International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 10 (2019) pp. 2303-2309

An Exploration of Failure Prediction and Failure Detection in a Cloud Based Environment

Kranthi Kumar. K, K.S. Sai Sushmitha and R. Rindha Reddy

Information Technology
Sreenidhi Institute of Science and Technology.
Yamnampet, Ghatkesar, Telangana 501301, India.

Abstract

In this paper, we investigated the various methods towards Failure Detection (FD) and Failure Prediction (FP) in a cloud-based environment. A failure in cloud is said to take place when there is a difference between the service provided to the customers and the service that was initially intended to be provided. In other words, the services by the system do not satisfy the constraints of the customer. In today's world, cloud is used extensively and failures like this must be avoided. Basically, in cloud environments there are three different types of failures, Virtual Machine (VM), software and hardware. For preventing failures, it is important to accurately predict or detect them in cloud and then adopt a suitable strategy to get rid of it. FD is a method that is used to identify the exact location of an already present failure in the system before it can cause any damage. FP is the process of accurately auguring whether a failure will occur or not at a particular location. We came across that by using approaches like Bayesian Interface, Recurrent Neural Network, and Big Data model one can effectively predict failures in cloud environment. And by using methods like Volterra series, deep learning one can detect failures. While predicting or detecting failures a lot of parameters must be put in consideration like-resource efficiency, resiliency and service recovery performance. All the papers have been evaluated carefully and have been presented in a useful manner.

Keywords: FD, FP, VM, resource efficiency, service recovery performance.

INTRODUCTION

The cloud is a term which means using computer, information technology (IT) and programming applications through a system with the help of data centres (alludes to a server equipment on the location of the client to store and access information through local network) using wide area networking (WAN) or Internet connectivity. A cloud acts like a literal cloud. A cloud normally evaporates water from a number of water-bodies and then goes to various areas and precipitates the condensed water there. Similarly, a cloud in technical terms collects different types of resources from different sources and produces a mixture of heterogeneous resources. Then moves to those places where the user is located and allows them to utilize the resources. Cloud computing is a kind of computing which is utilized for supervising applications by depending upon the shared resources of computing instead of personal devices or servers present in the locality. In its most basic portrayal, cloud computing is transporting the services ("cloud services") which it has, outside the firewall of an organization. A cloud based environment typically means an organization which makes use of various applications, resources and services provided by cloud computing provider's servers. In spite, of the presence of fault, the Cloud Service Providers (CSP) has to provide the demanded resources to their customers because cloud is on demand request and they must fulfill their service level agreement. The failure rate is very high in cloud because there occurs a lot of memory leaks during software updates. The general methods used for failure prediction are by the combination of machine learning strategies and equipment degradation physical models. Any technique must keep in mind the client requirements and must not compromise on the quality of services guaranteed. This paper is organized as section 2, introduction to FD and various FD approaches, section 3 contains introduction of FP and the most widely used FP methods. Then in section 4 a comparative study is done of the approaches in FD and then later FP. This section is followed by section 5 which is conclusion.

RELATED WORK

Failure Detection approaches

In this section we discussed about FD Module, resiliency into cloud systems, Adaptive Anomaly Detection, Perception Based Anomaly Detection, Self-tuning Failure Detection scheme, Thermal Anomaly-aware Resource Allocation model, Byzantine fault detection, Self-Evolving Anomaly Detection. In [1], the FD Module is an essential utility of the fault tolerant framework. This examination proposes a versatile fault identification component which utilizes the Volterra series for foreseeing the entry heartbeat messages and for decision making, it uses the decision tree. Present day complex and massive cloud computing [2] frameworks, are of no help whenever hardware and programming mistakes or human errors occur, this essentially impacts the reliability and execution of cloud. AFD is Adaptive Failure Detection Framework. It is not same as other FD approaches and doesn’t require an earlier history of failure and can self-adjust

itself by gaining knowledge from perceiving failures which occur during runtime. Keeping in view, the cloud execution information, AFD identifies conceivable failures that are checked by the operators of cloud. The failures are affirmed as either ordinary states or true failures. AFD takes advantage of those kinds of failures which were perceived but not detected and were revealed by the administrators of cloud in order to recognize unknown failure kinds.

Villarreal-Vasquez et.al [3] proposed an unconventional way to establish cloud systems with resiliency to an extent that they can alleviate failures to deliver continuous working of basic responsibilities. Here the arrangement depends on dispersed checking of cloud service/VM conduct and regular invigorating of the relatable cloud assets to permit self-versatile SDN (Software Defined Network) reconfiguration with an MTD (Moving Target Defence) method. It was exhibited that, with initial examinations the MTD based arrangement can accomplish worthy reconfiguration times. In this paper [4], it is introduced that AAD (Adaptive Anomaly Detection) does not require an earlier record of failure occurrences. To upgrade the exactness of anomaly recognition considerably further, responsive methodologies, for example, check pointing and repetitive execution, ought to be incorporated to deal with failures along with proactive and reactive failure handling strategy. In [5], the issue which is dealt here is, figuring out, the fault which is prevailing amid the failed nodes. Towards the end, an approach is suggested to surpass the issue of faults because of the movement of the nodes. The fault is taken care of based on the weighted methodology. The proposed system results in less clogged and fault free cloud systems whereas the already existing algorithm gives just numerous congestion freeways yet without fault toleration.

Uchiumi et.al [6] proposed that by discovering greater patterns utilizing decision tree analysis, misconfigurations can be detected. A pattern alteration technique is added to deal with the situation where the majority cannot be decided as the servers are categorized into groups of similar sizes. By utilizing the example adjustment, higher exactness of misconfiguration occurrences are acquired. When the misconfiguration detection was evaluated it performed and recognized that most (78.6% in real information) of the identified candidates are misconfigurations in reality. An exceptional anomaly detection framework [7] dependent on client observation as opposed to complex framework models is introduced here. Metrics-based framework models are very intricate and their abnormality identification components are for the most part, dependent on threshold scheme. PBAD (Perception-Based Anomaly Detection) conceals the intricacy of Cloud and focuses on what is really important, and how the service provided by Cloud applications is seen by the clients. In a production datacentre environment, when PBAD is executed and deployed, it is demonstrated that PBAD can precisely recognize anomalous parts through substantial tests. The results demonstrate that PBAD recognizes various sorts of anomalies and the combination of anomalies where existing frameworks are unsuccessful.

In [8], this paper investigates the properties of FD in connection to the real and programmed failure tolerant networks of distributed computing, and finds a non-manual basic examination strategy to self-tune the corresponding parameters to fulfill clients' necessities. In light of this general programmed strategy, a dynamic and particular Failure Detector scheme with self-tuning properties is proposed called SFD, it is a noteworthy leap forward, compared to the present strategies. Real and broad examinations were completed to measure the differences between the standard of service provided by SFD and a few other already present FDs. The results exhibit that this scheme can naturally modify SFD control parameters to get related services and fulfill client necessities, at the same time keeping up great execution. An adaptive anomaly recognition structure [9] in cloud computing frameworks has been proposed. First analyses is done of the correlation present between the primary components in which failure detections occur, where it is found that PCs maintaining the highest variance cannot properly conduct the characterization of the failure events, on the other hand, PCs having lower order, display high correlation with failure occurrences. Then the benefits from the Most Relevant Principal Components (MRPCs) are drawn so as to depict failure occasions and formulate a way on learning-based to deal with recognition of cloud peculiarities by utilizing MRPCs. The anomaly detector learns from recently confirmed identification results and uses undetected failure history to discover new ones.

Loet.al [10] proposed an agreeable intrusion recognition framework for cloud computing network to decrease the effect of DoS (Denial-of-benefit) assault. By doing this, if a DoS attack is done on one of the cloud computing regions, then cooperative IDS (Intrusion Detection Scheme) sends alert message to different IDS frameworks. IDS could accumulate a similar kind of attack sent from different IDSs. At that point it makes a judgment to decide the reliability of this alarm message by majority vote technique. Therefore, the proposed framework keeps the IDS framework from single purpose of failure. By agreeable operation among these operators, early location and aversion strategy is actualized. Subsequently, IDSs are deployed in distributed computing localities; keeping aside the victim one could stop these sorts of attacks. In this paper [11], a methodology that aids the solid affirmation of right sporadic cloud execution activities as a regular practice done in DevOps, particularly the arranged redesign of running VMs is introduced. Centre to this methodology is a correlation (regression-based) examination system that finds out the relationship between the event logs of activities and cloud asset changes. It is demonstrated that the determined regression model can also be utilized for producing runtime declarations so as to recognize abnormalities in running tasks. This methodology was assessed on the Amazon public cloud computing service (EC2) where, various activities run and arbitrary flaws were infused. The outcomes exhibit that this regression-based analysis method can distinguish infused anomalies with high exactness and review.

In [12], the developing significance, vast scale, and high number of servers with superior processing datacenters makes the servers vulnerable to vital attacks, failures (consisting of both computing under structure and cooling) and

misconfiguration. Such sudden incidents lead to many thermal peculiarities –fugues, hotspots and cold spots – which significantly affect the expense of activity of the datacenters. Such problems when not detected can cost a huge business, a loss of millions of dollars. The model-based thermal fault discovery solution along with a novel Thermal Anomaly-aware Resource Allocation (TARA) fundamentally enhances the recognition chances when contrasted with failure detection using conventional scheduling algorithms, for example, round robin, random and best-fit. This is on the grounds that TARA makes time-fluctuating thermal fingerprints of the datacentre, so as to expand the recognition exactness and limit its dormancy.

Rukavitsyn et.al [13] proposed a technique for adjustable learning of identifying model set on a foundation of data mining algorithms. The fundamental purposes behind alteration of the network traffic which impact the rise of false positive errors of the classifier are considered in this paper: the variation in the quantity of customers, services as well as number of requests. The relearning algorithm based on the calculation of threshold estimations of attributes of the traffic is formed. The testing technique of these threshold esteems for beginning of a relearning algorithm is created and exhibited. The utilization of relearning algorithms builds the resiliency of identifying a model to change the quantitative traffic of 547 attributes and permits bringing down of false positive errors.

Anomaly discovery and proactive failure administration [14] gives a vehicle to self-overseeing cloud assets and improving framework reliability. In this research, a wavelet-based multi-scale cloud abnormality detection technique with a learning-aid mother wavelet determination and gliding recognition windows for adjustive fault identification has been utilized. It dissects both domains of frequency and time to recognize unknown cloud practices and adjust to the failures happening even at run time. A model for cloud inconsistency identification structure on a cloud computing framework has been executed. Exploratory outcomes demonstrate that this methodology can recognize cloud failures with the most elevated precision among a few broadly utilized methodologies.

A model based on the Byzantine fault detection [15] technique for cloud computing has been proposed. Here the procedure comprises of a domain of problem demonstrations, framework integration and examination process. The procedure for fault recognition in cloud computing is that the Petri nets are utilized to develop diverse parts of cloud computing, the coordination system effectively incorporates these compositions into a Computing Fault Net (CFN). Based on this, essential properties of this model are understood, the related hypotheses and operational semantics of Petri nets support in demonstrating the viability and rightness of the set of rules that were followed step by step.

SEAD [16], a Self-Evolving Anomaly Detection structure along with mechanisms is presented in this paper. SEAD does not require an earlier record of failure, in this way, being able enough, for discovering failures not yet found previously. It can self-advance by gaining knowledge from recently created confirmed recognition results. The suggested structure along with various components in this paper can likewise help in failure forecast. The existing predictive techniques, for example, [17], [18], [19], [20], [21] were utilized for execution and failure logs in order to foresee when the fundamental framework will encounter an interpretative event. The outcomes from this paper can be used in deciding all the potential restrictions of the issue by studying the cloud execution data at runtime. Support Vector Machine (SVM) and one-class SVM for self-evolving anomaly detection were used in this paper.

Failure Prediction approaches

In this section we discussed about challenges faced in applying existing FP methods, integration of FP into the orchestration of current 5G systems, leveraging predictive maintenance, a prediction model which is based on the Long Short-Term Memory Network (LSTM), FailureSim, MPI-based application failure avoidance, software rejuvenation-based fault tolerance scheme.

Watanabe et.al [22], depicted the difficulties in applying the existing failure prediction techniques into the administration of cloud datacenters in which failure existence rates are moderately less. A technique to restrain 'unsure' predictions which are dependent on the normalcy of message patterns is proposed. In this methodology, the chances of failure and the synchronous occurrence of the message pattern are figured out. At that point, additionally the computation of the message pattern typicality present in the failure occurs and is assessed if the identified message pattern ought to be accounted for as an indication of failure. In a commercial cloud data server, the assessment of actual management condition, uncovered that this technique can enhance the complete performance of the system.

In [23], the initial move towards a superior comprehension of the advantages of incorporating FP into the orchestration of current 5G frameworks is introduced. Explicitly, it researched how a failure forecast module for the optical connections can be coordinated into the recuperation procedure of the optical cloud administrations. More particularly, it explored in what way a FP module for the optical links can be used as an asset for effective recuperation of several optical cloud services. Then, in terms of cloud service accessibility and asset effectiveness the advantages were evaluated. In view of the above goal, PPR (Proactive Path Restoration) + SR (Service Relocation) system was proposed, which can proactively migrate those cloud administrations navigating links that are most likely predicted to fail. In this paper [24], it is exhibited how publicly open and accessible data can be very important irrespective of its data size, despite the fact that huge amounts of data would most probably have permitted a lot more of system structure awareness into the existing data. In an environment which is based on distributed computing, a method to deal with failure prediction to expand framework accessibility, utilizing Liner Regression (LG) and SVM was introduced.

A compelling characterization model [25] is just the initial move towards utilizing predictive perpetuation. This paper, effectively consolidated remaining workload information (execution counters) with equipment information (SMART (Specific, Measurable, Attainable, Realistic, Timely)) for this model. In public cloud, it is not possible to own the workload so it cannot get information about health and performance. For this situation, information regarding the needs of data was plumbed through numerous layers of hardware foundation. The underlying outcomes from the model and early generation conditions couldn't be repeatedly produced at large scale because of a mix of fragmented information and equipment disparity. But under the states of timely and complete information, the methodology and results were energized, given its capacity to decrease benefit disturbances because of issues regarding faults in disk and upgrade operational expenses for fixing hardware equipment.

Islam et.al [26] investigated the Google release of scheduler demand and used information over an expansive (12.5K+) and broad computer cluster in light of the fact that with the development of heterogeneous, huge and shared clusters of computing, their effective use by the combined dispersed workloads and tenants remains a vital test. It was seen that unsuccessful and completed jobs and tasks have distinctive attributes of asset utilization, and that these distinctions have a much higher likelihood of showing way before the time the jobs' end. This paper focuses on the significance of failure prediction for asset planning and provisioning in computer clouds. A model for prediction was available which expands on an exceptional kind of Recurrent Neural Network (RNN) called LSTM and strategic relapse, for foreseeing failures by means of different properties and execution time arrangement information. It effectively anticipated the end statuses of not only jobs but also tasks in the Google cluster followed by decent exactness, accuracy, and review. FailureSim [27], a simulator for data centers of cloud is utilized for prediction of failure. The framework enables clients to allocate practices to hosts in any cloud data center. It characterizes many failing and working host practices. This simulator incorporates an arrangement framework that enables clients to foresee the signs of failure in host behaviour. Utilizing the most exact model, grouping framework could accurately foresee half of the coming up failures, while effectively figuring out how to anticipate 89% of every single falling situation before it happens.

Liu et.al [28] proposed a continuous real-time critical event pattern acknowledgment along with a prediction system to anticipate the critical events in a virtualized cloud computing condition. Here the system joins highlights of hash table, linked list and reversal pattern tree structure to build the prediction as well as the recognition speed for continuous real time critical event pattern prediction and identification. The reversal pattern tree gives all the relations between child nodes and the relation between parent nodes and child nodes. Then this connection is utilized in a prediction to expand the precision. This prediction procedure additionally uses temporal logic to ascertain the consistency of the relationship to alter the expectation result. The proposed system likewise lessens the space intricacy of real-time critical event pattern prediction and identification. In this paper [29], a model framework is shown that uses some constructs of cloud (IaaS and virtualization) related to prediction of failure to encourage MPI (Message Passing Interface) application failure-based evasion in such a way that is unambiguous to the application. In particular, the framework performs a prediction-driven live transport of the MPI processes which are running in virtual machines to the resources that were acquired from the administrator of resources. The verification of idea, utilized copying of the genuine real-time failure situation found in many production frameworks are done so as to outline the utility of such a framework for keeping failures away.

In [30], Scheduling of tasks is a vital issue which significantly impacts the cloud computing framework's execution. In this research, the task failures found in Google clusters data were inspected and it was discovered that 40% of the tasks and 42% of the jobs we're not completed effectively. The likelihood of foreseeing the scheduling result of a task was examined utilizing the historical data and models of statistics regarding the execution of those tasks which were scheduled recently and it was discovered that Random Forest models can accomplish a review up to 96.2% and an exactness of up to 97.4%. The schedulers executed in Hadoop and GloudSim were extended out to consolidate the predictions in failures of task. The difference between the new scheduler's scheduling performance and the first scheduler executed in GloudSim is that the, the entire number of finished tasks can be expanded by 40 % (individually 20 %) and the new scheduler can diminish the execution time. In Hadoop, the new scheduler can decrease the quantity of job failures precisely by 70% with less than 5 minutes of overhead time.

So as to neutralize software maturing hinders for administration parts in cloud applications [31], an unconventional and encompassing software rejuvenation-based failure tolerance scheme is introduced in order to enhance their running accessibility. Through a fundamental assessment, the versatile failure recognition and maturing degree assessment approach can precisely foresee which cloud service segments should be first rejuvenated, and then the checkpoint along with a trace log replay-based restoration approach is an effective and hopeful adaptation to internal failure avoidance system to ensure consistent running of cloud applications.

COMPARITIVE STUDIES

Failure Detection

This section deals with the comparisons of failure detection models.

AFD VS AAD1 (Adaptive Anomaly Identification by Exploring Metric Subspace in Cloud Computing Infrastructures)

Detection sensitivity (as in Eq. (1)), measures the extent of real positives that are effectively recognized in that capacity.

It plays a key role in drawing a comparison between several FD models.

$$\text{Detection sensitivity} = \text{ndf/tnf} \quad (1)$$

where ndf is the number of detected failures and tnf is total number of failures.

Detection specificity (as in Eq. (2)), measures the extent of real negatives that are accurately distinguished from all things considered. It is very helpful in determining how reliable the FD module is.

$$\text{Detection specificity} = \text{nds} / \text{tns} \quad (2)$$

where nds is the number of detected normal states and tns is total number of normal states.

The comparison between the performance evaluation of AFD and AAD1 is conducted. The AFD detector takes the cloud execution data and tries to discover all the possible failures. It modifies itself by learning from the checked identification results and failures which are identified but unfortunately not detected as reported by the cloud administrators. The AAD1 is MRPC based. It consists of a collection of PCs that are correlated strongly along with failure occurrences. So for each type of failure that occurs, a MRPC set is selected. This kind of MRPC based anomaly detection can accurately detect failures and at same time achieve low overhead. Using the reference from Eq. (1) and Eq. (2) and as in [2], AFD achieves 92.1% of detection sensitivity and 83.8% of detection specificity. On the other hand, as in [9], AAD1 which is an MRPC – based anomaly detector achieves detection sensitivity of 91.4% and detection specificity up to 3.7%. AAD1 gives more accurate results than AFD. AFD takes usually about 7.26 seconds for a particular control node which is present in the cloud so as to extract cloud metrics of cloud execution, hypersphere creation, and make detections of failures. The adjustment steps are substantially more lightweight, taking 2.17 seconds to detect failures and keep the hypersphere updated. AAD1 takes 6.81 seconds normally for a control node in the cloud to process cloud execution information, select MRPCs, and make detections of anomaly.

AD1 (Adaptive Anomaly Identification by Exploring Metric Subspace in Cloud Computing Infrastructures.) vs AAD2 (Adaptive Anomaly Detection System for Cloud Computing Infrastructures.)

AAD2 is relatively lightweight and it takes hardly few seconds of time to start the detector and few more seconds for FD and self-adaptation. A total of 518 metrics is profiled every minute. AAD2 covers the statistics of each and every part of a cloud server, including paging and page faults, CPU usage, task switching activity, process creation, swap and memory space usage, interrupts, network activity, data and I/O transfer, management of power, and many more. The AAD1 is MRPC based. It consists of a collection of PCs, which are correlated strongly along with failure occurrences. So for each type of failure that occurs, a MRPC set is selected. This kind of MRPC based anomaly detection can accurately detect failures and at same time achieve low overhead.

Using the reference from Eq. (1) and Eq. (2) and as in [5], AAD2 failure detector detects 92.1% of detection sensitivity and 83.8% of detection specificity. On the other hand, as in [9], AAD1 which is an MRPC – based anomaly detector achieves detection sensitivity of 91.4% and detection specificity up to 3.7%. AAD1 gives more accurate results than ADD2. ADD2 takes usually about 7.26 seconds for a particular control node which is present in the cloud so as to extract cloud metrics of cloud execution, hypersphere creation, and make detections of failures. The adjustment steps are substantially more lightweight, taking 2.17 seconds to detect failures and keep the hypersphere updated. AAD1 takes 6.81 seconds normally for a control node in the cloud to process cloud execution information, select MRPCs, and make detections of anomaly.

Failure Prediction

This section deals with the comparisons of failure prediction models.

FailureSim: A System for Predicting Hardware Failures in Cloud Data Centers Using Neural Networks (FS) Vs Predicting Application Failure in Cloud: A Machine Learning Approach. (PAF)

The FS basically classifies all the hosts into working or failing host behaviours. Whereas, the PAF predicts the failures at the task level and the job level. At the task level, the termination statuses of task submissions are classified into 3 classes- kill, evict and fail. At the job level, the termination statuses of job submissions are classified into 2 classes- failed and finished.

As in [27], the FS, information and algorithm for classification, classifies failure accurately half of the time, and additionally with the expulsion of BW (Business Warehousing), I/O (Input/Output), and other unique cases, this value increments up to generally 58%. Moreover, the FS characterization data and algorithm, forecasts the failure of a host with a precision of 89%, which increments to 98% with the expulsion of anomalies.

As in [26], in PAS, at the task level classification, 87% of exactness, is achieved. And at the job level, a precision of 81% is accomplished. So within the PAS, the task level classification proves to be more efficient than the job level classification. Therefore, on an average, the PAS has achieved 84% of exactness. But when FS and PAS are compared it is clearly evident that FS predict failures more accurately with greater exactness. Its precision is 5% more than PAS.

CONCLUSION

In this survey paper we give detailed information about the various approaches towards FD and FP. In order to improve the satisfaction of cloud customers it is very essential to make the delivered services fault free. And by regularly checking the faults in cloud, the unnecessary losses incurred can be prevented. With the advancements made in FD and FP modules a significant impact can be made on failure management also in cloud. Various cloud systems can adopt a FD or FP module which is suitable to their respective cloud environment. For preventing failures, it is important to accurately predict or detect them in cloud and then adopt a suitable strategy to get rid of it. FD is a method that is used to identify the exact location of an already present failure in the system before it can cause any damage. AFD achieved 92.1% of detection sensitivity and 83.8% of detection specificity. AAD2 failure detector detects 92.1% detection sensitivity and 83.8% of detection specificity. AAD1 anomaly detector achieved detection sensitivity of 91.4% and detection specificity up to 3.7%. FP is the process of accurately auguring whether a failure will occur or not at a particular location. PAS has achieved 84% of exactness. The FS module forecasts the failure of a host with a precision of 89%. This paper will allow the users to understand the concept better and come up with innovative ideas to develop better solutions.

REFERENCES

[1] Liu, J., Zhou, J., &Buyya, R.: Software Rejuvenation Based Fault Tolerance Scheme for Cloud Applications. IEEE 8th International Conference on Cloud Computing. doi:10.1109/cloud.2015.164. (2015).

[2] Lin Rongheng, Wu Budan, Yang Fangchun, Zhao Yao, & Hou Jinxuan.: An efficient adaptive failure detection mechanism for cloud platform based on volterra series China Communications, 11(4), 1–12. doi:10.1109/cc.2014.6827564, (2014)

[3] Pannu, H. S., Liu, J., Guan, Q., & Fu, S.: AFD: Adaptive failure detection system for cloud computing infrastructures. 2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC). doi:10.1109/pccc.2012.6407740. (2012)

[4] Villarreal-Vasquez, M., Bhargava, B., Angin, P., Ahmed, N., Goodwin, D., Brin, K., &Kobes, J.: An MTD-Based Self-Adaptive Resilience Approach for Cloud Systems. 2017 IEEE 10th International Conference on Cloud Computing (CLOUD). doi:10.1109/cloud.2017.101. (2017).

[5] Pannu, H. S., Liu, J., & Fu, S.: AAD: Adaptive Anomaly Detection System for Cloud Computing Infrastructures. 2012 IEEE 31st Symposium on Reliable Distributed Systems. doi:10.1109/srds.2012.3. (2012).

[6] Kumari, P., & Kaur, K.: A weight-based approach for node failure detection and recovery in mobile cloud computing. 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). doi:10.1109/rteict.2016.7807804. (2016).

[7] Uchiumi, T., Kikuchi, S., & Matsumoto, Y.: Misconfiguration detection for cloud datacenters using decision tree analysis. 2012 14th Asia-Pacific Network Operations and Management Symposium (APNOMS). doi:10.1109/apnoms.2012.6356072. (2012).

[8] Kim, J., & Kim, H. S.: PBAD: Perception-Based Anomaly Detection System for Cloud Datacenters. 2015 IEEE 8th International Conference on Cloud Computing. doi:10.1109/cloud.2015.95. (2015).

[9] Xiong, N., Vasilakos, A. V., Wu, J., Yang, Y. R., Rindos, A., Zhou, Y., … Pan, Y. A Self-tuning Failure Detection Scheme for Cloud Computing Service. 2012 IEEE 26th International Parallel and Distributed Processing Symposium. doi:10.1109/ipdps.2012.126. (2012).

[10] Guan, Q., & Fu, S.: Adaptive Anomaly Identification by Exploring Metric Subspace in Cloud Computing Infrastructures. 2013 IEEE 32nd International Symposium on Reliable Distributed Systems. doi:10.1109/srds.2013.29. (2013).

[11] Lo, C.-C., Huang, C.-C., & Ku, J.: A Cooperative Intrusion Detection System Framework for Cloud Computing Networks. 2010 39th International Conference on Parallel Processing Workshops. doi:10.1109/icppw.2010.46. (2010).

[12] Farshchi, M., Schneider, J.-G., Weber, I., & Grundy, J.: Experience report: Anomaly detection of cloud application operations using log and cloud metric correlation analysis. IEEE 26th International Symposium on Software Reliability Engineering (ISSRE). doi:10.1109/issre.2015.7381796. (2015).

[13] Lee, E. K., Viswanathan, H., &Pompili, D.: Model-Based Thermal Anomaly Detection in Cloud Datacenters Using Thermal Imaging. IEEE Transactions on Cloud Computing, 6(2), 330–343. doi:10.1109/tcc.2015.2481423. (2018).

[14] Rukavitsyn, A., Borisenko, K., &Shorov, A.: Self-learning method for DDoS detection model in cloud computing. 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). doi:10.1109/eiconrus.2017.7910612. (2017).

[15] Guan, Q., Fu, S., DeBardeleben, N., & Blanchard, S.: Exploring Time and Frequency Domains for Accurate and Automated Anomaly Detection in Cloud Computing Systems. 2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing. doi:10.1109/prdc.2013.40. (2013).

[16] Fan, G., Yu, H., Chen, L., & Liu, D.: Model Based Byzantine Fault Detection Technique for Cloud Computing. 2012 IEEE Asia-Pacific Services

Computing Conference. doi:10.1109/apscc.2012.56. (2012).

[17] Pannu, H. S., Jianguo Liu, & Fu, S.: A self-evolving anomaly detection framework for developing highly dependable utility clouds. 2012 IEEE Global Communications Conference (GLOBECOM). doi:10.1109/glocom.2012.6503343. (2012).

[18] S. Fu and C. Xu.: Exploring event correlation for FP in coalitions of clusters. In Proceedings of ACM/IEEE Supercomputing Conference (SC), (2007).

[19] S. Fu and C. Xu.: Quantifying temporal and spatial correlation of failure events for proactive management. In Proceedings of IEEE International Symposium on Reliable Distributed Systems (SRDS). (2007).

[20] Z. Lan, J. Gu, Z. Zheng, R. Thakur, and S. Coghlan.: A study of dynamic meta-learning for FP in large-scale systems. Journal of Parallel and Distributed Computing, 70(6):630–643, (2010).

[21] Y. Liang, Y. Zhang, A. Sivasubramaniam, M. Jette, and R. K. Sahoo.: BlueGene/L failure analysis and prediction models. In Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2006.

[22] R. K. Sahoo, A. J. Oliner, I. Rish, M. Gupta, J. E. Moreira, S. Ma, R. Vilalta, and A. Sivasubramaniam.: Critical event prediction for proactive management in large-scale computer clusters. In Proceedings of ACM International Conference on Knowledge Discovery and Data Dining (KDD), (2003).

[23] Watanabe, Y., Otsuka, H., & Matsumoto, Y.: Failure Prediction for Cloud Datacenter by Hybrid Message Pattern Learning. 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing. doi:10.1109/uic-atc-scalcom.2014.m. (2014).

[24] Natalino, C., Coelho, F., Lacerda, G., Braga, A., Wosinska, L., & Monti, P.: A Proactive Restoration Strategy for Optical Cloud Networks Based on FPs. 20th International Conference on Transparent Optical Networks (ICTON). doi:10.1109/icton.2018.8473938. (2018).

[25] Adamu, H., Mohammed, B., Maina, A. B., Cullen, A., Ugail, H., & Awan, I.: An Approach to FP in a Cloud Based Environment. IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud). doi:10.1109/ficloud.2017.56. (2017).

[26] Ganguly, S., Consul, A., Khan, A., Bussone, B., Richards, J., & Miguel, A.: A Practical Approach to Hard Disk FP in Cloud Platforms: Big Data Model for Failure Management in Datacenters. IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService). doi:10.1109/bigdataservice.2016.10. (2016).

[27] Islam, T., &Manivannan, D.: Predicting Application Failure in Cloud: A Machine Learning Approach. 2017 IEEE International Conference on Cognitive Computing (ICCC). doi: 10.1109/ieee.iccc.2017.11. (2017).

[28] Davis, N. A., Rezgui, A., Soliman, H., Manzanares, S., & Coates, M.: FailureSim: A System for Predicting Hardware Failures in Cloud Data Centers Using Neural Networks. IEEE 10th International Conference on Cloud Computing (CLOUD). doi:10.1109/cloud.2017.75. (2017).

[29] Liu, Y., & Wu, Z.: Non-intrusive Critical System Event Recognition and Prediction in Cloud. IEEE 7th International Conference on Cloud Computing. doi:10.1109/cloud.2014.93. (2014).

[30] Brandt, J., Chen, F., De Sapio, V., Gentile, A., Mayo, J., Pébay, P., … Wong, M.: Using Cloud Constructs and Predictive Analysis to Enable Pre-Failure Process Migration in HPC Systems. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. doi:10.1109/ccgrid.2010.31. (2010).

[31] Soualhia, M., Khomh, F., &Tahar, S.: Predicting Scheduling Failures in the Cloud: A Case Study with Google Clusters and Hadoop on Amazon EMR. IEEE 17th International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. doi:10.1109/hpcc-css-icess.2015.170. (2015).