ISSN:0975-9646

Performance Evaluation Of Quality Metrices Based Fault Predection Model For Replica Management In Cloud Computing

Ajitabh Mahalkari, Prof. Ritu Tondon
Department of Computer Science & Engineering
TRUBA college of Engineering and Technology,
Dewas bypass, Indore (M.P) -452001 India

***Abstract*— Cloud computing is an entirely new approach of computing that offers a non-traditional, but, shared computing paradigm for organizations and individuals. It provides a way to adopt information Technology and its features without huge expenditure on infrastructure and applications. Cloud computing provides multiple on-demand services accessible from a broad network with the ease of scalability even in shared environment so to achieve cost effectiveness.**

However, despite the potential gains of cloud computing still lack in providing 100% up-time and ensured data availability to the business organizations .These issues may be very critical for business process that they might even lose data. Taking this problem we tried to give a reliability and availability based fault tolerance approach based on replica distribution in previous work [14], In this work we are trying to evaluate the process by implementing the scheme and get the statistical performance analysis of the approach. In this work we shown the results that we could extract from it and the replica decision that could be preferable when actually this type of situations occurs. By the analysis of the result we get to know that the approach we gave in to the previous work[14] is useful for the cloud service providers to get rid of faulty conditions using the proposed fault tolerant policy.

***Keywords*— Include at least 5 keywords or phrases**

I. INTRODUCTION

Various cloud applications for fast process requires real time processing on remote devices and resources and causes more chances of fault occurrence due to unidentified latency issues and lesser control over computing nodes. It requires high reliability for each node before processing of requests. Thus a robust and vibrant system is requires which continuously manages a fault model and distributes the load between the nodes according to their reliability values. It applies a decision model before selection of nodes for processing through virtual machines. The above values is changed after ever computing cycles due to their behavior measures for handling the data and processing in a shared environment. It also follows the timely analysis which increases the reliability of the nodes. It is represented as minimum or maximum according to which the system applies protective and pre-emptive measures by using replicas distribution schemes. Management of nodes in a network will based upon the above reliability metric model. If a virtual machine manages to produce a correct result within the time limit, its reliability increases. And if it fails to produce the result within the time or correct result, its reliability decreases.

The applications and software’s are always developed in a parallel programming manner which provides better utilization of resources in a dynamic fashion. It offers dedicated resources and less computation time than traditional approaches. In cloud like environment applications required more computing resources and demands maximum memory and resource availability. But as it is a shared medium such resource availability is a key parameter for effective service delivery to the end user and hence the remote access will also be provided. Such remote access devices and memory for effective and dynamic processing causes various environmental parameters affecting this regularly. Hence the performance of such processes degrades due that access mechanism. Parallel tasking requires timely exchanges of messages for synchronous processing capability and hence increases for various fault situations. Such process involves message process intercommunication (MPI) for achieving parallelism in their execution.

Fault tolerance can be achieved by multiple error recovery techniques implemented at the application level. Such techniques lacks dynamic fault-tolerance and error-recovery mechanism that will allow for executions to recover from multiple failures precede execution or migrate seamlessly to another site in the event of unrecoverable failures. The behavior and performance of such applications vary with hardware, platform and network characteristics. These factors further limit scalability and lead to poor portability across infrastructure of the system and degradation of service applications and, high costs involvement.

A client coordinates with the service work provider to accomplish fault tolerance conduct for its applications. It makes the fault tolerant arrangement in view of the client end necessities such that a legitimate harmony between the accompanying factors is accomplished.

- Fault model: measures the granularity at which the fault tolerance arrangement must handle errors and failures in the framework. This factor is portrayed by the mechanisms connected to accomplish fault tolerance, robustness of disappointment detection protocols, and strength of failover granularity.

- Resource consumption: measures the sum and expense of resources that are obliged to understand a fault model. This factor is typically inborn with the granularity of the disappointment detection and recovery mechanisms regarding CPU, memory, bandwidth, I/O, etc.

- Performance: manages the effect of the fault tolerance procedure on the end-to-end quality of service (QoS) both amid disappointment and disappointment free periods. This effect is regularly portrayed utilizing fault detection latency, replica launch latency and disappointment recovery latency, and other application-dependent metrics, for example, bandwidth, latency, and loss.

II. BACKGROUND

Cloud computing is at the marketplace because of its huge medium of resources as far as server computation capability, massive storage, infrastructure sharing policies, etc. In this standard, combination of different technologies and components are made to achieve flawless exchanges of information with reduced managerial burdens. Hence, consumers had very little to add understanding of this framework for use and hence give the effective medium. To achieve its objectives cloud computing must give a safe and secure storage services for its clients. This can be given by utilizing fault tolerance mechanism by which more than one copy of information as far as replica is put away in different geographic locations around the globe at different servers. Before understanding the replica let us investigate the sorts of faults accessible to acquire.

○ Types of Faults

Types of faults may found listed below:

- *Network fault*: A Fault occur in a network due to network partition, Packet Loss, due to Packet corruption and losses, Node destination failure, Network link failure, etc.
- *Physical faults*: This type of Fault can occur in hardware involve in the system like fault in hardware peripherals CPUs, Fault in memory, Fault in storage, etc.
- *Media faults*: Fault occurs because of media node crashes.
- *Processor faults*: fault occurs in processor due to operating system crashes, etc.
- *Process faults*: A fault which occurs because of unavailability of resources required and software bugs, etc.
- *Service expiry fault*: The service time of a resource may expire while application is using it.

○ Categories of Faults

A fault can be categorized on the basis of computing resources and time. A failure occurs during computation on system resources can be classified as: omission failure, timing failure, response failure, and crash failure. Fault may be:

- **Permanent:** These failures occur by accidentally tearing of cables , power failures and other problems. These types of faults can be recovered with little effort and precautions reproduce these failures. These failures can cause major disruptions and some part of the system may not be functioning as desired.

- **Intermittent:** These are the failures appears occasionally. Mostly these failures are ignored while testing the system and only appear when the system goes into operation. Therefore, it is hard to predict the extent of damage these failures can bring to the system.

- **Transient:** These failures are caused by some inherent faults in the system. Although, recovery for these faults may be retrying roll back the system to previous state such as restarting software or resending a message. These failures are very common in computer systems.

○ Types of Fault Tolerance

Proactive fault tolerance: The Proactive fault tolerance policy is to predicting them in advance on the basis of theory , experience and best practices and proactively replace the suspected component means detects the problem before it actually come.

Reactive fault tolerance: Reactive fault tolerance policies reduce the effort of failures when the failure actually occurs and detected in the system. These types of fault tolerance policy and techniques provide robustness to a system.
Adaptive: All the procedure done automatically according to the situation.

III. LITERATURE REVIEW

Cloud computing is the balanced combination of various components and existing technologies which resulted as several services to the consumers. Application users of such services get increased day by day accordingly. Mainly the user is interested in backing up important data and accessing their data from different types of devices such as laptop, PDA, PC's, Mobiles, Tablets etc. Thus, effective cloud models are the need to full fill the market demands and especially when security of data is concerned

For expanding the reliability cloud services and applications some security and fault tolerance system are obliged which make the framework more strong. Amid the last few years different creators had exhibited their work for attaining their point towards fault tolerance and backup services helps. The creator of the paper [5] introduced a methodology of information escalated I/O for fault tolerance in nature's turf. The result is an architectural representation which offers fine grained information access control policies, substantial document offering apportion, high throughputs and concurrency of operations. The applicability of the result is checked in Open Nebula the earth. Giving a deep dive to the fault tolerance instrumental approaches, the writer of the paper [6] does the enhancements for results extraction. The work utilizes different burden adjusting techniques for cloud movement, expansion which diminishes the assets burden till the faulty gadget or segment is supplanted. The paper likewise gives a study on different matrix and dispersed environment on the fundamental of burden, quality and security.

Presently considering the different applicability zones where the fault can happen makes the result advancement all the more simple and compelling. In a manner to accomplish this virtualization fault tolerance component is given in the paper [7]. The paper viably investigates the machine portability by which the fault can abuses the cloud adaptability for operations. The paper is centering its intension towards creating the result which is equipped for self overcoming the fault conditions. It might be made possible by utilizing autonomic computing. It requires powerful replica and backup administration for fault localisation and evacuation. Some more replica duplicate and stacking plans are displayed in the work given in paper [8]. It is an auto oversaw key-quality store pool which rapidly utilized for distributing or appointing the assets to the cloud or clients information. It additionally underpins the movement in a versatile way. An arrangement is additionally produced by the methodology as indicated by which no backup or recovery might be made in harmony conditions.

A percentage of the papers had concentrated on security and security all the while. In approach to accomplish the objectives, this paper [9] presents a Cloudfit model for intrusion tolerant framework. The result is equipped for taking a shot at nature. The methodology takes a shot at BFT calculation by which more security policies could be connected. The calculation is connected infusion of a recovery based result which makes the framework fault tolerant and secure against intrusions. A portion of the creators had just worked with components of cloud performance, for example, expense, power, asset usage, offering, response time and so forth for identifying the fault conditions. In the wake of recognizing the fault and its writers the tolerance or overcoming system as replication and backup is taken. The paper [10] shows and novel asset portion and employment booking methodology for powerfully treatment of burdens and asset imparting. The recommended result in the paper upholds the heterogeneity focused around metric shares which enhances its performance. At the end some confirmation of the viability of the methodology is likewise given in the paper. The paper [11] investigates the impact of failures on applications and clients' information before applying any fault tolerant systems. The point is towards expanding the trust over the framework with high information conveyance rates and failure tolerant frameworks in any rising circumstances.

Presently the framework conduct is constantly changing alongside the classification of faults and thus the plan of tolerance is connected in the wake of comprehending the nature of the issue. Hence, the paper [12] presents an examination of the faults at distinctive circumstances by which their nature expectation might be made which helps in selecting the most suitable replica policies for backups. Mostly the paper had given with an investigation on some cloud segments, for example, server parts , systems and force circulations. It likewise measures the effect of fault on every individual segment and then outlines a component to determine the current issues and gives the complete reliability over the framework. Proceeding with the above work some more performance based fault assessment plans are given to the paper [13]. The paper creates wellbeing discriminating frameworks and gives high reliability utilizing virtual machines. The recommended plan is focused around a characterized rating framework for refashioning and adjusting the fault tolerance. The paper gives a continuous applications particular work of recovery and backups frameworks. Essentially here the framework gives both further and rearward recovery. The principle center here is a versatile performance of the transforming hubs and end or expansion of the hubs on the premise of the reliability.

Subsequently, from the above literature, it is unmistakably recognized that the current fault tolerance technique in cloud computing considers different parameter. The parameters are similar to their kind of fault tolerance (proactive, responsive and versatile), performance, response-time, scalability, throughput, reliability, availability, usability, security and related over-head.

IV. PROBLEM DOMAIN

To analyse the problem related to the effective fault tolerance mechanism various research articles has been studied after which following problem scenarios is analysed:

- In cloud computing consistent view of the resources is not monitored which lacks the actual condition & hence in case of faults heavy data losses or data availability reduction occurs.
- Centralized resource manager is not identified which shares the distributed load information for accurate analysis underutilized and over-utilized components.
- Decision making related to fault tolerance or replication scheme is not a proper boundary lined and hence causes incorrect decision of unmatched strategy with respect to occurred fault.
- Fault tolerant strategies in not matched up with clients' requirement.
- A common scheme for both proactive and reactive fault is not available.
- Generalized fault strategies are designed which causes uneven behavior at the time of new or variable faults.
- Fault model didn't take actual network condition before applying the replication schemes.

V. OBJECTIVES

- To increase availability of data in high & a low end system with scaled data characteristics.
- To analyze the actual component condition and take effective decisions accordingly.
- To measure dynamically changing loads & differentiated availability per application in lower overheads.
- To provide optimal resource allocation which matches the client’s requirements?
- To develop a common system for providing proactive & Reactive fault tolerance.
- Cost effective Query Processing & Storage Requirements

VI. SOLUTION DOMAIN

The proposed solution is used to overcome the shortcomings of distributed computing in a cloud environment. In this given architecture the cloud statics pay the crucial role while developing a fault tolerant system. According to the solution the fault tolerant strategies will acquire the dynamic properties of distribution and retrieval. Such properties can be provided as essential characteristics by using various current system details.[14] In a cloud environment there are mainly four essential requirements: Service Configuration, Shared Resources, Broad Network Access and Elastically extended device support. All this configuration policy will serve as an input for the proposed fault tolerant system. These configuration settings will effectively calculated by using various metrics. Among all of them this work concentrates on five types of metrics to measure the complete cloud behavior. These metrics are: Performance, Response Time, Throughput, Availability and Overhead Associated.

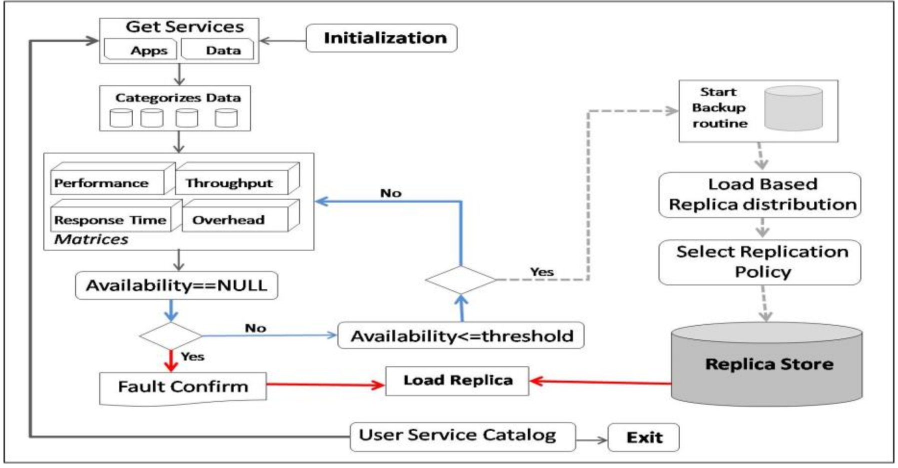


Figure 1: Design Architecture of Proposed Fault Tolerance Model

From the above mentioned metrics availability is calculated for taking the accurate & timely decision of taking backups as a fault tolerance mechanism. These decisions are based on various environmental factors for its accuracy & real time behavior. At average condition the normal value range of these metrics is defined and called as threshold. So if the availability of data is les then a specified threshold limits then reliability of each component related to backups is calculated. In this calculation load values at each component is taken as a base factor. From this the underutilized & over-utilized components is identified. According to that the component having low loads will have more chances to participate in replica distribution and the component having high load will remove from replica distribution device list.[14]

Distribution of replica is also a dynamic decision problem. In this work effective decision is taken which selects the best available replica distribution strategy from semi-active, semi-passive and passive. All the three used for different distribution conditions. Once the distribution is made it will store its copy on a replicated data store whose backup is also available.

Another case is when the availability is nil (Zero). Now in this case the system takes this condition as a fault point. Thus the reload scheme works here to provide the maximum availability. It accesses the stored replica from replica store & let it be loaded for further action. Here the reliability value is also stored in data store so as to decrease the calculation time for the same component. It stores the repeat components reliability values. So by taking the above architecture effective fault tolerance scheme can be implemented to assure zero data losses & maximum availability of client data. The above proposed scheme is capable of providing both type of fault tolerance: Reactive & Proactive.

The Suggested Metrics are given as:

- *Performance*– This is used to check the efficiency of the system. It has to be improved at a reasonable cost e.g. reduces response time while keeping acceptable delays.
- *Response Time* - is the amount of time taken to respond by a particular algorithm. This parameter should be minimized.
- *Throughput*–This is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.
- *Availability*-- The probability that an item will operate satisfactorily at a given point with in time used under stated conditions. Availability of a system is typically measured as a factor of its reliability as reliability increases, so does availability.
- *Overhead Associated*-- determines the amount of overhead involved while implementing a fault tolerance algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a fault tolerance technique can work efficiently

VII. RESULT EVALUATION

The suggested approach serves its primary objective of providing the decisional replica management and distribution within some given backup primitive. These primitives work towards analyzing the operations of the developed tool. Working on the tool doesn't means it work well in all the situations and needs to be extensively tested. For measuring the performance of the system, certain criteria specific factors are selected for a unit limit. These values can be capture prior for their actual implementation known as performance analysis. The process of capturing the record after the single or multiple executions are known as performance measurement. This paper covers the aspect of the measurement with some predefined performance monitoring factors which analyses the complete process and direct their usages for futuristic purposes. The result is visualized by using some tables and graphs having row representing the record of single execution and column showing the parameters on which the value is measured. After getting all the record the visualization shows the effectiveness of the approach. Here the parameters are CPU Utilization, RAM Utilization, Page Faults, Bandwidth Speed (Upload and Download), and Fault Detection Condition. On the basis of these initial primitive the later factors are calculated which actually guides or directs the systems performance. These later analysis factors are:

(i) *Performance*
(ii) *Response Time*
(iii) *Throughput*
(iv) *Overhead Associated*
(v) *Availability*
(vi) *Replica Decision*

For getting the values on the above heads, the values are taken under some specific test bed or environments. These constant environment conditions and inputs are abbreviated here as test case or scenarios. There are 12 cases used in this work for getting in depth analysis of the approach on performance factors. The evaluation is robust hence the tools behavior is measured with different orientations and a complete analysis is achieved.

S. No.	Scenario (cases)	CPU Utilization	RAM Utilization	Page Faults	File Size (bytes)	Required No. Pkts.	Upload Spd.(kbps)	Download Spd.(kbps)	Fault Condition
1.	Case1	9%	40%	51	181473	124	69.46	8.65	No Fault
2.	Case2	3%	27%	50	181473	124	60.24	6.42	Fault
3.	Case3	3%	12%	31	181473	124	33.38	5.06	No Availability
4.	Case4	6%	41%	47	1055017	722	28.35	4.74	No Fault
5.	Case5	4%	26%	40	1055017	722	23.50	6.60	Fault
6.	Case6	1%	13%	27	1055017	722	25.31	1.29	No Availability
7.	Case7	10%	43%	46	14350057	9828	55.26	9.59	No Fault
8.	Case8	6%	27%	39	14350057	9828	55.05	6.56	Fault
9.	Case9	2%	14%	26	14350057	9828	24.16	5.82	No Availability
10.	Case10	10%	41%	46	31780	21	35.01	10.49	No Fault
11.	Case11	3%	26%	39	31780	21	52.61	3.40	Fault
12.	Case12	5%	13%	26	31780	21	25.26	3.38	No Availability

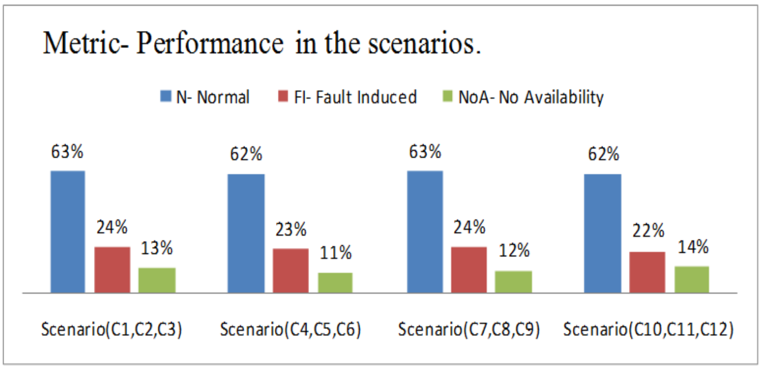
Table 01: Below table showing different case scenarios with different system attributes. According to the we took decision of fault occurrence.

Table Summary: The above table covers the all aspect how the replica can be taken in backup systems. The decision on the basis of which the system predicts the faults is majorly known as availability conditions. These conditions depend on the different resource constraints that are currently occupied and that are required. If there is mismatch between both there is an associated probability that the data availability can be loosed. The above factors analyses it on the basis of given primitives. As it was clearly seen by the table that the how the system is effectively detecting the presence of fault or forecasting its future presence. The above factors are taken on fixed system constraints known as cases.

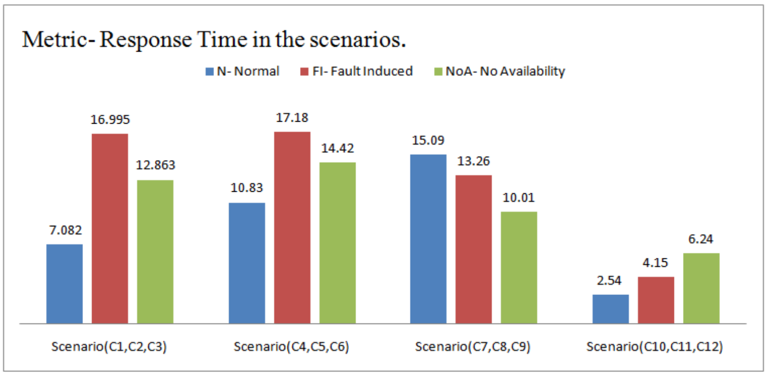
S. No.	Scenario (cases)	Mode	Performance (PF.)	Response Time (RT.) Sec.	Through-put (TP.)	Overhead Associated (OA)	Availability (AVL.)	Replica Decision
1.	Case1	N	63%	7.082	8.623	2.73%	91.353	No need
2.	Case2	FI	24%	16.995	6.428	2.73%	37.408	Create Replica
3.	Case3	NoA	13%	12.863	3.60	2.73%	29.83	Load Replica
4.	Case4	N	62%	10.83	9.74	2.74%	74.48	No need
5.	Case5	FI	23%	17.18	60.6	2.74%	36.595	Create Replica
6.	Case6	NoA	11%	14.42	3.29	2.74%	22.94	Load Replica
7.	Case7	N	63%	15.09	9.50	2.74%	75.249	No need
8.	Case8	FI	24%	13.26	6.56	2.74%	37.55	Create Replica
9.	Case9	NoA	12%	10.01	3.28	2.74%	29.063	Load Replica
10.	Case10	N	62%	2.54	10.23	2.64%	77.136	No need
11.	Case11	FI	22%	4.15	3.41	2.64%	32.29	Create Replica
12.	Case12	NoA	14%	6.24	4.40	2.64%	21.73	Load Replica

*N- Normal Mode, *FI- Fault Induced, *NoA- No Availability

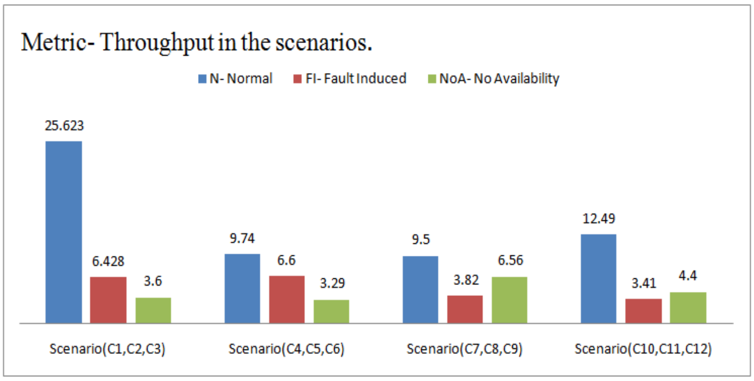
Table 02: Table showing the record of five metrics values on which availability of data depend on. Replica decisions can be taken on the basis of availability.



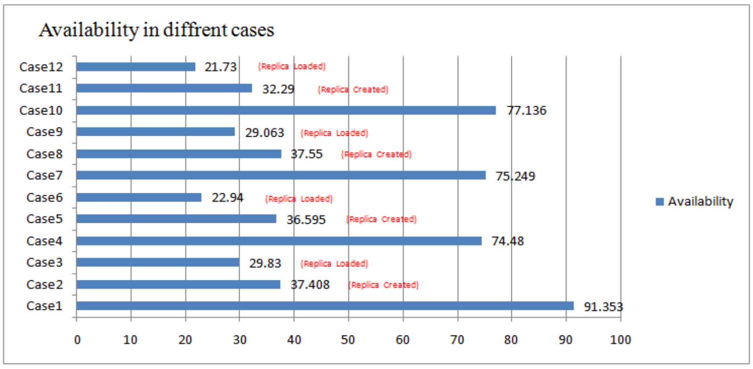
Graph01: Showing Performance in the different scenarios.



Graph02: Showing Response Time in the different scenarios.



Graph03: Showing Throughput in the different scenarios.



Graph04: Showing Availability in the different cases.

VII. EVALUATED BENEFITS

The proposed work will provide the effective decision making regarding the replica distribution as a fault tolerance policy which dynamically takes the updates form current conditions. At the result evaluation level of our work following benefits is identified which definitely proves their accuracy & effectiveness in near future of approach implementation[14].Overhead related to system performance & cost is reduced. Dynamic changes are incorporated in the replica distribution scheme selection for accurate decision which increases system throughput.

VIII. CONCLUSION

The approach of fault tolerance is very important to be incorporate in the service agreements to ensure effectiveness of services delivered to the customers. We tried to understand the situations when fault can occur into the system along with types of faults and its remedial actions. We found that still there is a need of a effective fault prediction technique so as to take appropriate action against it proactively and reactively both. We tried to simulate the situations when fault can occur on the basis of the performance matrices, We also tried to get the solution approaches with the help of replication creation. We have chosen replica approach so as to give assurance to data at high level. We could get the statically results above shown and could conclude that, the approach we are proposing in work[14] is achievable and can be implemented after some more improvements in accordance with the results we evaluated above.

REFERENCES

[1] Ravi Jhawar, Vincenzo Piuri and Marco Santambrogio, “Fault Tolerance Management in Cloud Computing: A System-Level Perspective”, in IEEE Transaction, ISSN: 1932-8184,doi: 10.1109/JSYST.2012.2221934,2012.

[2] Dinesh Rajan, Anthony Canino, Jesus A Izaguirre, and Douglas Thain, “Converting A High Performance Application to an Elastic Cloud Application”, in Department of Computer Science and Engineering University of Notre Dame , Indiana

[3] Sheheryar Malik and Fabrice Huet, “Adaptive Fault Tolerance in Real Time Cloud Computing”, IEEE World Congress, ISSN: 978-0-7695-4461-8/11, DOI 10.1109/SERVICES.2011.108, 2011.

[4] Wenbing Zhao, P. M. Melliar-Smith and L. E. Moser, “Fault Tolerance Middleware for Cloud Computing”, in IEEE 3rd International Conference on Cloud Computing, ISSN: 978-0-7695-4130-3/10, DOI 10.1109/CLOUD.2010.26, 2010.

[5] Houssem-Eddine Chihoub, Gabriel Antoniu and Maria S. Perez-Hernandez, “Towards a scalable, fault-tolerant, self-adaptive storage for the clouds”, in INRIA, Rennes - Bretagne Atlantique, France

[6] Akanksha Chandola Anthwal and Nipur, “ Survey of Fault Tolerance Policy for Load Balancing Scheme in Distributed Computing”, in International Journal of Computer Applications , ISSN:0975 – 8887, Volume 74– No.15, July 2013

[7] Alain Tchana, Laurent Broto and Daniel Hagimont, “Fault Tolerant Approaches in Cloud Computing Infrastructures”, in ICAS 2012 :The Eighth International Conference on Autonomic and Autonomous Systems, IARIA Journal, ISBN: 978-1-61208-187-8,2012.

[8] Nicolas Bonvin, Thanasis G. Papaioannou and Karl Aberer, “A Self-Organized, Fault-Tolerant and Scalable Replication Scheme for Cloud Storage”, in ACM Journal, doi: 978-1-4503-0036-0/10/06, 2010.

[9] Hans P. Reiser, “Byzantine Fault Tolerance for the Cloud”, in University of Lisbon Faculty of Science, Portugal, at http://cloudfit.di.fc.ul.pt

[10] Gunho Lee, Byung-Gon Chun and Randy H. Katz, “Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud”, in University of California, Berkeley.

[11] Ravi Jhawar and Vincenzo Piuri , “ Fault Tolerance and Resilience in Cloud Computing Environments”, in Computer and Information Security Handbook,2nd Edition Morgan Kauffman, ISBN:978-0-1239-4397-2,2013.

[12] Ravi Jhawar and Vincenzo Piuri, “Fault Tolerance Management in IaaS Clouds”, in IEEE Transaction on Cloud Computing, 2012.

[13] Shivam Nagpal and Parveen Kumar , “A Study on Adaptive Fault Tolerance in Real Time Cloud Computing”, in International Journal of Advanced Research in Computer Science and Software Engineering (IJarcsse), ISSN: 2277 128X, Volume 3, Issue 3, March 2013.

[14] Ajitabh Mahalkari, Prof. Ritu Tondon " A Replica Distribution Based Fault Tolerance Management For Cloud Computing" International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6880-6887.