ISSN:0975-9646

Sapna Engle et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5497-5502

RAI-DD: Reliability, Availability Identification & Dynamic Decision Based Replica Distribution for Cloud Computing

Sapna Engle
Dept. of Information Technology
SVITS
Indore, India

Gaurav Shriwastava
Dept. of Information Technology
SVITS
Indore, India

Abstract: **Cloud computing is the technology based extension for distributed and grid computing. Its main aim is to share the resources in an effective and efficient manner. Cloud technology follows various characteristics for such improved sharing like utility and autonomic computing. Autonomic computing means fault tolerant and self-recoverable system. Its services are built something like virtualization of computing power economically available to large number of users. As the scalability and autonomic computing increases with more shared resources with higher degree of communication, the node failures compare to conventional systems is increased. Thus, new tools and approaches are needed to build reliable and robust systems. In this work a new RAI-DD (Reliability, Availability Identification & Dynamic Decision) based replica distribution architecture is proposed to better utilize associated scalability of computing cloud and to provide client transparent novel fault tolerant system for various cloud applications. At the initial level of work the approach is proving its efficiency by high scalability and dynamic load balancing. At the time of uneven and dropdown time the system sustains the fault tolerance and in all condition low overhead is desired.**

Index Terms**:** Cloud Computing, Fault Tolerance, Replica Distribution, Metrics Oriented, Availability, RAI-DD (Reliability, Availability Identification & Dynamic Decision)

I. INTRODUCTION

Cloud applications is used for real time remote based access environment in which the services is provided to consumer by using various paradigm of distributed, grid, autonomic and utility computing. In such shared network multiple technology integration is used whose management is handled by the intermediately brokers and agents. They usually create a communications between the hosts and the provider and guarantees it's for the successful exchanges of the data. This vibrant environment of different resources with frequent changes causes more chances of occurring fault due to unidentified latency issues and lesser control over computing nodes.

In such environment the reliability should be more concerning the users trust over these systems for processing. For overcoming the above issues some fault tolerance models needs to be provided in support of the existing mechanisms. These faults models distribute the load among the various nodes and processing systems according to their reliability values. It also takes the continuous backups based on decisions of occurrence of fault to avoid certain data losses. The process of assigning the reliability values changes after every computing cycle and measured by behaviour analysis of node for processing and sharing of data. This process is a time based approach in which the minimum and maximum reliability decides the way of using backups and its ratio. It means by the minimum values of reliability the pre-emptive approaches are used and for maximum values the protective mechanisms are used. For achieving the above fault tolerance various mechanisms is used out of which the replica distribution is the major one. In this the replica is continuously exchanges between the different locations and devices. These replicas are transferred from one to other locations and hence require a message communication between the various copies of the same data. 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 with 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 behaviour and performance of such applications vary with hardware, platform and network characteristics. These factors further limit scalability and lead to poor portability across platforms and, high development and deployment costs.

A client coordinates with the dedicated service provider to achieve fault tolerance behaviour for its applications. It creates the fault tolerant solution based on the client end requirements such that a proper balance between the following factors is achieved.

- *Fault model:* measures the granularity at which the fault tolerance solution must handle errors and failures in the system. This factor is characterized by the mechanisms applied to achieve fault tolerance, robustness of failure detection protocols, and strength of fail-over granularity.
- *Resource consumption:* measures the amount and cost of resources that are required to realize a fault model. This factor is normally inherent with the granularity of the failure detection and recovery mechanisms in terms of CPU, memory, bandwidth, I/O, and so on.

- *Performance:* deals with the impact of the fault tolerance procedure on the end-to-end quality of service (QoS) both during failure and failure-free periods. This impact is often characterized using fault detection latency, replica launch latency and failure recovery latency, and other application-dependent metrics such as bandwidth, latency, and loss rate.

II. BACKGROUND

Cloud computing is gaining pace due to its shared medium of resources in terms of their computation power, storage, infrastructure etc. In this paradigm, integration of various technologies and components are made to achieve flawless exchanges of information with less managerial loads. Thus consumers had not much to gain understanding of this system for usage and hence provide the effective medium. To achieve its goals cloud computing must provide a safe and secure storage feeling for its users. This can be provided by using fault tolerance mechanism by which more than one copy of data in terms of replica is stored in different geographic locations. Before understanding the replica let us take a look over the types of faults available to pre-empt. These are:

- *Proactive fault tolerance:* The Proactive fault tolerance policy is to avoid recovery from fault, errors and failure by predicting them and proactively replace the suspected component means detect the problem before it actually come.
- *Reactive fault tolerance:* Reactive fault tolerance policies reduce the effort of failures when the failure effectively occurs. This technique provides robustness to a system.
- *Adaptive:* All the procedure done automatically according to the situation.

The above faults can affect the environment in different situations all it needs to make the system which overcomes from any situation. For defending any of the above faults category various mechanism is been suggested over the last few year. Among them replica based schemes are giving their strong presence. Now for designing and improved fault tolerance mechanism this schemes needs to be clearly studied. So the schemes are given here as:

- *Semi-active replication:*
  In this scheme the input can be given to any of the existing replica and if some modifications is made only in primary replica than it should be reflected in all simultaneously. In this both the primary and secondary replicas generates the outputs but only primary replica is available for the end user. If the primary replica fails to load, than immediately the secondary replica loads to the memory.
- *Semi-passive replication:*
  In this scheme the regular check points are made for time based updates and modifications. These check points having primary replica and creates the buffer between the each check points. This buffer information with its state is regularly transferred between several backups. It does not execute the instruction but saves the latest state of primary replica. If the primary replica fails than, the secondary replica loads in to the system.
- *Passive replication:* In this scheme the state information is regularly stored on the backups on an offline based mode. In case of failure this passive backup copy of replica is loaded to newly started VM instance. Here the backup replica is configured for a specific application or can store the different instances of VM. Both the strategy is operated in offline mode.

Third function is interface for location measurement which is must for users operating or using such location information or updates. This unit passes the measured information to users in an application specific formats. This information may be passed to a user or some other node for further detection of positions and its updates. Thus, the most feasible way to implement cooperative localization is by making each node share its own position estimate. Additional to the position estimate, it is important for other nodes to have a measure of confidence of their position estimate since they may use it in their own position estimation process.

III. LITERATURE SURVEY

As cloud computing model is getting popularity day by day its user's quantity is also abruptly raising with certain particularities. These can be taken as its limitations with frequent changes in service orientations. It is mainly based on browser dependent programming and hence to provide backups and recovery changes needs to be made in existing system. To increase the reliability of end users in such system it has to provide fault tolerance behaviours. In process of that various authors had presented their work during the last few years among which some of the work is taken as literature for understanding the nature of cloud fault tolerance. These are:

In the paper [5], a data intensive I/O operation based fault tolerance is presented using a scalable architecture. The work is focusing their intensions over the working area of Open Nebula. Experimental instances design of scalable architecture offers a large file sharing and fine grained data access control with high throughput and concurrency. The approach takes the issues of fault tolerance, scalability and adaptability for improving the cloud environment. Later on some more approaches were presented to solve the fault tolerance based on load balancing scheme for distributed environment [6]. According to that the load balancing techniques are classified as static and dynamic according to their traffic diversification. A brief survey is also given here for comparing the various schemes on grid and distributed environment. Mainly through load balancing some proactive technique of fault tolerance provides quality of service value with the increase of demand of resources on cloud for vital applications.

Some of the papers take virtualizations as a problem area for applying fault tolerance. It uses machine portability option by which it exploits adaptability. Cloud computing platform implies splitting into three layers: hosts, virtual machines and applications. The paper [7] focuses on fault tolerance in cloud computing platforms using autonomic repair in case of faults. Here autonomic means self-correcting mechanism by the service. After which if a fault occurs the system is capable of removing the fault behaviour and when required can loads the correct replica from the backup system. Experimental evaluation of the suggested approach shows that the approach is capable of removing cooperative faults from the system.

This paper [8], proposes an auto-managed key-value store pool that dynamically allocates the resources of a data cloud to several applications in a cost efficient way. The suggested approach is well managed and likely to load the replica copy in failures situations dynamically and maintains the availability. It uses data partition later on used for data migration. A policy is also generated by the approach according to which no backup or recovery can be made in equilibrium conditions. Finally, we have implemented a fully working prototype of our approach that clearly demonstrates its applicability in real settings.

In the paper [9], the author suggest CloudFIT model in which a architecture for intrusion tolerant applications is deployed dynamically in the cloud. It also explore to what extent existing BFT algorithms can be used for increasing security and availability in the proposed architecture and what issues still need to be resolved in the future. Having the right abstraction will make it easier to argue about the correctness of proactive recovery algorithms, and also apply a recovery strategy to multiple BFT algorithms. Ideally, the same specification of the recovery component can be used for multiple BFT algorithms.

To further improve the cost factors and performance of cloud data analytics load distribution can be taken as critical issue. Thus some policies which effectively analyses the underutilized and over utilized nodes is required by the providers. In the paper [10], author suggested some novel resource allocation strategies and job scheduling among the various jobs in cloud cluster. The architecture suggested by the paper supports the heterogeneity based on metric shares which improves its performance. The aim to share the resources as per the dynamic requirements and provides the availability of the devices. At the end some proof of effectiveness of the approach is also given in the paper.

Carrying on the above mechanism fault tolerance delivers the availability of the services and let the trust factor and reliability increased for the end users. Some of the articles had also focuses on characterizing the recurrent failures in a typical Cloud computing environment [11]. They analyse the overall effects of failures on user's applications and provide the correct mechanism to overcome the appeared fault. Concern is towards high delivery rates of data and failure tolerance in case of any disaster or uncertain conditions.

Some of the researcher had identified that for effective fault tolerance scheme nature of the faults have to be understand previously. For this various cloud components and their behaviour in different situations is measured and after which some policies or replication schemes can be designed. Impact of individual failure related to a specific application will helps in improving the current solutions. The paper [12] presented a failure model containing the analysis of cloud infrastructure components like server components (VMM's), networks and power distributions. It also measures the impact of fault on each individual components and then designs a mechanism to resolve the existing issues and provides the complete reliability over the system.

In the paper [13] a reliability based model is suggest for improving the performance of existing fault tolerance mechanisms. The approach reflects the real time behavior for safety critical systems and provides high reliability for processing components through virtual machine for the cloud. The model is based on certain rating system of refashioning adaptive fault tolerance in real time cloud computing. If a virtual machine produces correct result and on time then its reliability increases otherwise vice versa. But if any of the nodes does not achieve the level than backward recovery is performed by the system. Basically here the system provides both forward and backward recovery. The main focus here is adaptive behavior of the processing nodes and removal or addition of the nodes on the basis of the reliability.

Thus from the above literature it is clearly identified that the existing fault tolerance technique in cloud computing consider various parameter. The parameters are like there type of fault tolerance (proactive, reactive and adaptive), performance, response-time, scalability, throughput, reliability, availability, usability, security and associated over-head.

IV. PROBLEM STATEMENT

Fault tolerance is the factor which provides reliable data services to the end users in case of critical situations and increases trust on the service. Several fault tolerance schemes proposed over the last few years. Mainly the schemes are data backup oriented in normal system operating conditions and do recover the same in case of failures. Thus the approach is purely based on multiple copies of same dat. There are various performance and replications issues which need to be solved for an improved system. Out of all there are some identified issues for this work to proceeds are given here as:

1) In cloud computing consistent view of resources is not monitored which lacks the actual condition & hence in case of faults heavy data losses or data availability reduction occurs.
2) Centralized resource manager is not identified which shares the distributed load information for accurate analysis underutilized and over-utilized components.
3) Decision making related to fault tolerance or replication scheme is not properly boundary lined and hence causes incorrect decision of unmatched strategy with respect to occurred fault.
4) Fault tolerant strategies in not matched up with clients requirement.
5) Common scheme for both proactive and reactive fault is not available.

6) Generalized fault strategies is designed which causes uneven behaviour at the time of new or variable faults.
7) Fault model not takes actual network condition before applying the replication schemes.

Thus this work aims to improve availability, scalability and reliability by taking the effective timely decisions of generating, distributing and loading replicas is different situations. It also focuses on optimal resource consumption and query processing for developing a novel improved fault tolerance mechanism.

V. Proposed RAI-DD Approach

The proposed solution is used to overcome the shortcomings of distributed computing in cloud environment. In this given architecture the cloud statics pays 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. In cloud environment there are mainly four essential requirements: Service Configuration, Shared Resources, Broad Network Access and Elastically extended device support. All this configuration policies 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.

From the above mentioned metrics availability is calculated for taking the accurate & timely decision of taking backups as 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.

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 replica data store whose backup is also available.

Another case is when 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.

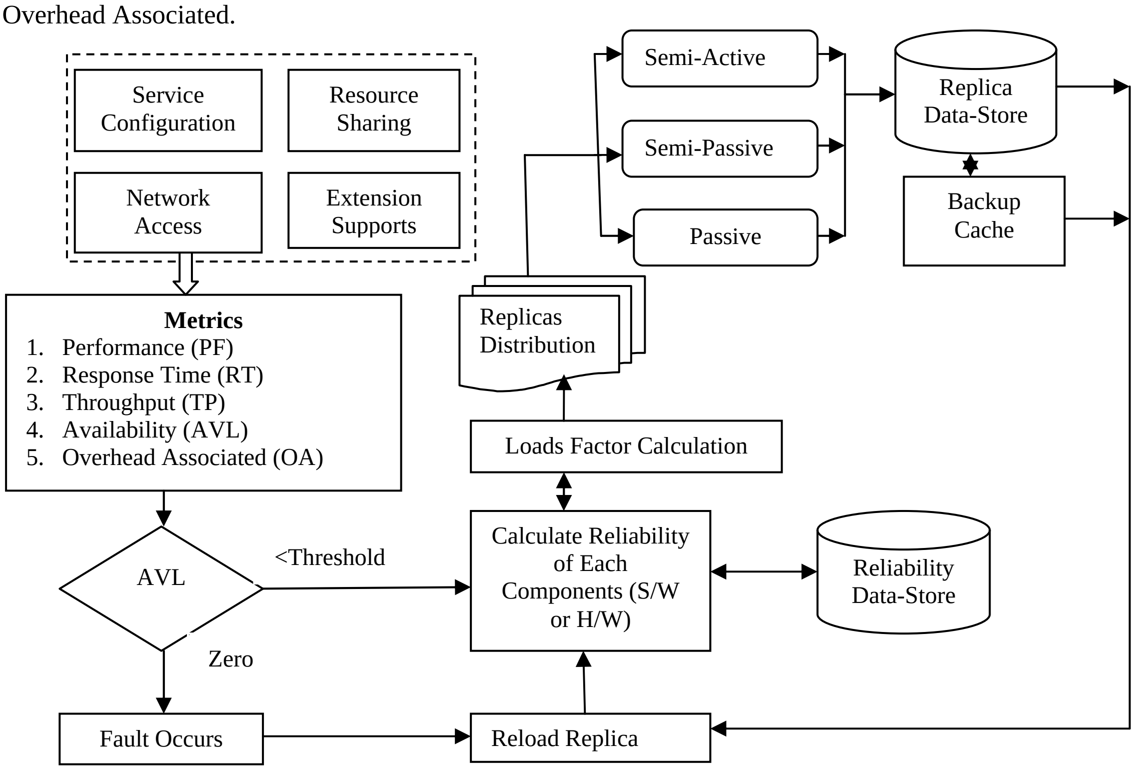

Figure 1: Design Architecture of Proposed RAI-DD Fault Tolerance Model

So by taking the above architecture effective fault tolerance scheme can be implemented. It assures zero data losses & maximum availability of client data. The above proposed scheme is capable of providing both type of fault tolerance: Reactive & Proactive. The proposed system is used to take the decisions related to fault occurrence effectively and on time. The system is separated in five major components:

Module 1: Current Condition Assessment (Metrics)
Module 2: Fault Tolerance Based on Availability
a. Reactive
b. Proactive
Module 3: Fault Tolerance Replication Scheme Selection
Module 4: Calculations of Reliability Values for Each component
Module 5: Backups & Restore Data Stores

From the above characterized modules of proposed system it can be clearly seen that the detection & removal of fault mainly depends upon the components of cloud environment. As the component can be of any type like software or hardware so detection of unwanted behaviour related to availability of data must be real time or dynamic. Decisions related to the replication scheme can only be applied after assigning & measuring actual conditions. Fault tolerant scheme can only be applied after measuring this real time values. Also the kind of replication scheme used as fault tolerance mechanism will also generates their local data whose further backup is taken so in near future if the same components are repeated with similar configuration then the system overhead of calculation is reduced.

Application Domain

In this work, main focus is on fault tolerance policy based replica distribution algorithm & their selection decision making. Thus this work identifies the common load balancing techniques in cloud computing and further investigates techniques having fault tolerance provision in replica distribution scheme. In near future it also includes some approaches implemented to grid computing as both are type of distributed computing.

By comparing the techniques on different metric and tried to find the scope for improving fault tolerance policy in load balancing schemes. In near future research could be conducted on development of load balancing algorithm for cloud, taking in account fault management and also minimizing migration time of job in case of failure of node occurs and further guaranteeing optimal performance of system. More load balancing algorithm could be developed which take into account proactive technique of fault tolerance in cloud computing for enhancing the efficiency and providing quality of service value with the increase of demand of resources on cloud for vital applications. There are few of the applications where fault tolerance can be effectively utilized is as follows:

1. Infrastructure Management: (Amazon Web Services, Google App Engine)
2. Workload Distribution & Access Control: (Role Base Access, Implicit Authentication, VeriSign)
3. Service Agreement Monitoring: (Nimbus, Open Nebula, Social Networking)
4. Real Time Systems: (Satellite Image Processing, Gmaps)

Some other applications are: IP Monitoring, Storage Solutions, User Management, User Role Based Access Control and Network Configuration with expansion.

VI. Performance Evaluation

RMSE (Root Mean Square Error): This parameter will serve to detect the performance of location estimation in terms of accuracy near by the actual position and identified position. If the difference is less means the approach will serve its aim.

Computational Overhead: To identify the location updates frequently the number of steps in processing will increases the burden of computation on each node which reduces the battery life. This will be again a performance parameter.

Time of Detection: This is the time taken to detect the position changes in the nodes locations. If the time of information is near to real time than the approach will proves its effectiveness.

Localization Error: It is the cumulative values which depend upon the above identified parameters. Localization error must be reduced with number of phases of approach and increased nodes and sensor in the network.

VII. Expected Outcome

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 initial level of our work following benefits is identified which definitely proves their accuracy & effectiveness in near future of approach implementation.

- Meta data schemes can be easily handled via application level feedbacks.
- Low overheads & real time processing provides pre-emptive approach for fault detection & avoidance.
- All the three categories of faults can be analysed & hence overall system availability is guaranteed.
- System & application level monitoring can be performed which causes real time analysis & operations.
- Data migration is fully performed with minimum probability of data losses.
- Overhead related to system performance & cost is reduced.
- Dynamic changes are incorporated in replica distribution scheme selection for accurate decision which increases system throughput.

VIII. Evaluation Parameters

The Suggested Metrics are given as:

(i) *Performance*– This is used to check the efficiency of the system. It has to be improved at a reasonable cost e.g. reduce response time while keeping acceptable delays.

(ii) *Response Time* - is the amount of time taken to respond by a particular algorithm. This parameter should be minimized.
(iii) *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.
(iv) *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.
(v) *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

IX. CONCLUSION

Now a days the technology is most frequently updating in the market and among them cloud computing is showing its strong presence. It is the combination of different technologies serving the user's needs to provide reliability and availability. Both the above areas require continuous delivery of the services in any environments even with the failures. Thus to achieve the goals several mechanism had been proposed to provide fault tolerance in the system and among most of them replica based schemes are common. This work suggests some modifications in existing schemes for further improvement in performance and reliability. The approach introduces novel Reliability, Availability Identification & Dynamic Decision (RAI-DD) Based Replica Distribution for Cloud Computing. At the initial level of results approach is proving its efficiency. Later on some real implementation will shows the same.

X. 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.