

# A New Enhanced Mechanism for Fault Tolerance using Resource Optimization

Harpreet Kaur  
M.Tech Research scholar  
Sri Guru Granth Sahib World University  
Fatehgarh Sahib, Punjab

Amritpal Kaur  
Assistant Professor  
Sri Guru Granth Sahib World University  
Fatehgarh Sahib, Punjab

## ABSTRACT

In these days there is a very little fraction of peoples who does not know about the cloud. Cloud offers various services to its users on demand. In Cloud Computing one of the major problem areas is Fault Tolerance. In cloud there is a term named virtualization that is very important and common. In virtualization that there is a dynamic number of virtual machines having different operating systems run on a single physical machine. In this paper a fault tolerance technique i.e. Enhanced Energy Aware Fault Tolerance Resource Optimization (EEAFTRO) technique is proposed. The purpose of this proposed technique is to optimize the resources efficiently so that the possibility of fault becomes less. If after effecient resource optimization or resource allocation fault occurs, then by using effective fault tolerance technique it can be handled. The proposed algorithm helps to minimize the execution time of tasks.

## Index Terms

Cloud Computing, Fault Tolerance, Resource utilization, with fault, without fault.

## Keywords

SaaS, PaaS, IaaS, VM's, Reactive, Proactive.

## 1. INTRODUCTION

As the internet services and virtualization [1] comes into play the use of cloud increases very rapidly. The reason for the fast interest of peoples in cloud is its virtualization concept due to that the virtual resources are provided to the user. Through internet technology cloud is provided to the customers as a service and the customer pay as per use.

The service provider is known as cloud [2]. Basically three types of services are provided by cloud to its customers. First one infrastructure as a service (IaaS), this service provides the infrastructure to the customers where customers can manage storage and computing resources. Second service is Platform as a service (PaaS), this service provides the platform to its customers where the customers can create their own software and use it. Third service is software as a service (SaaS), this service provides the software to its customers. Customers can directly use the software through internet without installing any software on their own computer.

In cloud environment [3] the jobs are continuously submitted by the users to the data centers and the resources are provided by the cloud to the jobs submitted for execution. When the jobs complete the execution then it leaves the node and that node becomes available to any other job which does not starts its execution till now. There will be some situations when the node will become unavailable due to any failure and then there is a need to migrate the job from that failed node to another node that is working correctly and is available. In this paper a new Enhanced Mechanism for Fault Tolerance using Resource

Optimization is proposed. In this technique the resources are optimized by using Genetic algorithm. The main focus of proposed mechanism is an efficient allocation of resources with higher capabilities to reduce the failure rate and in case of failure of any virtual machine; the task migration policy is used. In the proposed work these are 3 scheduling phases performed; these are static scheduling, dynamic scheduling, and optimized scheduling.

The rest of the paper is organized as follows. Section 2 gives the previous work done in this field. Section 3 describes gives the proposed work that is Enhanced Energy Aware Fault Tolerance Resource Optimization (EEAFTRO). Section 4 gives the result analysis and comparison. Section 5 gives the conclusion and future scope of the proposed technique.

## 2. LITERATURE SURVEY

Authors in papers [4][5][6] describes the various existing fault tolerance models. These various fault tolerance techniques are used for handling different types of faults. Paper [7] discusses the various workflow scheduling algorithms used in Cloud Computing.

Yue Gao, Sandeep K. Gupta, Yanzi Wang, Massoud Pedram [8] proposed an Energy Aware fault Tolerance Scheduling Framework. Scheduling algorithm has two different phases that are static scheduling and dynamic scheduling. In static scheduling the task workload graph is created and according to that the jobs are executed and in dynamic phase the jobs are migrated during execution in case of failure and re-executions. By using this scheduling algorithm the reliability and performance increases while the energy overhead decreases approximately 76% and the failures rates reduced 50% compared to the current replication techniques.

Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li [9] proposed a scheduling model based on an algorithm named MO-GA. The proposed model reduces the energy consumption but increases the profit of service providers. The different operators are used like crossover, encoding rules, and selection. The jobs scheduling is done according to multi objective genetic algorithm. The resources allocation is done according to the suitability of resources to the applications to improve the effectiveness and efficiency.

Anjali D.Meshram, A.S.Sambare, S.D.Zade [10] proposed a FTMC model i.e. Fault Tolerance Model for cloud. The reliability of the computing nodes is checked and if any node is not performing well for application execution then it is removed. There are two algorithms Reliability Assessment algorithm and Decision Maker algorithm. There are different modules in the working of FTMC model those are Acceptor module, Timer module, Reliability Assessor, and decision maker. firstly the reliability of virtual machine is set and the reliability changes after every computing cycle. In starting the reliability of any virtual machine is full means 100% and if any

machine produces the result within time limit then its reliability increases but in other case if it fails to produce the result within time then its reliability decreases and when any machine's reliability goes below the threshold then it is removed.

Giuseppe Portaluri, Stefano Giordano, Dzmitry Kliazovich, Bernab e Dorransoro [11] proposes a power efficient resource allocation algorithm for user submitted tasks. The algorithm is based on the Genetic algorithm. By this technique the performance and scalability of millions of tasks increases. The various operations are performed on the population so the resources are allocated to the tasks and this technique is power efficient.

### 3. PROPOSED EEAFTR0 TECHNIQUE

Cloud Computing satisfies its users by providing many different services. The user level is highly increased to utilize the services in cloud computing. Fault Tolerance is a major issue in cloud computing. By using an efficient technique of fault tolerance the availability and reliability of critical services and application

execution is increased. To minimize the failure impact on the system and application execution the failures should be handled.

The previous work which took under consideration proposed an Energy-Aware Fault Tolerant Scheduling Framework. This algorithm includes a static scheduling phase the task workload graph is created and according to that the jobs are executed and in dynamic phase the jobs are migrated during execution in case of failure and reexecutions. By the use of this scheduling algorithm the reliability and performance increases while the energy overhead decreases approximately 76% and the failures rates reduced 50% compared to the current techniques of replication. But by inefficient allocation of resources the systems performance is also degraded.

So to enhance this work, the main focus of proposed technique (EEAFTR0) is an efficient allocation of resources with higher capabilities to reduce the failure rate. This will be achieved by using user required resource distribution as according to their task count and also by migrating only the pending tasks by early detection of faults.

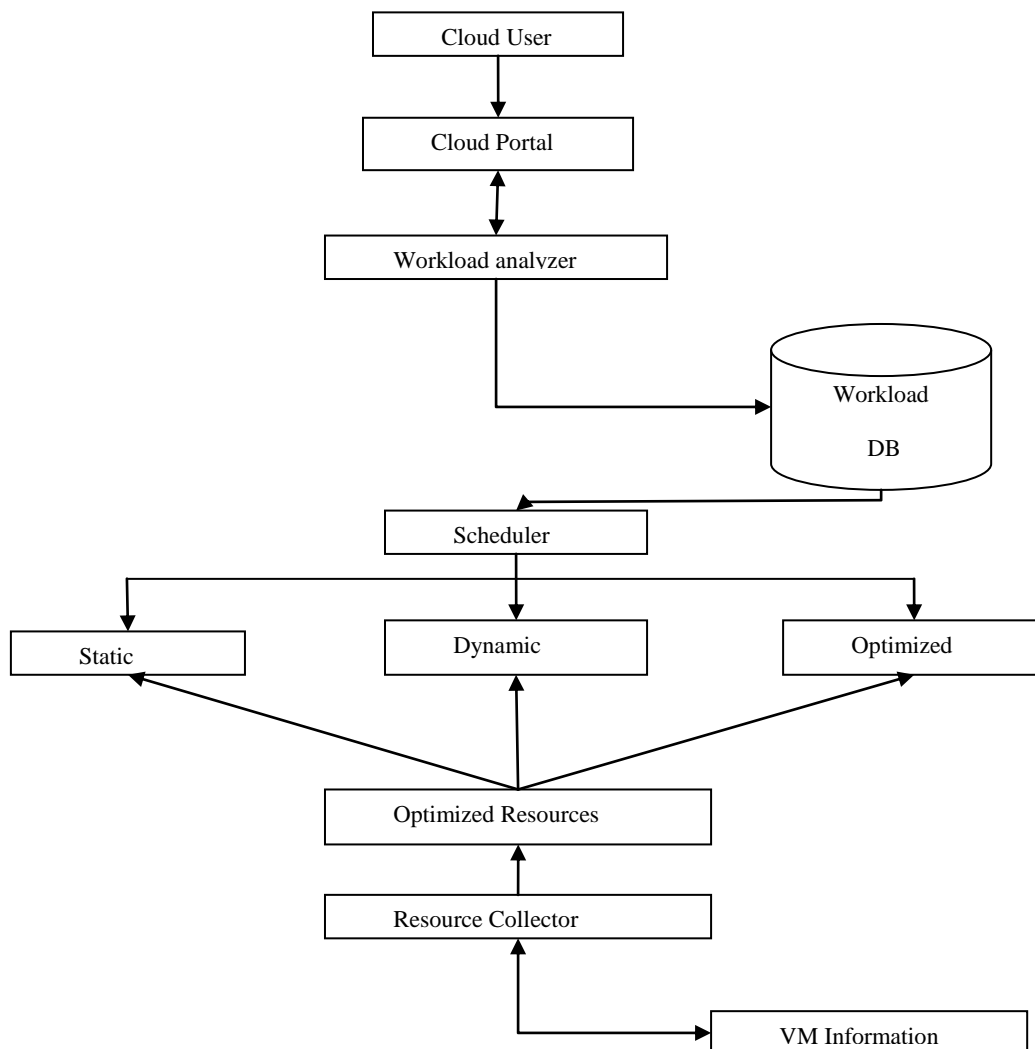


Fig 1: Flowchart of the Work

After the study of previous work in fault tolerance field, it comes to know that the main problem issue in fault tolerance is detection of fault and recovers them. Due to the concept of virtualization in Cloud Computing the Fault Tolerance is still a big challenge. To attain the proposed technique the local cloud environment is

generated in which a client-server environment is created and by using this environment, the users can access the cloud by their personal login usernames and passwords. Users submit their tasks on the cloud and the workload analyzer save all these jobs to the database so that these jobs can be further executed. The main task

of resource collector is to get the information of all the virtual machines created on resources. After getting the information about all the resources the scheduling is done for efficient allocation of resources and if any machines is near about to fail then by advance detection of failure the job is migrated to any other virtual machine and it continuous its execution from the end point.

Flowchart (see Fig 1) shows the implementation of the new technique. This work is user for reducing the wastage of resources by efficient allocation of resources which also tends to reduce the failure rate and minimizing the execution time.

The algorithm design in steps is as follows:

Step1: To use this local cloud environment the user first of all creates an account and will be login by using this account.

Step2: Once a user login then becomes able to submit the jobs to cloud environment.

Step3: the whole information of all the resources i.e. R\_id, R-name, R-RAM, R\_Temp is saved and performs step 4.

Step 4: according to the temperature of resource the virtual machines are created. If  $R\_Temp < 5$  then 5 VM's will be created on one resource but if  $10 < R\_Temp < 5$  then 3 VM's will be created but if  $R\_temp > 10$  then 1 VM will be created on one resource.

Step 5: the whole information of the submitted jobs is saved i.e. J\_id, J-name, J-desc, J-inst and Job scheduler comes into action and performs the Step 6.

Step 6: static scheduling is done of jobs. The resources are allocated to the users as per requirement. If the selected resources are less than the required resources then the average (avg) of free resources is calculated and selected resources+avg is assigned to the users.

Count=no. of users;

tot\_resources= no of machines;

for each users

Count1=no of task to be submitted by each user end

T\_count1=total number of tasks

Assign resources

for each users

```

select_resources(i)=T_count1/count;
required_resources=count (number of tasks for user i)
if required_resources(i)<Select_resources(i)
    tot=select_resources(i)-required_resources(i)
    tot_resources=tot_resources+tot;
end

if select_resources(i)<=required_resources
    add_resource=avg_count(free resources)
end

select_resources(i)=T_count1/count+

```

add\_resource  
end

Step 7: then scheduler checks if the machine\_failure==true and Detect\_Crash==true then the dynamic scheduling is performed.

Step 8: if any machine is about to fail then it checks any other machine which is free and maps the job to that machine and the job continuous it's execution from the end point. If any resource is not available then it waits until the resource becomes available and then starts execution.

#### 4. RESULTS & COMPARISON

The performance of this proposed technique is analyzed by using four parameters that are Time, Resource utilization, With Fault and Without Fault. There are 30 resources and the number of tasks is different. The analysis is done for 15 tasks, 30 tasks, and 60 tasks. The different cases are as follows:

- **Case 1:** when there is no fault the comparison of previous technique EAFTS and proposed technique EEAFTR0 is done. Execution time is calculated for 15 jobs and 30 jobs. Number of resources is 30.

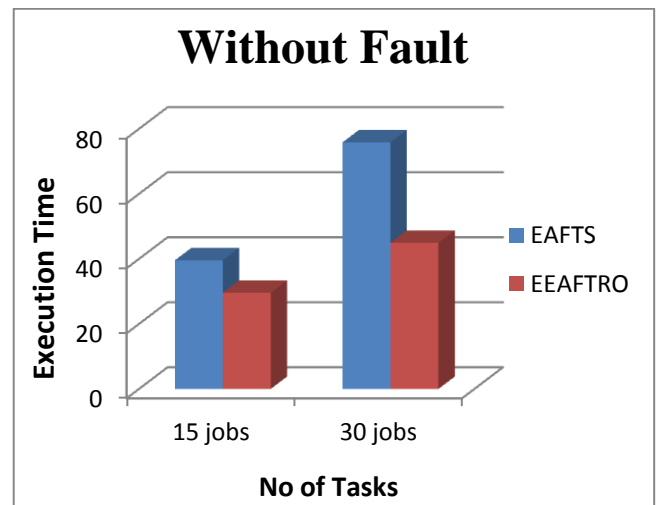


Fig.2: Execution Time Vs No. of Tasks for 15 and 30 tasks in case of no fault

**Case 2:** When there is no fault the comparison of previous technique EAFTS and proposed technique EEAFTR0 is done. Execution time is calculated for 30 jobs and 60 jobs. Number of resources is 30.

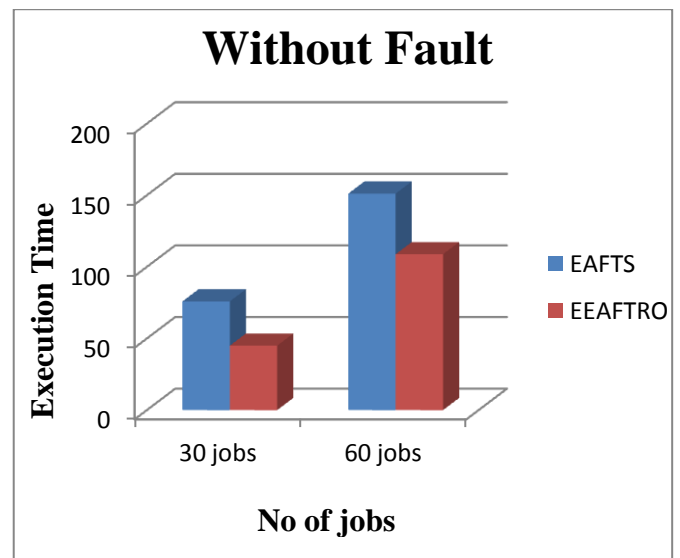
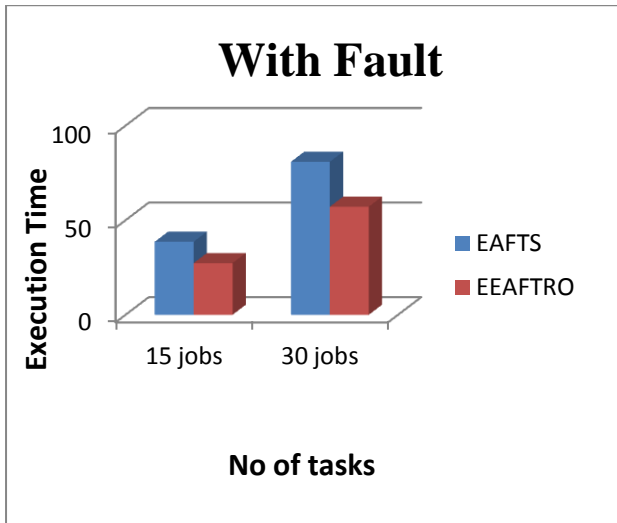


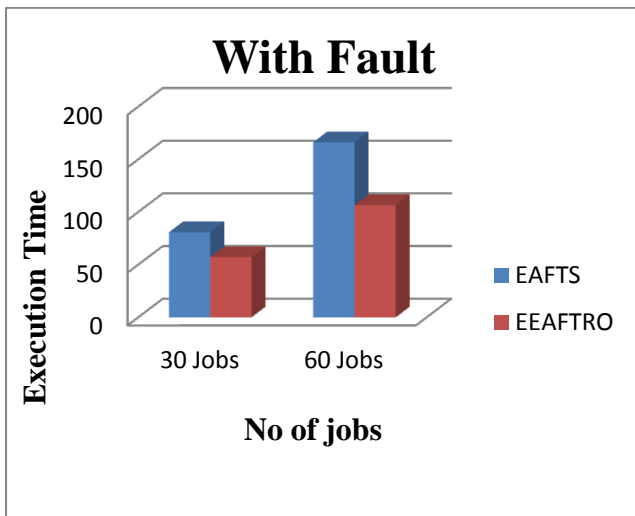
Fig.3: Execution Time Vs No. of Tasks for 30 and 60 tasks in case of no fault

- **Case 3:** When there is fault, the comparison of previous technique EAFTS and proposed technique EEAFTRO is done. Execution time is calculated for 15 jobs and 30 jobs. Number of resources is 30.



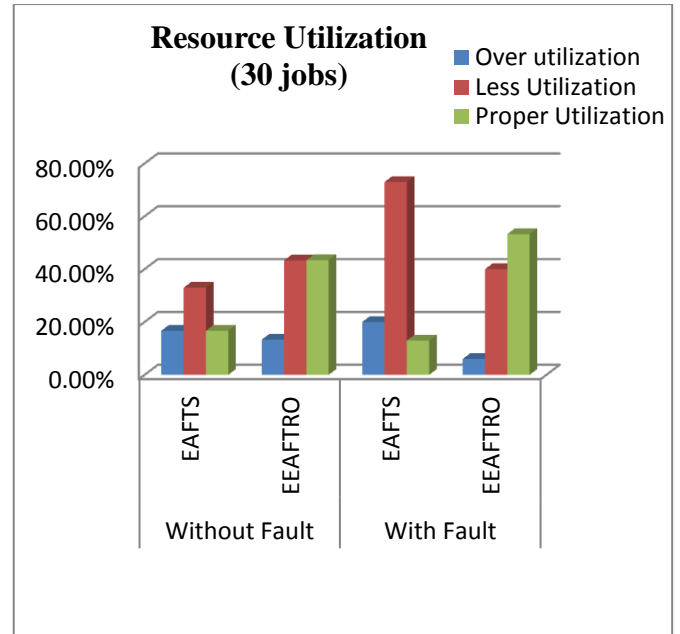
**Fig.4 : Execution Time Vs No. of Tasks for 15 and 30 tasks in case of fault**

- **Case 4:** when there is fault, the comparison of previous technique EAFTS and proposed technique EEAFTRO is done. Execution time is calculated for 30 jobs and 60 jobs. Number of resources is 30.



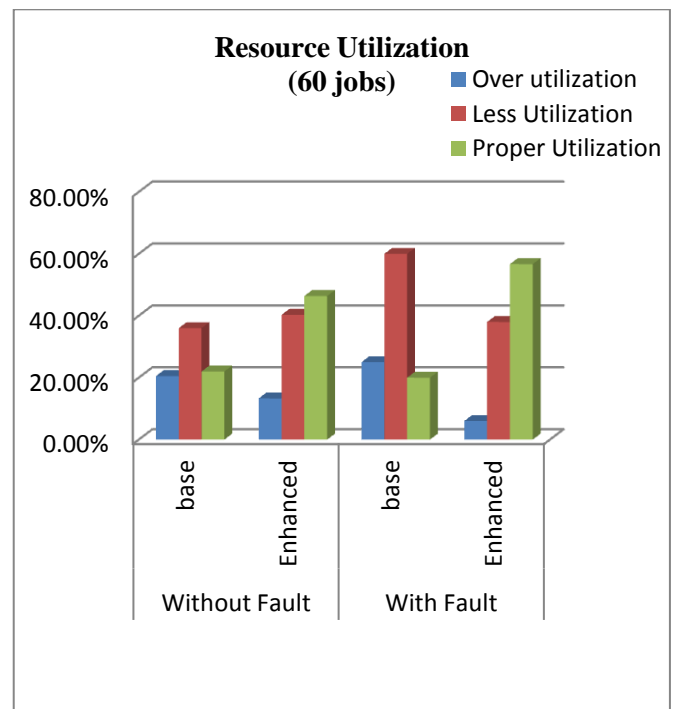
**Fig.5: Execution Time Vs No. of Tasks for 30 and 60 tasks in case of fault**

- **Case 5:** Resource utilization is calculated for both techniques in both case when there is no fault and when there is fault presented. The number of tasks is 30 and the number of resources is also 30 in this case.



**Fig.6: resource utilization of EAFTS versus resource utilization of EEAFTRO for 30 jobs**

- **Case 6:** Resource utilization is calculated for both techniques in both cases when there is no fault and when there is fault presented. The number of tasks is 60 and the number of resources is 30 in this case.



**Fig.7: resource utilization of EAFTS Vs resource utilization of EEAFTRO for 60 jobs**

In each case, the proposed Enhanced Energy Aware Fault Tolerance resource Optimization gives result as per expectation, which is verified by comparing with existing previous technique that is Energy Aware Fault Tolerance Scheduling.

## 5. CONCLUSION

In this paper, the basic introduction of cloud computing is given. Then the previous work is reviewed for Fault Tolerance and Scheduling techniques.

As the prevalence of computing still continues to rise, the need for efficient fault tolerance techniques is increasing. Fault Tolerance is a major problem area in Cloud Computing and there are number of algorithms and techniques to handle these faults. The proposed Enhanced energy Aware Fault Tolerance resource Optimization Technique handles the faults reactively and proactively. Proactively handling means before actual occurrence of fault, by efficient scheduling fault can be handled and reactively means after occurrence, fault can be handled by enhanced fault tolerance technique. This new approach helps in minimizing the execution time thus saves energy and increases the performance of application.

Analysis of the results of the simulation experiment shows, that the proposed algorithm effectively improves the resources utilization i.e. the resources are properly utilized more in the proposed technique and minimizing the execution time in case of fault or no fault.

As result analysis represents that the implementation is done in the local cloud environment. So in future the proposed algorithm will be tested and analyzed in the real time cloud environment.

## 6. REFERENCES

- [1] Asmita Pandey, Pooja, “ Cloud Computing – An on Demand Service Platform”, International Conference on Advances in Management and Technology (iCANT - 2013), pp 5-9
- [2] Pardeep Kumar\*, Amandeep Verma, “Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 5, May 2012 ISSN: 2277 128X, pp 111-114
- [3] Dilbag Singh, Jaswinder Singh, Amit Chhabra,” Evaluating Overheads of Integrated Multilevel Checkpointing Algorithms in Cloud Computing Environment”, I. J. Computer Network and Information Security, 2012, 5, 29-38, pp 29-38
- [4] Dr. Lakshmi Prasad Saikia Yumnam Langlen Devi, “Fault Tolerance Techniques and Algorithms in Cloud Computing”, International Journal of Computer Science & Communication Networks, Vol 4(1),pp 01-08
- [5] Jasbir Kaur, Supriya Kinger, “Analysis of Different Techniques Used For Fault Tolerance”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, pp 4086-4090
- [6] Prasenjit Kumar Patra, Harshpreet Singh, Gurpreet Singh, “Fault Tolerance Techniques and Comparative Implementation in Cloud Computing”, International Journal of Computer Applications (0975 – 8887) Volume 64– No.14, February 2013, pp 37-41.
- [7] Nallakumar R, Sruthi Priya K.S., “A Survey on Deadline Constrained Workflow Scheduling Algorithms in Cloud Environment”, International Journal of Computer Science Trends and Technology (IJCSST) – Volume 2 Issue 5, Sep-Oct 2014, pp 44-50
- [8] Yue Gao, Sandeep K. Gupta, Yanzhi Wang, Massoud Pedram, “An Energy-Aware Fault Tolerant Scheduling Framework for Soft Error Resilient Cloud Computing Systems”, 978-3-9815370-2-4/DATE14/©2014 EDAA IEEE
- [9] Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li, “Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm”, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013, pp134-139
- [10] Anjali D.Meshram, A.S.Sambare, S.D.Zade, “ Fault Tolerance Model for Reliable Cloud Computing”, International Journal on Recent and Innovation Trends in Computing and Communication ISSN 2321 – 8169 Volume: 1 Issue: 7, pp 600-603
- [11] Giuseppe Portaluri, Stefano Giordano, Dzmityr Kliazovich, Bernab e Dorronsoro, “A Power Efficient Genetic Algorithm for Resource Allocation in Cloud Computing Data Centers”, 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), pp 58-63