iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** VI **Month of publication:** June 2019

DOI: http://doi.org/10.22214/ijraset.2019.6156

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

Fault Tolerance using Fitness Algorithm in Cloud Computing

Puja Chaturvedi[1], Brona Shah[2]
[1]*M.E.(Computer),* [2]*Asst. Prof. Computer/IT, Silver Oak College Of Engineering & Technology, Ahmedabad, India*

Abstract: ***With the immense growth of internet and its users, Cloud computing, with its incredible possibilities in ease, Quality of service and on-interest administrations, has turned into a guaranteeing figuring stage for both business and non-business computation customers.. The dynamic environment of cloud results in various unexpected faults and failures.as the number of request increases over the datacenter then the failure probability load is also increases .so the requested task should be balance so there is minimum failure occur over the datacenter.so to overcome this issue ,we proposed a adaptive genetic algorithm using the levenberge marquaradt algorithm(LM) which overcome the difficulties that arise due to fault and existing implemented algorithm .the paper analyze the fault on different parameter in term of request count and request failed***
Keywords: Cloud Computing, fault tolerance, L M equation, adaptive genetic algorithm.

I. INTRODUCTION

The word cloud computing is the general term for the delivery of hosted services over the internet.it is more promising for both client or consumer and provider in different field like education ,marketing, engineering, business , industries and in so many field .cloud computing make the long held dream of utility computing possible with various properties like scalability ,multi-tenancy ,loose coupling , reliable delivery , easier management ,low cost, backup and recovery ,increased storage space, automatic updating ,versatile compatibility and so on[1] . but this is the one side of the coin as every coin has another side that one in cloud as in form of failure and management of that failure is one of the major issue in cloud.

Fault tolerance enable the system to continue operation possibly at reduced level rather than failing completely There are many type of fault occur over the system like transient error which is temporary error that like to disappear soon ,intermittent error that randomly occur over the sub component ,permanent error which is difficult to solve and continue to exist until the faulty component is repaired ,byzantine fault in which system behave incorrectly or unpredictable manner and last is crash failure in which system component stop functioning or remain inactive which is one of the major reason of the fault ,all these error can reduce the reliability rate of the datacenter and task completion rate .[2]

The fault tolerance system can be of two type one is proactive and other is reactive .in proactive fault tolerance predicts the fault and replace the suspected component by other working component or transferred the task in other data centre i.e. as software rejuvenation ,proactive fault tolerance using self-healing and pre-emptive migration etc. and in reactive fault tolerance are used to reduce the impact of failure on a system when failure occur i.e. as checkpoint ,replication ,job migration, retry , task resubmission and rescue workflow etc.[1] This paper focus on the fault tolerance base on method to increase the accuracy of the system by minimizing the fault probability and task completion time using fitness algorithm which will overcome the gaps of previously implemented algorithms .given paper is divided into six section in which II describe the existing related work followed by contribution of our work in III , proposed methodology , implementation of algorithm and conclusion in section IV, V and VI respectively.

II. RELATED WORK

Author JI WANG, WEIDONG BAO, XIAOMIN ZHU, LAURENCE T. YANG, AND YANG XIANG, proposed an approach of FESTAL[3] which focuses on the reliability and availability of cloud for real time application in safety critical nature . the technique used is resource scaling up and resource scaling down to copies the task to some idle system and eliminate the long idle system for better utilization. author also incorporate the use of primary and backup copies in case of failure and BB overlapping technique . this technique provide better performance ratio of task over host time and gain ratio but it do not consider the task dependency for the particular system.

To overcome the above problem XIAOMIN ZHU, JI WANG, HUI GUO, DAKAI ZHU, LAURENCE T.YANG, AND LING LIU[4] suggest the new advance approach FASTER which base on the same technique as FESTAL but do consider the task dependency .FASTER also incorporate the backward shifting method to make full use of the idle resources and incorporates task

©IJRASET: All Rights are Reserved

overlapping and VM migration for high resource utilization and vertical/horizontal scaling-up technique to quickly provision resources for a burst of workflows but it has disadvantage of not take consideration of predicting the fault .

MOHIT KUMAR GOKHROO, MAHESH CHANDRA GOVIL, EMMANUEL S. PILLI.[5] Suggest new approach in fault tolerance for detection and mitigation both in cloud computing .the approach base ON the method of detecting the fault on running status of the job. The detection algorithm periodically monitors the progress of job and reports the stalled job due to failed VM to fault tolerant manager (FTM). The prediction method is a proactive measures to detect abnormal condition before it actually occur .the advantages of this approach is in low resource wastage and avoid the penalty of service level agreement .

BASHIR MOHAMMED, MARIAM KIRANAND IRFAN AND ULLAH AWAN, KABIRU.M. MAIYAMA[6] suggest IVFS (integrated virtualized strategy for fault tolerance in cloud computing environment) allow the system to possibly function at a reduced level, rather than failing completely. as a major concern in guaranteeing availability and reliability of critical services or application execution in cloud environment, cloud fault tolerance research focuses on detection and recovery strategies. Author present the A working model of the strategy and a mathematical relationship to represent the fault tolerance model system using the concept virtualization and checkpoint/replay scheme. the checkpoint/replay model is developed using the reward renewal process (RRP) theory a backward recovery is performed and the Vm is immediately restarted and recovered from the last successful checkpoint. but due to checkpoint the performance loss is occur by overhead checkpoint recovery time and recomputing time

J.SONIYA, J.ANGELA JENNIFA SUJANA, DR.T.REVATHI [7]Focus on fault tolerance and resource allocation policies for cloud environments. In the existing system, Primary Backup (PB) model is used, but it does not contain any dynamic resource allocating mechanism. It propose a dynamic resource allocating mechanism with fault tolerance to improve resource utilization. It incorporate a backup overlapping mechanism and efficient VM migration strategy for designing novel Dynamic Fault Tolerant Scheduling Mechanism for Real Time Tasks in cloud computing. Proposed model aims at achieving both fault tolerance and high resource utilization in the cloud .The technique used is dynamic resource allocation mechanism and dynamic resource expansion and contraction .

PUNEET GUPTA AND S P GHRERA[8] suggest the fault tolerance mechanism base on the load over the datacenter as the request over the datacenter is increases which lead to the increase in the failure probability ,so the request is to be maintain by using a learning algorithm that consider the two parameter to reduce the failure rate one is the time taken to complete the request which known as makespan and fault probability. But the paper do not consider the priority between the total time completion and task fault rate and also the proposed algorithm do not give better result in case of high task request.

There are various fault tolerance strategies which are proposed by authors that also play a crucial role in fault tolerance mechanism such as Y. ZHANG, Z. ZHENG, and M. R. LYU proposed BFTCloud [9] for voluntary resource cloud computing The architectures operate on five basic operations: primary node selection, replica selection, request execution, primary node updating, and replica updating. The primary node is selected based on QoS requirements. The request for the service is handled by the primary node. G. FAN, H. YU, L. CHEN, and D. LIU. Propose CFN[10] for service resources, cloud module, the detection and failure process ... etc. It is used to create the different components of Cloud Computing which gets integrated dynamically into CFN model. Based on CFN model, the properties of the components are analyzed developing a fault detection strategy at each level which dynamically detects the faults in the execution process. P. DAS, and P. M. KHILAR. suggested VFT[11] a reactive fault tolerant technique; it consists of a Cloud Manager (CM) module and a Decision Maker (DM) which are used to manage the virtualization, load balancing and to handle the faults. The first step involves virtualization & load balancing and in the second step fault tolerance is achieved by checkpoint and fault handler. The virtualization includes a fault hander. Fault handler finds these unrecoverable faulty nodes and It also helps to remove the temporary software . G. CHEN" H. JIN, D. ZOU, B. B. ZHOU, W. QIANG, and G. HU. Suggest SHelp[12] is an error handler which are run in different VMs hosted on one physical machine. It uses the Checkpoint/Restart as the checkpoint and rollback tool. Authors introduced two new techniques, namely, weighted rescue points and two-level rescue point database. When a fault occurs, the application is rolled back to a latest checkpoint, and first uses error virtualization at a rescue point which has the largest weight value among the candidate rescue points, then at the rescue point with the second largest weight value and so on until the fault is bypassed. S. MALIK, AND F. HUET. Suggest FTRT[13] is the highly intensive computing capabilities and scalable virtualized environment of the clouds help the systems to execute the tasks in real time. The proposed technique depends on the adaptive behavior of the reliability weights assigned to each processing node. The technique uses a metric to evaluate reliability. The nodes are removed if the processing nodes fail to achieve the minimum required reliability level.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

III. OUR CONTRIBUTION

In the survey I conclude that in existing approaches are either discuss about task scheduling and resource management and utilization and various proactive and reactive method to decrease the fault over the system but the existing algorithm do not consider the factor or parameter that responsible for the software fail. The problem with these approaches are that they do not consider the load over the datacenter or Vm to cope with the request task and efficient time to complete that task, which we conclude that as number of request over the data center increases which increase the load and failure probability over the data center and decrease the reliability of completing the maximum task to complete over the particular Vm . A fault management system should match the task to the right Vm with all resources so the chances of occurrence of fault is less. So to overcome this issues we propose an efficient fault management system by considering the different factors such as the request failed count and request complete count by the Vm to complete the task. Also applying a priority base genetic algorithm using the Levenberg Marquaradt [14] fitness equation to decrease the fault probability rate.

IV. PROPOSED METHDOLOGY

A. Proposed Solution

The proposed method is to incorporate the fault aware learning based resource allocation algorithm by proposing the priority base genetic algorithm .the priority base genetic algorithm helps to find the solution which cannot be achieved by any static or dynamic algorithm which help in learning the previous past probability rate to reduce the future fault rate and it also help to find the fittest solution .the priority base genetic algorithm help in choosing the priority between the task completion or failure probability which take consideration the service level agreement which cannot be done on simple genetic algorithm.

The proposed algorithm incorporate LM (Levenberg Marquaradt) fitness equation to find the fittest solution and also to minimize the machine fluctuation by different task size and waiting time based on linear equation .Algorithm is uses the Poisson distribution for random request failure at virtual machine such as host and datacenter.

B. Proposed Model

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems. It is an optimization method based on population and is based on Darwin's theory of evolution . In GA, each possible solution is represented by a chromosome. An initial population is taken randomly and it is used as a starting point. A fitness function is calculated for each chromosome so that it is known whether the chromosome is suitable or not. Crossover and mutation functions are performed on the selected chromosmes and offsprings for new population are created. This process is repeated until enough offsprings are created.

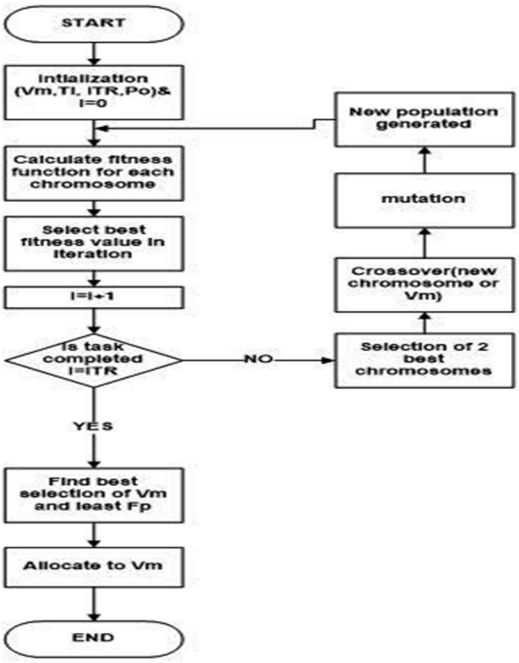


Fig. 1. Proposed model

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

C. Phases Of Proposed Algorithm

1) *Intialization:* In this phase we have a set of tasks (T1, T2, T3, T4, T5, T6…. T n) and a set of resources in term of virtual machine (VM1, VM2, VM3, VM4, VM5…. VM m) are pre allocated on hosts in distributed datacenter . Here we initialize asset of sequences or schedules allocated randomly, each sequences act a chromosomes for genetic algorithm. The complete set of chromosomes is said to be a population, acting as a input for algorithm.
2) *Selection:* In this phase we evaluate the fitness value for each set of sequence or chromosome, which depends up on the computing capability, total time taken to complete the schedule and the failure probability of complete schedule. Since faults over a datacenter are random in nature and follows Poisson distribution, which over a period of time t and t + ΔT and fault probability as FPi defined the equation given below over the Host i and λ is fault rate over a time T

$$FPi(t<=T<=t+\Delta|T>t)=\frac{\exp(-\lambda t)-\exp(\lambda(t+\Delta T))}{\exp(-\lambda t)} \quad (1)$$

$$FPi\ (t)=1-\exp(-\lambda\Delta t) \quad (2)$$

The levenberg-marquardt (lm)[14] algorithm is the most widely used optimization algorithm. it outperforms simple gradient descent and other conjugate gradient methods in a wide variety of problems defined in (3)where Δm_j =free request queue for server j and Q_L is queue load of threshold that define threshold of virtual machine using CPU,RAM and bandwidth and fitness equation in (4) and (5) in which λ_j=service rate FR_j=fault rate and W_j is waiting time for a request at server j

$$\Delta m_j = Q_L_{threshold} - Q_L_{current}\ ) \quad (3)$$

$$F_j=\propto_1\Delta m_j + \propto_2\lambda_j + \propto_3(\frac{1}{FRj}) + \propto_4(\frac{1}{Wj})\ ) \quad (4)$$

$$\propto_1 +\propto_2 +\propto_3+\propto_4=1\ (5)\ ) \quad (5)$$

3) *Crossover:* In this step two fittest solutions based on least make span and failure probability is selected. We have used multi point crossover to generate new fittest sequences/ chromosome. Steps to generate crossover are as follows.

a) The two fittest chromosomes are selected
b) A new fittest chromosome is generated using multi point cross over by interchanging the set of two chromosomes.
c) The new chromosome replaces the chromosome with highest fitness value.

4) *Mutation:* In this phase new merging the new offspring, which can be better solution with remaining which, regenerates population keeps the total population size constant after each iteration. After specific count of iteration predefined as an input to genetic algorithm, best chromosome is selected i.e. the chromosome with least fitness value is selected for schedule

D. Proposed Algorithm

Given below is the proposed algorithm which base on the genetic algorithm phases descri be in section c

FGDATA(VM list VM_i ,Task list T_i , population size Po ,iteration itr)

1. VM_i=VM_List();
2. FI=getFault();
3. I=no of VM
4. T_i=Task_List();
5. C=Gentic_algo(Vm_i ,t_i ,Po, itr);
6. Allocate_Resource(C)
7. End

Fig. 2. Initialization

Genetic_algo(VMi_i , T_i , Po, ltr)

1. Po= Initiate_Population(T_i);
2. Evalution();
3. C1=getFittest1();
4. C2=getFittest2();
5. Crossover(C1,C2)
6. Mutation(Po,C1,C2);
7. Return(getFittest());
8. End

Fig. 3. Genetic Algorithm

©IJRASET: All Rights are Reserved

1. Evalutin() {
2. For each C_i i=0-P_0
3. For each T_i
4. $Temp_F_j=\alpha_1\Delta m_j + \alpha_2\lambda_j + \alpha_3(\frac{1}{FR_j}) + \alpha_4(\frac{1}{W_j})$
5. $Fitness_i = Fitness_i + temp$
6. End
7. END
8. }

Fig. 4. Evaluation

1. Allocate_Resource(c){
2. C_i=Gethromosome();
3. for each Ci
4. Allocate(Ci);
5. End
6. }

Fig. 5. Allocation

V. IMPLEMENTATION

A. Software Specification

For the implementation of our proposed algorithm , we will using CloudSim 3.0[15] which is the event driven simulator and implementing code using JAVA language in IDE eclipse which is the most powerful object oriented programming language used in CloudSim, CloudSim module can be easily extendable with user's requirement .Proposed algorithm is being tested over the various test case with 10 server ,2 GB RAM ,CORE_QUAD,2- HOST and requesting poisson distribution model for random request and fault model in distributed environment.

B. Result

Testing of proposed algorithm is done with fault aware genetic allocation algorithm proposed by Punit Gupta and S.P. Ghrera. Figure shows the improvement in number of request completed vs total number of request based on ratio of given data and the next figure shows the increase in request size completed over the total size of request in existing algorithm and proposed algorithm

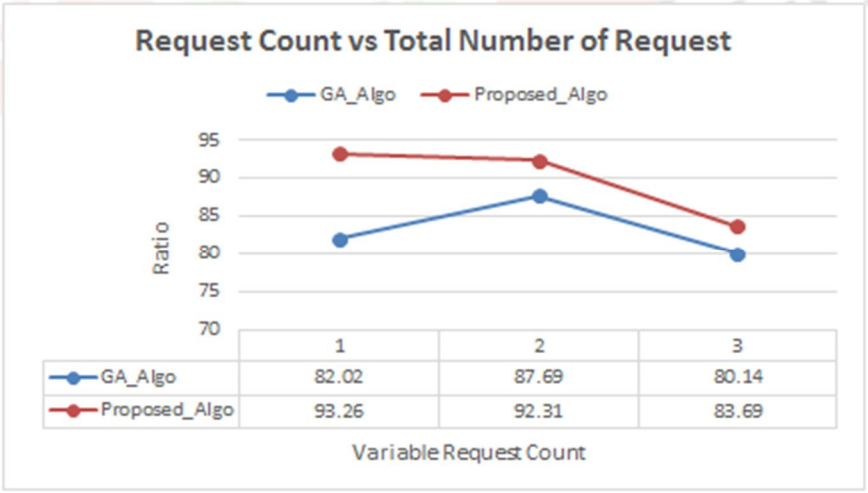


Fig. 6. Comparison in improvement in request completed

©IJRASET: All Rights are Reserved

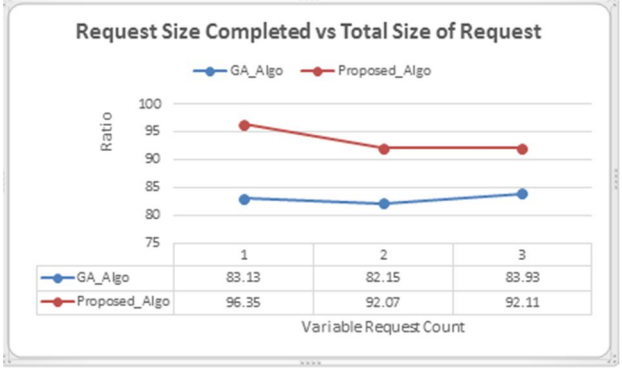


Fig. 7. Comparison in improvement in request size

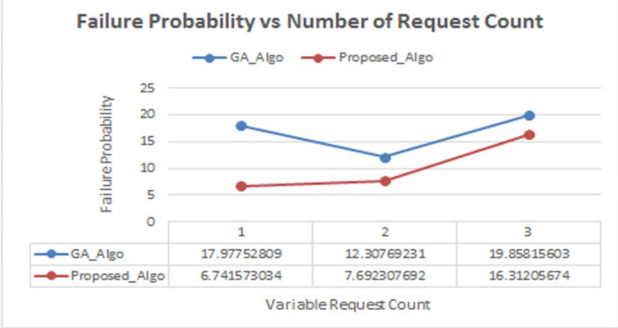


Fig. 8. Comparison in decrease in failure probability

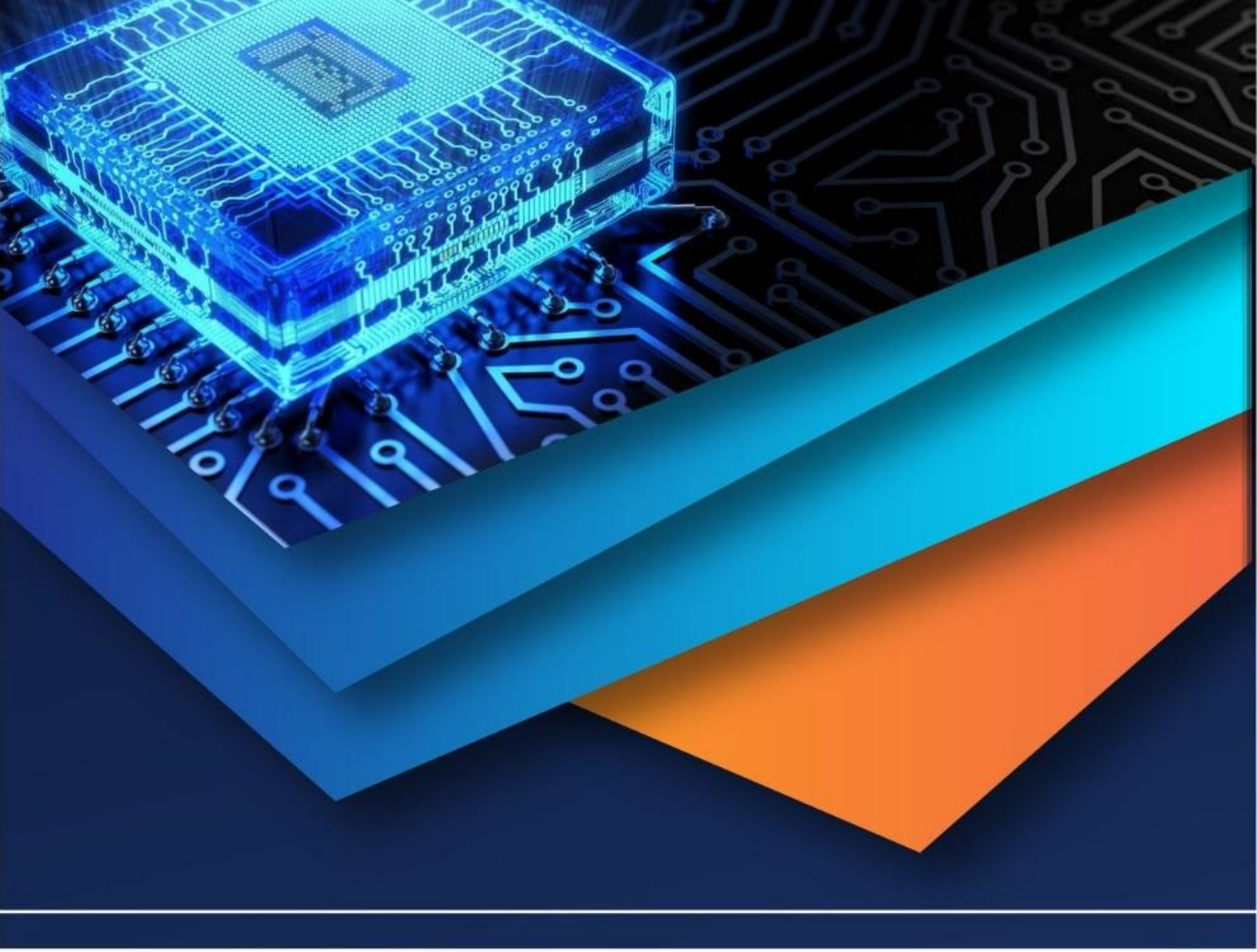
VI. CONCLUSION

Reliability and availability of cloud is become major concern of cloud service provider to enhance the scalability and elasticity for real time world and scientific workflow where deadline should be satisfied so fault tolerance mechanism is used the given paper propose an efficient fault tolerance mechanism which incorporate adaptive learning method which also include the fitness equation .the propose architecture overcome the limitation of the existing work but also show the increase in parameter which consider during the fault.

REFERENCES

[1] Zeeshan Amin,Nisha Sethi,Harshpreet Singh, "A Review Paper on Fault Tolerance in Cloud Computing", International Journal of Computer Applications, 2015
[2] Kashif Bilal, Osman Khalid, Saif Ur Rehman Malik, Muhammad Usman Shahid Khan, Samee U. Khan and Albert Y. Zomaya "Fault Tolerance in the Cloud" ,ENCYCLOPEDIA OF CLOUD COMPUTING, 2016
[3] Ji Wang, Weidong Bao, Xiaomin Zhu, Laurence T. Yang, and Yang Xiang, "FESTAL: Fault-Tolerant Elastic Scheduling Algorithm for Real-Time Tasks in Virtualized Clouds" ,IEEE, 2014
[4] Xiaomin Zhu, Ji Wang, Hui Guo, Dakai Zhu, Laurence T.Yang, and Ling Liu "Fault-Tolerant Scheduling for Real-Time ScientificWorkflows with Elastic Resource Provisioning inVirtualized Clouds", IEEE, 2015
[5] Mohit Kumar Gokhroo, Mahesh Chandra Govil, Emmanuel S. Pilli , " Detecting and Mitigating Faults in Cloud Computing Environment", IEEE, 2017
[6] Bashir Mohammed, Mariam Kiranand Irfan and Ullah Awan, Kabiru.M. Maiyama, "An Integrated Virtualized Strategy for Fault Tolerance in Cloud Computing Environment ", IEEE, 2016
[7] J.Soniya, J.Angela Jennifa Sujana, Dr.T.Revathi, "Dynamic Fault Tolerant Scheduling Mechanism for Real Time Tasks in Cloud Computing ", IEEE, 2016
[8] Punit gupta ,S.P. Ghrera ,"Fault And QoS Based Genetic Algorithm forTask Allocation in Cloud Infrastructure",International Journal Of Control Theory and Application,2016
[9] Y. Zhang, Z. Zheng, and M. R. Lyu. "BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing. " , IEEE, 2011.
[10] G. Fan, H. Yu, L. Chen, and D. Liu. "Model Based Byzantine Fault Detection Technique for Cloud Computing. ",IEEE, 2012
[11] P. Das, and P. M. Khilar. "VFT: A virtualization and fault tolerance approach for cloud computing,IEEE,2013.
[12] 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. ", IEEE ,201
[13] S. Malik, and F. Huet. "Adaptive Fault Tolerance in Real Time Cloud Computing. " ,IEEE, 2011
[14] Ananth Ranganathan,"The Levenberg-Marquaradt Algorithm",2004
[15] Rodrigo N. Calheiros,Rajiv Ranjan ,Rajkumar Buyya ,"CloudSim: a ttoolkit for modelling ana simulation of cloud computing environment and evaluation and evaluation of resource providioning algorithm" ,IEEE,2011

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)