A RESEARCH REVIEW ON FAULT TOLERANCE TECHNIQUES IN IAAS CLOUD COMPUTING PLATFORM

[1]Rimmy Yadav, [2]Sanjeev Anand, [3]Tarun Sharma [4]Er. Gurjinderpal Singh
[1]Assistant Professor, [2]Research Scholar, [3]Assistant Professor, [4]Head of Department (CSE/IT)
CTIMIT, Maqsudan, Jalandhar

Abstract- The emergence of Cloud computing has brought new dimension to the world of information technology. Cloud computing provides various benefits to its intended clients like on- demand resource provisioning, reduced costs, agility etc. One key research challenge in the cloud computing is to ensure continuous reliability and guaranteed availability of resources provided by it. So fault tolerance is deemed requirement in the cloud computing. In fault tolerant environment, a system can perform its intended functionality even in the presence of failures. In this paper, a study on various fault tolerant techniques has been developed by industrial and research communities. Furthermore, the performance advantages and disadvantages of the fault tolerance techniques in the IaaS model of the cloud computing are analyzed. Issues and future challenges related to the reliability, scalability, performance and availability are reported.

I. INTRODUCTION

The era of cloud computing has change the facet of the information technology (IT) which helps the leading organizations whether it is large scale or medium scale. Traditional information processing systems are now adapting the growing and emerging technologies of the cloud computing.The cloud computing is the next evolutionary steps of distributed computing.The goal of this computing model is make a better use of the distributed resources. Large scale and high performance computing applications are now running smoothly in the cloud computing environment **[1]** because of its performance and reliability. Despite of its high performance, reliability, availability of the resources, issues such as fault tolerance, load balancing, security, Service Level agreement issues (SLA) for cloud service providers, consistency of data, etc., are the major concerns in the cloud computing environment. But Fault tolerance is one of the major issues to retain and enhance the reliability of cloud resources. Fault tolerance is the major concern in the cloud computing. Fault tolerance is the ability of system to perform is task even in the occurrence of the failure [2]. Huge amount costs invested in the cloud will be lost due to any failure such as Node failure, communication failure,Virtual machine failure [3]. Failure of Virtual Machines (VMs)will create a barrier to achieve the performance, reliability.In this paper, failure in IaaS model of cloud computing are elaborated along with different fault tolerant strategies. To comprehend the impact of failures and various fault tolerance approaches on the cloud computing environment, most well- known and widely used approaches are studied. The performance advantages and disadvantages of each FT approach are discussed. The study will help to define the strength and deficiencies of features with respect to the adaptation to the cloud computing services. Several studies related to the faults in IaaS model of Cloud computing with fault tolerance techniques are available in the literature.

The highlighted contributors of the review are:

(a) Analyzing the performance advantages and disadvantages of various fault tolerance approaches in the IaaS cloud environment.
(b) Impact of different technical issues on the performance of the systems and mechanism are identified.

1.1 Cloud Computing and its Service Models

In cloud computing environment, we have to functionality of cloud computing, cloud delivery model (public, private and hybrid), virtualization, fault tolerance or failure over mechanism. Everything is a Service (XaaS) like IaaS (Infrastructure- as- a Service), PaaS (Platform- as- Service), and SaaS (Software-as- a- Service). Failures and inaccessibility of the services and resources in cloud computing are arising as a critical issue. Despite of the failure and unavailability of the resources, still organizations are adopting the cloud computing as a competitive tool to gain a reliable and well position in the market place. Everyone in the field of cloud computing tries to explain its functionality and services in their own way, but there is a need to explain its regularization issues. There are many large scale cloud computing standards such as Amazon EC2, Google App Engine, Microsoft Azure, IBM Blue Cloud, and Netflix etc. so these large scale organizations are investing in cloud computing research and infrastructure to nurture its functionality, availability, reliability and to provide un- interrupted cloud services.

1.2 Benefits of the Cloud Computing

Cloud computing offers diverse shared pool of resources to provide better services to the users. It helps to reduce the makes span of the running applications, lower deployment cost and decreasing the human efforts to manage the cloud resources. Some of the important features of the cloud computing are:

1) Increased Throughput: Vast amount of the VMs deployed in large number of data centers reduces the computational time required to complete a cloud task.
2) Minimize risk of Infrastructure: Self- deployment of VMs helps to reduce the cost of installation of various infrastructure components such as severs installation, topology control management etc.

3) Low cost of entry: the rapid application development helps to reduce the time to market. This helps the decision makes to make better decision to increase its reliability in the market place.
4) Scalability: the performance capabilities of the cloud enable the user of sundry areas and commercial areas to use its services.
5) Availability: fail- over mechanism helps to process the tasks even in the presence of the failures such as, check pointing mechanism, replication strategies, and so on. [1][2].

1.3 Faults in IaaS Platform

Fault tolerance is the major concern in the cloud to avail reliable and scalable services to its clients. Fault tolerance is the ability of the system to continue to its intended functionality even in the presence of the failure. In general a failure represents a condition in which the system deviates from its intended functionality or the expected behavior. The Hypothesized cause for an error is a fault which represents a fundamental weakening in the system. Failure in IaaS model of the cloud computing will deteriorates the performance of other participated service models such as PaaS and SaaS. In IaaS, failure of Virtual Machine (VM) will directly impact on the functionality of running cloud application. There are numerous major reasons behind the fault existence in IaaS cloud computing environment are huge workload, communication link failure, application failure, VMs failure, network topology changes, delay issues, intermittent nodes failure etc. [3][4].

Fault prevention are most widely used approaches to enhance the reliability of components in IaaS and some of them are; VM Restart FT, Checkpoint/ Restart fault recovery, Self-Decentralized protocol with message gossip mechanism to detect the VM failure in IaaS, Proactive FT to detect and recovery from task failure, VM's component failure, Log- based recovery approach to recover lost data due to VM failure or network failure, RabbitMQ with Reactive fault tolerance approach to monitor the aliveness of VM and communication between other VMs, Xen- Hypervisor to control management of VMs. Despite of its advantages, these fault tolerance techniques suffered from various critical issues such as performance overhead and communication overhead using checkpointing/ restart fault recovery approach, migration cost and computational time increasing using preemptive migration technique, non- detection experience by the Monte- Carlo simulation.

II. RELATED WORK

In [3], authors proposed a reliability based optimization framework, named ROCloud to improve the reliability of the applications by fault tolerance in cloud computing. The main idea of their developed framework to identify the significant components whose failures can greatly impact on the application reliability. Authors in [4] developed a distributed application oriented multi- level checkpoint/ restart to provide efficient and transparent fault tolerance in virtual clusters in cloud computing. They combine the existing Checkpointing/ Restart fault recovery mechanism with the virtualization technology to cope with the virtual machine failures in IaaS model of the cloud computing.[5] modeled the systems behavior with a structure- state process to characterize the failure recovery behavior of the virtual machines. The structure space process approach is used to analyze the job completion time of all the virtual machines in cloud computing environment. [6] Highlight the performance anomaly diagnosis in production IaaS cloud often takes a long time due to its inherent complexity. A run time performance anomaly fault localization tool is proposed to diagnosis external faults (e.g., interference from other co- located applications) or internal faults (e.g., software bugs.) in IaaS model of the cloud [7] a decentralized protocol to self-deploy and configure a set of software components distributed over a set of virtual machines is proposed. The deployed process detects and handles the network failures and virtual machine failures in IaaS model of computing cloud. Message gossip protocol, a fault detection mechanism, is used to monitor the state of all Virtual Machines (VMs). The proposed protocol involves a high degree of parallelism.

[26]presented a proactive fault tolerance approach to reduce the overall wall clock execution time of High Performance Computing (HPC) systems. It employs the three major policies, 1) Gather/ Lease a VM from the cloud service provider, 2) Remove the unhealthy VM, 3) An efficient fault recovery mechanism to take necessary actions. Application resiliency is greatly improved by at reduced cost. [9] developed a cost- based fault tolerance scheme to recover from the node failures for the complex tasks and long running files in the cloud computing to minimize the overall computation time of the tasks executed concurrently. For the demonstration of the proposed work, DoomDB , an ego shooter game is designed. The purpose of the game is to kill the nodes in cluster of VM before analytical result is generated in a given time frame. This fine grained approach effectively deals with the nodes failure. [10] Designed an autonomic fault tolerant scheduling for the scientific work flow applications in cloud. Hybrid heuristic approach is applied for scheduling the scientific applications. Virtual migration based fault tolerant technique is implemented to cope with the virtual machine failures. Proactive fault tolerance for prediction of the faulty VMs is used to effectively reduce the execution time of the scientific work- flow applications.[11] designed a log based recovery scheme which helps to recover the lost data. They proposed a distributed storage model based on Reed Solomon in which the indices of enhanced data blocks are used as a password to protect the data.by employing the failure recovery process, data can be recovered without knowing the password. The design of storage model along with erasure code, a failure recovery process, enhance the availability and security of the data in IaaS level of the cloud computing.[12] focused on the compute project of the Open Stack in cloud computing. The compute project (i.e. Nova) provides basic functionality for hosting virtual machines in cloud environment. To enhance the internal node communication in VM clusters, RabbitMQ, an Advanced Messaging Queue Protocol (AMQP) is used in a synchronous fashion. Reactive and Proactive fault tolerance approach is applied to deal VM failure and Incident failure in cloud. Authors in [13] presented a robust scheduling algorithm named ICPCP (The IaaS Cloud Partial Critical Path) with an efficient resource allocation and fault tolerance policy to reduce the overall computational time and computational cot. Workflow failures due to the server failure, task failure due to the dynamic environment of cloud and virtual machine failures due to unbalanced load are many covered in their research work. [14] Presented an efficient intermediate data fault tolerant framework to deal with the intermediate data failure which occurs due to the server failure. [15] Developed a computationally intensive efficient method to evaluate the reliability of the cloud computing systems using Monte- Carlo Simulation (MCS) to deal with the stochastic hardware failures and stochastic VM request. VMs can be added or replaced on the basis of the reliability of each VM's

computation to compute a task[16] developed a probabilistic byzantine tolerance approach which to replicate the tasks on the dynamic replication nodes in case of node failures. Probabilistic assessment output by a node helps to add or remove the nodes to complete the computational task. All the nodes are treated as byzantine nodes to produce the correct output. Indifferent values helps to remove the nodes from the architecture. [17]Developed model defined fault tolerance management approach which deploys FT mechanism following a high level model. Model defined fault tolerance enables developers to define, select FT mechanism in a high level and to automatically utilize them into specific platform. A preemptive Migration, a fault tolerant approach is implemented to handle the VM failure. Faulty virtual machines are replaced with the non- faulty machines. It helps to avoid the host failure and increase the reliability of the resources in the cloud.

[18] Focused on the single- point- of- failure (i.e., Node failure) that frequently occurs in the Hadoop based application systems. Xen- Hypervisor virtualization technology and Open Nebula virtual machine management tool to handle the single- point- of- failure. Dynamic Resource allocation algorithm (DRA) with the heart beat messaging protocol, a failure detection mechanism, is used enhance the availability of the VM. Work load is effectively managed even in the presence of the failure. Authors in [19] presented a well-designed novel scalable and autonomic VM management framework, named Snooze, to recover from the VM failure in IaaS model of the Cloud. Hierarchical configuration is adopted to allow the systems to self- heal from the failures.

[20] Proposed a High Adaptive Fault Tolerance in Real Time Cloud (HAFTRTC) computing. Performance and behavior of the virtual machines are analyzed using the adaptive fault tolerance technique. Based on the reliability if VMs such as bandwidth, ram VM is selected to complete and replicate the task in case of VM failure. Priority based scheduling is applied to select best optimal reliable VM from two VMs with identical reliability based metric. The system continues to work even in the presence of the VM failures.

TABLE I. LOOP HOLES OF PROPOSED/ EXISTING FAIL-OVER TECHNIQUES.

Ref.	Techniques	Disadvantages
[3]	Balance Reduce Algorithm with reactive fault tolerance approach.	• Job completion time increased. • Large amount of recovery time is wasted.
[13]	Robust Scheduling algorithm (ICPCP: The IaaS Cloud Partial Critical Path) with efficient resource allocation, checkpointing fault recovery mechanism is applied.	• Time to Re-boot VM slow down the overall performance. • Computational time of VM varies. • Fault Prediction mechanism not implemented.
[14]	Intermediate data fault framework with two fault tolerant algorithms, 1) Inner_ Task_ IDF, 2) Outer_ Task_ IDF. Master- slave configuration is adopted and task replication.	• Not detect the server failure. • Time overhead varied. • Task replication slows down the performance of algorithms.
[15]	Computationally intensive efficient method using Monte- Carlo Simulations (MCS), Task Replication.	• Monte- Carlo Simulation did not detect all the component failures. • Failed to provide fault tolerance in dynamic configuration environment. • Neglect software bugs, human errors and other sub- systems like external storage, Storage Area Network. • Migration of computational nodes for the load- balancing is not implemented.
[16]	Probabilistic byzantine tolerance.	• Performance slowdown. • Computational cost is large.
[17]	Model defined fault tolerance with FT mechanism and Preemptive migration approach.	• Migration cost and computational time varied.
[19]	Scalable and autonomic VM management framework named Snooze.	• Logs based messages lost due to the performance overloading and failure of Group Leader (GL) module in proposed framework. • Worst performance when scaling to the large scale computing environment.
[13]	Robust Scheduling algorithm (ICPCP: The IaaS Cloud Partial Critical Path) with efficient resource allocation, checkpointing fault recovery mechanism is applied.	• Time to Re-boot VM slow down the overall performance. • Computational time of VM varies. • Fault Prediction mechanism not implemented.
[20]	Light weight multi- threaded service named FTB-IPMI (Fault Tolerance Backplane- Intelligent platform management interface), Rule based fault prediction engine.	• False predictions are generated by portable rule based fault prediction engine. • Efficiency and reliability of the resources decreased.

[21]	Distributed application oriented multi- level checkpoint/ restart fault recovery approach for IaaS model of Cloud.	• Fault prediction, a pre-condition of fault tolerance is not implemented. • Unable to recover the VM failure completely because VM running state is not saved. • Waiting time is increases. • Checkpoint/ Restart increase the size of the backup files. • A host scheduling and file transferring increase the extra overhead time and cost.
[22]	Structure state process to characterize the failure behavior of VM in cloud, and a service oriented dependability metric i.e. Defects per million with three types of replication (Cold, warm and Hot) technique is implemented.	• Inadequacy of the resources and limitations of the buffer space deteriorates the performance of overall architecture.
[23].	Failure drill an instance of online testing is applied to analyze the overall behavior of VM and other web based components.	• Extra migration cost which in turn consumes more network bandwidth and storage space. • There is a risk of safety. • Slow down the VM performance and failed to recover the VM from failure.
[24]	A run time performance anomaly fault localization tool and online system call trace analysis algorithm is applied.	• Covers only two fault features are analyzed such as, fault impact scope and fault outset time dispersion. • Unable to handle the VM and task failure in large scale environment.
[25]	Self- deploy decentralized protocol with message gossip mechanism (A fault detection technique.)	• Data is lost when VM fails. • Failed to add/ remove the VM dynamically. • Replication is not implemented.
[26]	Proactive fault tolerance approach.	• Heavily relies on the administrator system. • Failure of administrator leads to overall performance degradation. • Unable to handle VM faults and network failures in scale manner.
[27]	Cost- based fault tolerance and DoomDB, an ego shooter game is proposed.	NA
[28]	Autonomic fault tolerant scheduling, hybrid heuristic approach and proactive fault tolerance technique is applied.	• Unable to predict VM faults for balanced scientific workflow. • Computational time increases.
[29]	A log based recovery based on Reed Solomon.	NA
[30]	RabbitMQ, an Advanced Messaging Queue Protocol (AMQP) with reactive and proactive fault tolerance approach.	• Unable to predict the network failure, process failures. • Performance overhead worsens the performance of the overall proposed approaches.
[31]	Algorithm of byzantine failure tolerance with erasure coding, DepSky system architecture is applied.	• VM failure and network congestion problem arises. • Failure recovery consumes extra computation time. • Replication of VMs task increases the performance overhead.
[32]	Dedicated checkpointing for application- level and process level.	• Cost of migration from failed VM to healthy VM increased. • There is need of predicting the VM failure at an early stage.
[33]	Virtualization with reactive fault tolerance technique and checkpointing fault recovery.	• Overall wall clock execution time is increased. • Only theoretical explanation of the proposed algorithm is discussed.
[34]	Proactive fault tolerance and preemptive migration with feed- back loop control mechanism.	• Not capable of covering all types of failures. • Double bit- flips in ECC memory, a software bug, not handled by it. • Scalability issues due to undefined interfaces and standardized metrics. • Absence of fault prediction and fault removal, reliability and availability decreased.
[35]	Leader- Follower based consensus algorithm with decentralized communication pattern.	• Slowdown the performance when process crashes occurred.
[36]	Self- healing policy with live migration of VM	• Monetary cost increase by the redundancies. • Handling the original data with its redundant copies is unmanageable when a failure exists.

[37]	System- level and modular perspective fault tolerance approach.	• NA
[38]	Fault tolerant sandbox with multiple error detection and mechanism.	• Fault tolerance and scalability is limited due to unavailability of the run time monitoring and runtime information analyses of VM components.
[39]	Decentralized parallel dual- assembly pipeline (DAP) with VM migration policy.	• Monetary cost varied while recovering the lost message. • Management of entire architecture is complex.
[40]	Fault tolerance model with forward recovery mechanism.	• Cost of renting the resources increased. • No fault prediction and fault detection mechanism is implemented to enhance the reliability and availability of the resources. • Overall computation time i.e. wall clock time is increased.
[41]	Delay- tolerant fault-tolerant algorithm with Dual Direction File Transfer Protocol (DDFTP).	• Not implemented under shared heterogeneous cloud environment. • Performance of proposed algorithm will more effective when fault prediction and detection is applied.
[42]	Exclusive and collaborative fault tolerant with Checkpointing fault recovery approach.	• Computation time is increased with the inclusion of the checkpointing. • Worst performance with sensitive applications such as MPI applications.
[43]	Stepwise refinement approach with static reconfiguration.	• Redeployment and restoration of applications on virtual machines can also have a significant cost.
[44]	Imitator, a new fault tolerance mechanism with VM replications technique.	• Imitator incurs small normal execution overhead, and provides fast crash recovery from failures.
[45]	Architecture with different operating system combined for Real Time Application of VMs.	• Need to develop a predictive model of the virtual machines execution time when deploying specific OS using approximated provisioned resources.

III. CONCLUSION, OPEN RESEARCH ISSUES, AND FUTURE DIRECTIONS

In the last few decades, computation power of IaaS model has been increased in massive quantities by various computing devices such as VMs, Load- Balancers, Hypervisor, etc.,. Continuous increases in the computation power and advancements in recent technologies are the main reasons behind availability and reliability of the resources [1][2][3][4][5][6].In this paper, different failures in IaaS along with various fault tolerance strategies that are widely used to enhance the reliability of the resources. A detailed comprehensive review on fault tolerance technique along with the (a) advantages, (b) disadvantages, (c) performance is explored in this paper. The knowledge provided in this paper can be further deed to design and model new mechanism or approaches in the cloud.

3.1 Managing overall computational time using fault tolerance:

Maintaining the availability and reliability of the IaaS cloud resources is a very complicated task as compared to other computing system such as distributed computing systems, grid computing systems etc. while maintain the reliability while balancing the cost of migration of VMs from faulty nodes to faulty nodes and replication of task are the another major concern.Several replication techniques have been proposed to address this issue.[39][40][41].For example, in [1], Xen-Hypervisor Virtualization technology with Open Nebula management tool, Dynamic Resource allocation algorithm with heart beat messaging protocol performs worst and affects the overall performance of the entire architecture and unable to guarantee the availability and integrity of the stored data.Several tools and techniques have been developed to increase the performance of the resources in IaaS such as Autonomic fault tolerant scheduling, hybrid heuristic approach and proactive fault tolerance [10], but increase in computation time, developed are moving from this widely used technique. However, overall performance degradation due to the checkpointing, replication [11] [12] [13] are the major concerns in the cloud computing environment.

Other issues related to the use of various well defined fault tolerance approaches in the IaaS cloud model of computing to support reliability and high performance.

a) There exists a lack of reliable fault prediction mechanism to predict the process failure, VM failures, degree of replication after the occurrence of failure.
b) Most of the fault tolerance approaches works statistically in practical scenario but performance get biased and decreased due to its statistical nature.
c) Current fault tolerance strategies and optimizations are targeted only at a few faults in IaaS.

We have highlighted some of the abovementioned research issues involved in the IaaS model of cloud computing. Future directions may involve striving for the solutions for the above issues. For example, one way to deal with the reliability issue is to develop performance model with measurement tools. However, model for reliability based fault tolerance in IaaS cloud have not been extensively studied. Therefore, we have only outlined some of the technical problems that await the solution and new may also arise as a result of the technological changes.

References

[1] Yang, C. T., Liu, J. C., Hsu, C. H., & Chou, W. L. (2014). On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *The Journal of Supercomputing*, *69*(3), 1103-1122.

[2] Wu, Y., Song, H., Xiong, Y., Zheng, Z., Zhang, Y., & Huang, G. (2014, November). Model defined fault tolerance in cloud. In *Proceedings of the 6th Asia-Pacific Symposium on Internetware on Internetware* (pp. 116-119). ACM.

[3] Qiu, W., Zheng, Z., Wang, X., Yang, X., & Lyu, M. R. (2013). Reliability-based design optimization for cloud migration. *IEEE Transactions on Services Computing*, *7*(2), 223-236.

[4] Chen, G., Jin, H., Zou, D., Zhou, B. B., Qiang, W., & Hu, G. (2010, September). Shelp: Automatic self-healing for multiple application instances in a virtual machine environment. In *2010 IEEE International Conference on Cluster Computing*(pp. 97-106). IEEE.

[5] Mondal, S. K., Muppala, J. K., Machida, F., & Trivedi, K. S. (2014, November). Computing defects per million in cloud caused by virtual machine failures with replication. In *2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing* (pp. 161-168). IEEE.

[6] Dean, D. J., Nguyen, H., Wang, P., & Gu, X. (2014). PerfCompass: Toward runtime performance anomaly fault localization for infrastructure-as-a-service clouds. In *6th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 14)*.

[7] Etchevers, X., Salaün, G., Boyer, F., Coupaye, T., & De Palma, N. (2014, March). Reliable self-deployment of cloud applications. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (pp. 1331-1338). ACM

[8]

[9] Binnig, C., Salama, A., & Zamanian, E. (2014, June). DoomDB: kill the query. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (pp. 913-916). ACM.

[10] Bala, A., & Chana, I. (2015). Autonomic fault tolerant scheduling approach for scientific workflows in Cloud computing. *Concurrent Engineering*, *23*(1), 27-39.

[11] Ahmed, M., Vu, Q. H., Asal, R., Al Muhairi, H., & Yeun, C. Y. (2014). Lightweight secure storage model with fault-tolerance in cloud environment. *Electronic Commerce Research*, *14*(3), 271-291.

[12] TaheriMonfared, A., & Jaatun, M. G. (2012). Handling compromised components in an IaaS cloud installation. *Journal of Cloud Computing: Advances, Systems and Applications*, *1*(1), 16.

[13] Poola, D., Garg, S. K., Buyya, R., Yang, Y., & Ramamohanarao, K. (2014, May). Robust scheduling of scientific workflows with deadline and budget constraints in clouds. In *2014 IEEE 28th international conference on advanced information networking and applications* (pp. 858-865). IEEE

[14] Song, B., Ren, C., Li, X., & Ding, L. (2014, September). An efficient intermediate data fault-tolerance approach in the cloud. In *2014 11th Web Information System and Application Conference* (pp. 203-206). IEEE.

[15] Snyder, B., Ringenberg, J., Green, R., Devabhaktuni, V., & Alam, M. (2015). Evaluation and design of highly reliable and highly utilized cloud computing systems. *Journal of Cloud Computing*, *4*(1), 11.

[16] Arantes, L., Friedman, R., Marin, O., & Sens, P. (2015, September). Probabilistic byzantine tolerance for cloud computing. In *2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS)* (pp. 1-10). IEEE.

[17] Wu, Y., Song, H., Xiong, Y., Zheng, Z., Zhang, Y., & Huang, G. (2014, November). Model defined fault tolerance in cloud. In *Proceedings of the 6th Asia-Pacific Symposium on Internetware on Internetware* (pp. 116-119). ACM.

[18] Yang, C. T., Liu, J. C., Hsu, C. H., & Chou, W. L. (2014). On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *The Journal of Supercomputing*, *69*(3), 1103-1122.

[19] Feller, E., Rilling, L., & Morin, C. (2012, May). Snooze: A scalable and autonomic virtual machine management framework for private clouds. In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)* (pp. 482-489). IEEE.

[20] Kumar, P., Raj, G., & Rai, A. K. (2014, September). 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)* (pp. 138-143). IEEE

[21] Chen, G., Jin, H., Zou, D., Zhou, B. B., & Qiang, W. (2015). A lightweight software fault-tolerance system in the cloud environment. *Concurrency and Computation: Practice and Experience*, *27*(12), 2982-2998.

[22] Mondal, S. K., Muppala, J. K., Machida, F., & Trivedi, K. S. (2014, November). Computing defects per million in cloud caused by virtual machine failures with replication. In *2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing* (pp. 161-168). IEEE.

[23] Ganesan, A., Alagappan, R., Arpaci-Dusseau, A. C., & Arpaci-Dusseau, R. H. (2017). Redundancy does not imply fault tolerance: Analysis of distributed storage reactions to single errors and corruptions. In *15th {USENIX} Conference on File and Storage Technologies ({FAST} 17)* (pp. 149-166)

[24] Dean, D. J., Nguyen, H., Wang, P., & Gu, X. (2014). PerfCompass: Toward runtime performance anomaly fault localization for infrastructure-as-a-service clouds. In *6th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 14)*.

[25] Etchevers, X., Salaün, G., Boyer, F., Coupaye, T., & De Palma, N. (2014, March). Reliable self-deployment of cloud applications. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (pp. 1331-1338). ACM.

[26] Egwutuoha, I. P., Chen, S., Levy, D., Selic, B., & Calvo, R. (2012, November). A proactive fault tolerance approach to High Performance Computing (HPC) in the cloud. In *2012 Second International Conference on Cloud and Green Computing* (pp. 268-273). IEEE.

[27] Binnig, C., Salama, A., & Zamanian, E. (2014, June). DoomDB: kill the query. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*(pp. 913-916). ACM.

[28] Bala, A., & Chana, I. (2015). Autonomic fault tolerant scheduling approach for scientific workflows in Cloud computing. *Concurrent Engineering*, *23*(1), 27-39.

[29] Ahmed, M., Vu, Q. H., Asal, R., Al Muhairi, H., & Yeun, C. Y. (2014). Lightweight secure storage model with fault-tolerance in cloud environment. *Electronic Commerce Research*, *14*(3), 271-291.

[30] TaheriMonfared, A., & Jaatun, M. G. (2012). Handling compromised components in an IaaS cloud installation. *Journal of Cloud Computing: Advances, Systems and Applications*, *1*(1), 16.

[31] Tebaa, Maha, and Said EL Hajji. "From Single to Multi-clouds Computing Privacy and Fault Tolerance." *IERI Procedia* (2014): 112-118.

[32] Nicolae, Bogdan, and Franck Cappello. "BlobCR: Virtual disk based checkpoint restart for HPC applications on IaaS clouds." *Journal of Parallel and Distributed Computing* 73.5 (2013): 698-711

[33] Das, Pritam, and Pabitra Mohan Khilar. "VFT: A virtualization and fault tolerance approach for cloud computing." *Information & Communication Technologies (ICT), 2013 IEEE Conference on*. IEEE, 2013.

[34] Engelmann, Christian, et al. "Proactive fault tolerance using preemptive migration." *Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on*. IEEE, 2009.

[35] Hanna, Fouad, Jean-Christophe Lapayre, and Lionel Droz-Bartholet. "Fault tolerance management in distributed systems: A new leader-based consensus algorithm." *High Performance Computing & Simulation (HPCS), 2014 International Conference on*. IEEE, 2014.

[36] Hasan, Tanim, Asif Imran, and Kazi Sakib. "A case-based framework for self healingparalysed components in Distributed Software applications." *Software, Knowledge, Information Management and Applications (SKIMA), 2014 8th International Conference on*. IEEE, 2014.

[37] Jhawar, Ravi, Vincenzo Piuri, and Marco Santambrogio. "Fault tolerance management in cloud computing: A system-level perspective." *Systems Journal, IEEE* 7.2 (2013): 288-297.

[38] Liu, Lei, et al. "Data Redundancy Replication Strategy Based on Relevant Failure in Cloud Environment." *Advanced Materials Research*. Vol. 989. 2014.

[39] Li, Junguo, et al. "Supporting Reconfigurable Fault Tolerance on Application Servers." *Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on*. IEEE, 2009.

[40] Malik, Sheheryar, and Fabrice Huet. "Adaptive fault tolerance in real time cloud computing." *Services (SERVICES), 2011 IEEE World Congress on*. IEEE, 2011.

[41] Nuaimi, Klaithem Al, et al. "A survey of load balancing in cloud computing: challenges and algorithms." *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*. IEEE, 2012.

[42] Tchana, Alain, Laurent Broto, and Daniel Hagimont. "Approaches to cloud computing fault tolerance." *Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on*. IEEE, 2012.

[43] Troubitsyna, Elena. "Developing fault tolerant distributed systems by refinement." *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*. IEEE, 2010.

[44] Wang, Peng, et al. "Replication-based fault-tolerance for large-scale graph processing." *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*. IEEE, 2014.

[45] Garraghan, Peter, Paul Townend, and Jie Xu. "Real-time fault-tolerance in federated cloud environments." *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on*. IEEE, 2012.