using proactive approaches are very limited. It is worth noted that proactive fault tolerance generally requires a highly focused system monitoring which consequently increases the system overhead. Due to the consistent advancements in the AI techniques, the system monitoring can be optimized which may result in the overall overhead reduction. Moreover, it is also important to note that most of the proactive fault tolerance frameworks have a limited fault applicability, i.e., they cannot be applied to all types of faults exist in cloud environment. However, as the machine learning algorithms evolve, this problem can also be sorted out by developing machine learning algorithms covering all the fault types.

V.CONCLUSIONS

Considering fault tolerance being one of the important research areas in cloud, this paper reviewed eminent proactive fault tolerance techniques proposed for the cloud environment. In the review, we focused over the categorization of fault tolerance techniques as well as the methodology, key features, and limitations of various proactive fault tolerance frameworks.

Moreover, the research perspective regarding proactive fault tolerance in cloud is also considered to assist the young researchers.

VI.REFERENCES

[1] M. Hasan and M. S. Goraya, “Priority based cooperative computing in cloud using task backfilling,” *Lect. Notes Softw. Eng.*, vol. 4, no. 3, pp. 229–233, 2016.

[2] N. Garg and M. S. Goraya, “Task deadline-aware energy-efficient scheduling model for a virtualized cloud,” *Arab. J. Sci. Eng.*, 2017.

[3] A. Thakur and M. S. Goraya, “A taxonomic survey on load balancing in cloud,” *J. Netw. Comput. Appl.*, vol. 98, no. April, pp. 43–57, 2017.

[4] G. Bosilca, R. Delmas, J. Dongarra, and J. Langou, “Algorithm-based fault tolerance applied to high performance computing,” *J. Parallel Distrib. Comput.*, vol. 69, no. 4, pp. 410–416, 2009.

[5] M. Hasan and M. S. Goraya, “Resource efficient fault-tolerant computing service framework in cloud,” *Int. J. Comput. Sci. Eng.*, vol. 9, no. 03, pp. 51–60, 2017.

[6] M. Hasan and M. S. Goraya, “Fault tolerance in cloud computing environment: a systematic survey,” *Comput. Ind.*, vol. 99, pp. 156–172, 2018.

[7] D. Sun, G. Chang, C. Miao, and X. Wang, “Modelling and evaluating a high serviceability fault tolerance strategy in cloud computing environments,” *Int. J. Secur. Networks*, vol. 7, no. 4, pp. 196–210, 2012.

[8] M. Cheraghlou, A. Khadem-Zadeh, and M. Haghparast, “A survey of fault tolerance architecture in cloud computing,” *J. Netw. Comput. Appl.*, vol. 61, pp. 81–92, 2015.

[9] H. Tu, “Comparisons of self-healing fault-tolerant computing schemes,” in *Proceedings World Congress on Engineering and Computer Science*, 2010, vol. 1, pp. 1–6.

[10] G. Kumar Saha, “Software - Implemented Self-healing System,” *CLEI Electron. J.*, 2018.

[11] A. Polze, P. Tröger, and F. Salfner, “Timely virtual machine migration for pro-active fault tolerance,” in *Proceedings 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, 2011, pp. 234–243.

[12] R. W. Ahmad, A. Gani, S. H. Siti, M. Shiraz, F. Xia, and S. A. Madani, "Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues," *J. Supercomput.*, 2015.

[13] K. Wolter, *Stochastic models for fault tolerance: Restart, rejuvenation and checkpointing*. 2010.

[14] A. Bala and I. Chana, "Fault Tolerance-Challenges , Techniques and Implementation in Cloud Computing," *Int. J. Comput. Sci.*, vol. 9, no. 1, pp. 288–293, 2012.

[15] B. Kemme, M. Wiesmann, A. Schiper, F. Pedone, and G. Alonso, "Understanding replication in databases and distributed systems," 2002.

[16] M. Hasan and M. S. Goraya, "A framework for priority based task execution in the distributed computing environment," in *Proceedings IEEE International Conference on Signal Processing, Computation and Control*, 2015, pp. 155–158.

[17] Y. Ge, P. Malani, and Q. Qiu, "Distributed task migration for thermal management in many-core systems," 2010.

[18] R. Koo and S. Toueg, "Checkpointing and Rollback-Recovery for Distributed Systems," *IEEE Trans. Softw. Eng.*, 1987.

[19] S. Sidiroglou, O. Laadan, C. R. Perez, N. Viennot, J. Nieh, and A. D. Keromytis, "ASSURE : automatic Software Self-healing Using REscue points," in *Proceedings Architectural Support for Programming Languages and Operating Systems*, 2009, pp. 37–48.

[20] G. Chen, H. Jin, D. Zou, B. B. Zhou, W. Qiang, and G. Hu, "SHelp: automatic self-healing for multiple application instances in a virtual machine environment," in *Proceedings - IEEE International Conference on Cluster Computing, ICCC*, 2010, pp. 97–106.

[21] I. P. Egwutuoha, S. Chen, D. Levy, and B. Selic, "A fault tolerance framework for high performance computing in cloud," *Proc. - 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2012*, pp. 709–710, 2012.

[22] D. Bruneo, S. Distefano, F. Longo, A. Puliafito, and M. Scarpa, "Workload-based software rejuvenation in cloud systems," *IEEE Trans. Comput.*, vol. 62, no. 6, pp. 1072–1085, 2013.

[23] J. Liu, J. Zhou, and R. Buyya, "Software rejuvenation based fault tolerance scheme for cloud applications," in *Proceedings IEEE 8th International Conference on Cloud Computing*, 2015, pp. 1115–1118.

[24] D. Sun, G. Zhang, C. Wu, K. Li, and W. Zheng, "Building a fault tolerant framework with deadline guarantee in big data stream computing environments," *J. Comput. Syst. Sci.*, pp. 1–20, 2017.