A Probabilistic Approach for Fault Analysis in Cloud Environment

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Software Engineering

Submitted By

Anu

(801331002)

Under the supervision of:

Dr. Anju Bala

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"A *Probabilistic Approach for Fault Analysis in Cloud Environment*"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Anju Bala* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Anu)

This is to certify that the above statement made by the candidate is correct and true to l best of my knowledge.

5/7/1

(Dr. Anju Bala)

Assistant Professor

Computer Science and Engineering Department

Countersigned by

(Dr. Deepak Garg)

Head

Computer Science and Engineering Department

Thapar University, Patiala

(Dr. S. S. Bhatia)

Dean (Academic Affairs)

Thapar University

Patiala

Acknowledgment

First of all I would like to thank the almighty God for his blessings and for driving me with faith, hope and strength in completing the research work.

I would like to express my deepest gratitude to my guide **Dr. Anju Bala**, Assistant Professor, Computer Science and Engineering Department, Thapar University for her great support, motivation, guidance and co-operation. I appreciate her knowledge and skills in my area. She guided me throughout the research work for understanding the research field and in getting to a solution to the problem identified.

I would like to thank **Dr. Deepak Garg**, Head of Department and **Dr. Damandeep Kaur**, P.G. Coordinator, Computer Science and Engineering Department, Thapar University for their inspiration and motivation to complete the task.

A special thanks to my parents and friends for their motivation and encouragement which went a long way in successful completion of my thesis.

Anu

Abstract

Cloud Computing has emerged as a revolutionary paradigm in information and communication technology. Prominence of this technology can be validated easily with its phenomenal features such as pricing-per-use, scalability and on demand availability of computing resource. But at the same time reliability and security are some of the current issues in this technology. Fault tolerance is an important approach to overcome these issues to improve overall performance of cloud computing services.

Fault tolerance is the ability of a system that ensures continuity of operations even in the presence of faults in its components to increase the reliability of the computing system. Virtualization of the instances in the cloud services is considered to provide computing instances anywhere according to the users demands. There can be some fault prone instances which are needed to be handled to minimize the fault occurrences and their adverse effects on the system. Fault handling approaches include prediction of fault occurrences in the system and their prevention to make sure fault-free execution of the computing tasks in the cloud services.

The proposed work presents a statistical analysis of virtual machines for finding fault occurrence probability to improve reliability. The status of the virtual machines is monitored periodically and faulty scenarios are understood with the monitored information to find criticality status of the machines. Statistical analysis for virtual machine is done based on a probability distribution model. An application interface is designed which is capable of finding availability, predicting fault occurrence probabilities and estimating sustainability of a virtual machine.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The aim of this chapter is to discuss the characteristics, evolution, service models, deployment models and issues of Cloud Computing. Introduction to fault tolerance is also given. In the end brief structure of the thesis is given.

1.1 Cloud Computing

Cloud Computing is a technology that has become a part of everyday life [1]. Revolution in Computer Science and Engineering can be easily seen with the development of the emerging trends of computing like Cloud Computing. Prominence of this technology can be validated easily with its phenomenal features as pricing-per-use, scalability and on demand availability of computing resources. Everyone in the industry has their own definition of Cloud Computing, a standardized definition for Cloud Computing has been given by the National Institute of Standards and Technology(NIST) "Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [2]. Cloud Computing is a descendent of technologies like Grid Computing, Utility Computing, and Service oriented architecture *etc*. [1].

1.2 Characteristics of Cloud Computing

The definition of Cloud Computing itself conveys its characteristi**cs**-" networked access to a pool of resources irrelevant of the location which can be virtualized to satisfy the scalability enhancement in business along with fault tolerant infrastructure and secure resource sharing".

Flexibility: Capability of automatic scaling up or off as per need without human interventions [4]. It provides access to the internet via electronic devices such as laptops, Smartphone *etc.* Due to this feature increasing business demands of high bandwidth can be easily fulfilled using remote servers.

Scalability: It is easy to add hardware devices with no modifications to the software infrastructure [4]. Multiple OS can run with the help of hypervisor or VM. Change in requirements can be easily satisfied as Cloud Computing is infinitely scalable.

Resource Pooling: A number of customers can be satisfied at the same time as service provider is providing number of computing resources clubbed together [3]. From customers perspective there are infinite number of resources available for them. From service provider's view a pool of resources is available which can be shared using virtualization.

On-demand access: Computing resources as well as the content stored in the Cloud can be accessed at any point of time from anywhere. Cloud Computing is location independent and time independent.

Pay-per-use: Similar to water and electricity bills Cloud Computing is payable as per its use. Pricing terms and conditions are settled prior to the resource provisioning with the consent of both service provider and customer. In Service Level Agreement (SLA) along with the resource leveraging, time enhancements, pricing norms are also mentioned.

In laymen language Cloud Computing can be defined as a technology which provides a pool of resources via internet for shared access and is based on a key concept of Virtualization and Server Consolidation.

1.3 Evolution

Initially when computers came into existence access to them was limited because of the infrastructure expenses and low functionality. However, with the increase in the computation power it has become easy and economical to access computing services and applications. The evolution of computing has followed a path which includes working on mainframes to distributed networked structures [5]. Cloud Computing began with a new concept which was introduced in the telecommunication industry in early nineties. Ramnath Chellappa introduced the term Cloud Computing [13] for the first time in a lecture talk delivered in a meeting for intermediaries in Cloud Computing. Virtualization which is the soul of Cloud Computing was actually used in Telecommunication Industries for the first time. When in place of point-to -point data circuits they started

using Virtual Private Network [6]. Cloud Computing follows the same concept and has covered servers and network infrastructure under the same approach *i.e.* Virtualization. It has not been a direct path to what we have today in Cloud Computing. It has emerged from already existing computing practices as shown in Figure 1.1. Similar to virtualization Cloud Computing has got its other features from utility computing, distributed computing, grid computing *etc.*

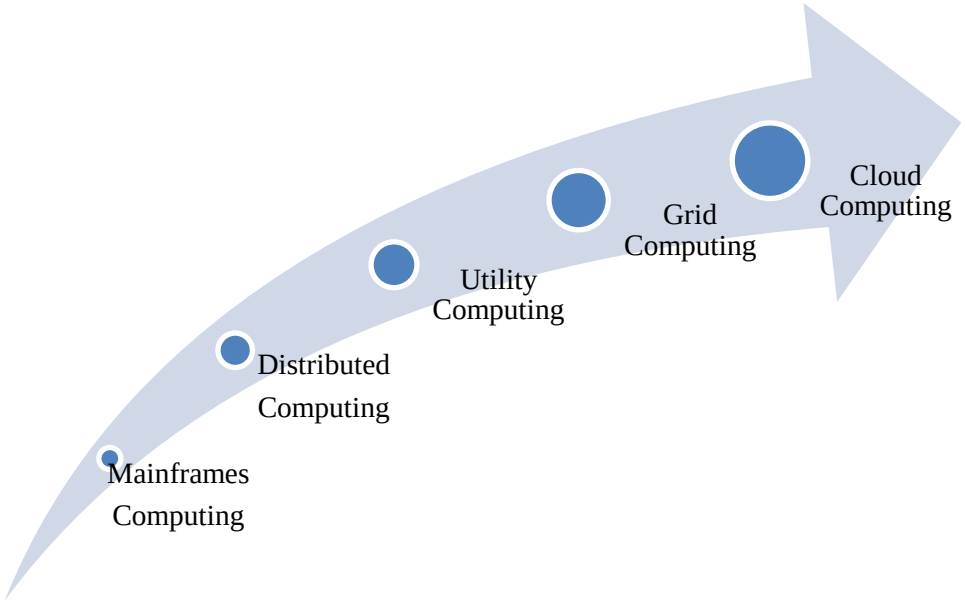

Figure 1.1: Evolution of Cloud Computing

1.4 Service Model

Cloud Computing is serving its customers with application, hardware and software as services deployed as its different service models [7]. Three service models are called by the name of the service they are providing, Infrastructure as a Service (IaaS), then the middle one Platform as a Service (PaaS) and lastly the top most from user's perspective Software or Application as a Service (SaaS or sometimes AaaS). The SLA (Service Level Agreement) determines the QoS (Quality of Services) demanded by the customers and provided by the service provider. Cost, duration and type of the service are specified in SLA [8].The service models are deployed as a layered architecture shown in the Figure 1.2

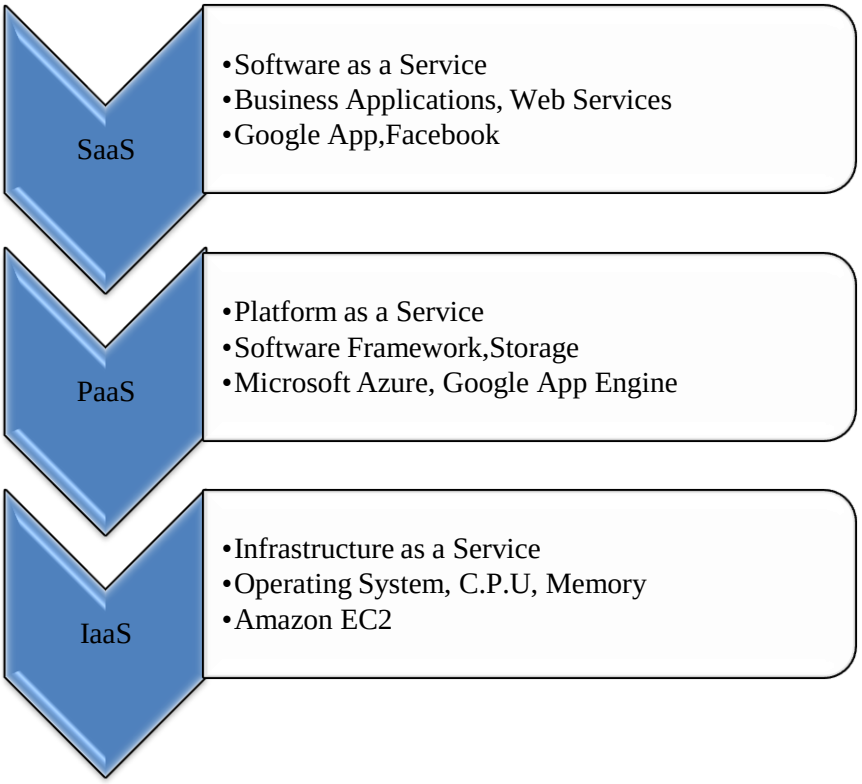

Figure 1.2: Service Model [10]

1.4.1 IaaS

Infrastructure as a service (IaaS) first layer is of the physical resources which are shared among different jobs and tasks of same job; this layer provides the infrastructure for storing and handling data and computations. The virtual instances can be rented as per need and can be scaled dynamically [9]. In IaaS (Infrastructure as a service) Virtualization is the fundamental concept; IaaS is placed above the Storage Centers in Cloud Computing hierarchy. The requested resources are delivered as virtualized environment from the service provider to the customer on demand. This infrastructure includes computational device, storage devices, network devices *etc* [10]. Examples of IaaS are Amazon EC2, Microsoft Azure *etc*.

1.4.2 PaaS

In the middle is Platform as a service for handling user's application. Services at this layer are domain specific and language dependent sometimes. All the phases of software development life cycle are covered under this service model. The operating system,

database, development and deployment platforms are provided to the customers. PaaS (Platform as a service) comprises of a set of services for development of an application in the Cloud environment. From application designing to its implementation, verification, validation and then finally its maintenance every phase of software development lifecycle is implemented at this service level [11]. Examples of PaaS are AWS Elastic Beanstalk, Windows Azure *etc*.

1.4.3 SaaS

On the top is the SaaS, developing tools are used over the internet instead of installing them. It is also known as AaaS (Application as a service) [11]. Multi-tenancy feature of the Cloud Computing can be recognized at this service level. Multiple customers can have access to the single instance of an application at the same time [12]. Examples are NetSuite, Google Apps *etc*. Performance and Scalability which are the prominent features of Cloud Computing are essence of SaaS. Web browser acts as the interface for accessing the application provided by the Cloud service Provider. Customer control over this service layer is very less.

1.5 Deployment Models

Depending on the infrastructure ownership, management and operation Cloud Computing services can be deployed on different deployment models. Resources can be on Cloud provider site, user site or on third party premises. Cloud Computing deployment models are categorized as shown in Figure 1.3 [3]:

1.5.1 Private Cloud: The ownership in this deployment model belongs to the organization and infrastructure works for the organization only and management can be done by the organization or a third party. Location of the infrastructure can be within the organization premise or somewhere away from the premise. Private Cloud is more expensive but at the same time more secure then Public Cloud. Other benefits include optimized and maximum use of in-house resources. Security and Privacy can be easily managed. Control and access is in the hands of the one organization only.

1.5.2 Public Cloud: This is dominant among all categories of deployment models. The Cloud provider has the complete access to the resources and manages it. The Service

Level Agreement (SLA) determines the policy, pricing criteria, availability parameters for the service to be delivered and the application to be deployed on the Cloud *etc*. The Public Cloud is open for use of general public. However, infrastructure is located in the Cloud provider's premise only. Visibility and hold of its user over the Cloud is zero and is shared among different users. Public Cloud is a good option for developing and testing applications which demands scalability of the infrastructure. Examples of Public Cloud are Amazon's Elastic Cloud Compute (EC2) and Google App Engine.

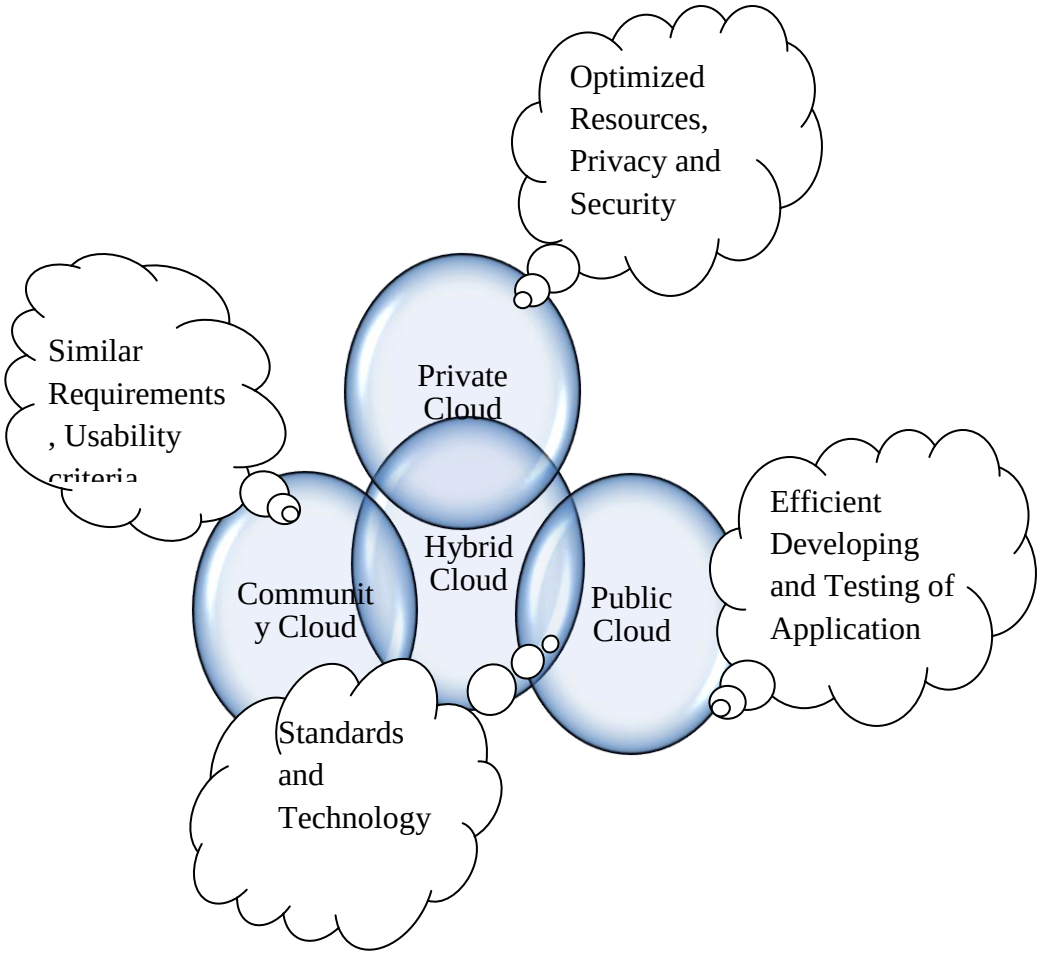

Figure 1.3: Deployment Models

1.5.3 Community Cloud: Depending on the similar kind of attribute requirements of the infrastructure Community model is shared among different organizations. The policy and usability criteria are defined in the agreement for all the users. Management can be done

by one of the organizations or by third party. Infrastructure can be on premise or off-premise. Life Science discipline is using Community Cloud for sharing clinical tools.

1.5.4 Hybrid Cloud: The private, public and community cloud are combined to form a Hybrid Cloud. Together they are used as single entity; some standards and technology are used for gathering the different infrastructure together as per the use. This deployment model provides more flexibility to the business. It supports cloud bursting and organizations for which criticality of application is of concern can deploy on private cloud and for less security concern private cloud can be used.

1.6 Issues in Cloud Computing

Some challenges are there in Cloud Computing due to which it is not fully acceptable in business. The benefits of Cloud Computing are on one hand increasing its popularity but the associated issues are placing barriers in deploying entire business on Cloud Computing [4] [12].

Privacy Issues and Security Issues: Increase in number of points of access with the increase in associated parties and customers leads to the risks of losing privacy of the business information. Organizations have no control on the storage of data and application this fact reduces the reliability of Cloud Computing and hence affects its market value. Security is of major concern as Cloud Computing is facing threat from both sides. The service provider and the user both need to maintain Cloud Computing security. Sharing of resources among customers of different nature is causing security an important issue to be tackled. One physical resource may be shared among two different parties out of which one may be using Cloud for phishing purpose.

Integrity Issues: Authorization plays a vital role in maintaining the integrity in Cloud environment [4]. A Cloud provider may fail to manage the authorization which leads to loss in integrity of information associated with an application. Only authenticated user should be able to make alterations in the application at physical or logical levels.

Availability Issues: User may experience poor availability due to the expansion of Cloud data centers irrespective of the user's location. Everything in Cloud depends on internet

and hence on network resources. Therefore any issue with the network resources cause availability issue.

Performance and Reliability Issues: Meeting high performance in Cloud is not easy as applications keep on migrating from one location to the other. This movement can be caused due to lack of resources, unavailability, fault occurrences *etc*. Reliability of Cloud environment is an important issue to be addressed. Faults occurring in the physical servers, virtual machines and application influence the reliability of Cloud Computing. Reliability can be improved by implementing efficient fault tolerant approaches in Cloud environment.

Fault tolerance issue: Fault tolerance is however a feature of Cloud Computing as it allows an application execution even in the faulty circumstances. At the same time it becomes an issue because mechanism for handling faults depends highly on the type of service being provided, availability of the resources and ease of access to the resources *etc.* Handling faults in Cloud Computing requires root cause analysis of fault occurring along with taking remedial actions.

1.7 Fault and Fault Tolerance

Fault is an incorrect step which causes any deviation in the system behavior from the expected one that result in a failure.

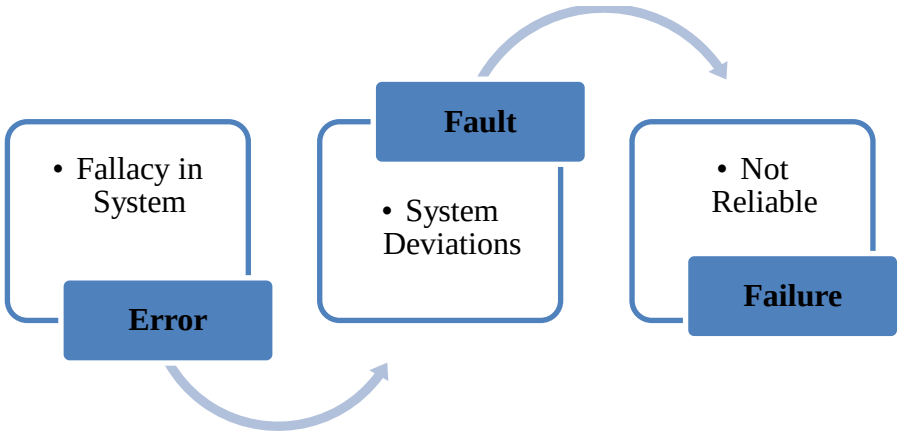

Figure 1.4: Error, fault and failure

Figure 1.4 depicts how fallacy in the system leads to fault and effects system functionality. Change in system behavior is unacceptable as it fails to give the results expected by its user. Deviation can be caused due to any kind of fault, hardware, software or network fault. The delivered service is not matching the required specifications causing it to fail. An error state leads to the failure of the system and reason behind this error is the fault occurring in the system [14].

Fault tolerance means a system keeps on functioning irrelevant of the faults occurring in it so that the expected task is completed on time and required results can be generated however performance may be degraded [15]. Fault tolerance is of great importance for cloud computing technology. Some of the benefits of the fault tolerance are discussed below [27] [29] [30]:

- Security and reliability of cloud computing system can be improved by deploying fault tolerance in the resources.
- Fault tolerance ensures correct and continues operation of the cloud infrastructure.
- Fault tolerance reduces the number of failure therefore cost of detecting and correcting errors can be avoided.
- Because of fault tolerance there is no need of spending effort and time on recovering lost data.

Fault tolerance depends on understanding the behavioral changes in the system when some fault occurs and also to know right remedy so that there would be less impact of fault as well as of fault tolerance technique. It can be performed at different point of time *i.e.* at the development time, testing time and lastly at the execution time. Handling of faults can be done in three different ways [18] [19].

- Fault Avoidance: An optimistic approach according to which it is possible to develop fault free applications by means of rigorous development procedure. Occurrence of fault can be avoided by being more focused and specific.

- Fault Removal: Faults which are recognized during application execution are removed by applying proper root cause analysis approach. Reason for fault occurrence is determined and then rectification is done.

- Fault Tolerance: Fault tolerance is an approach that allows halt-free execution of an application in a faulty scenario. Fault tolerance can be performed before occurring of the fault or after recognition of the fault.

1.8 Research Motivation

Cloud Computing is making computations easy and efficient. Ranging from real time applications to mission critical systems and social networking it is proving itself as the best solution in every aspect. Sharing information across internet, solving computation problems and storing large volume of data has become more proficient and reliable. Therefore, there is a lot more scope of research in Cloud Computing. Cloud Computing has to go through number of refinements so that it can be adopted full- fledged in IT industry [3]. Along with number of benefits there are certain problems associated with this technology. A lot of research work is going on in the direction of improving this technology with a motive of increasing efficiency and swiftness. However, the fault tolerance is still one of the issues for which no efficient methodology is available. Fault tolerance requires complete analysis of the Cloud Computing environment so that the nature of fault and fault scenario can be understood.

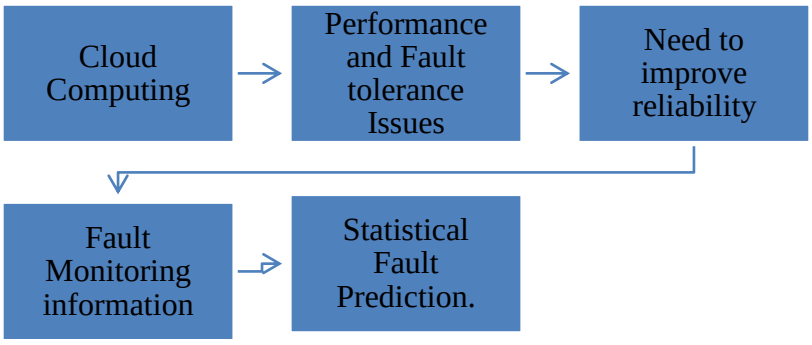

Figure 1.5: Research outlook

Also, improving reliability of Cloud environment is a challenge. Thus, an approach that can handle the faults occurring in Cloud environment and is capable of improving reliability of Cloud Computing should be there. Research outlook is shown in Figure 1.5 in which issue of Cloud Computing that needs an improved approach is given and requirement of a statistical approach is depicted. Any approach that can be proved statistically is capable of giving efficient results. Considering this a statistical approach should be used for prediction of faults. Continuous monitoring helps in improving performance. Therefore, the information generated by a monitoring system can be involved in making predictions for fault occurrences. Hence, in Cloud Computing an approach that can handle faults statistically should be implemented. Reliability can be improved by implementing an efficient fault tolerance approach.

1.9 Thesis Outline

Thesis structure for the research work done is shown in Figure 1.6.

- Chapter 1 Introduces thesis work.
- Chapter2Literature review is briefed in this chapter, the basics of the research topic and existing approaches are discussed.
- Chapter 3 Research gaps, identified problem and objectives of thesis are mentioned in this chapter.
- Chapter 4 The Proposed approach for fault prediction and reliability improvement is described in this chapter, fault monitoring system configuration and statistical analysis distribution model is also given.
- Chapter 5 Implementation and Results are discussed. Interface design of the proposed methodology is described and graphical results are given in this chapter.
- Chapter 6 Conclusion of the thesis work, its contribution and future scope of the proposed approach is given in this chapter.

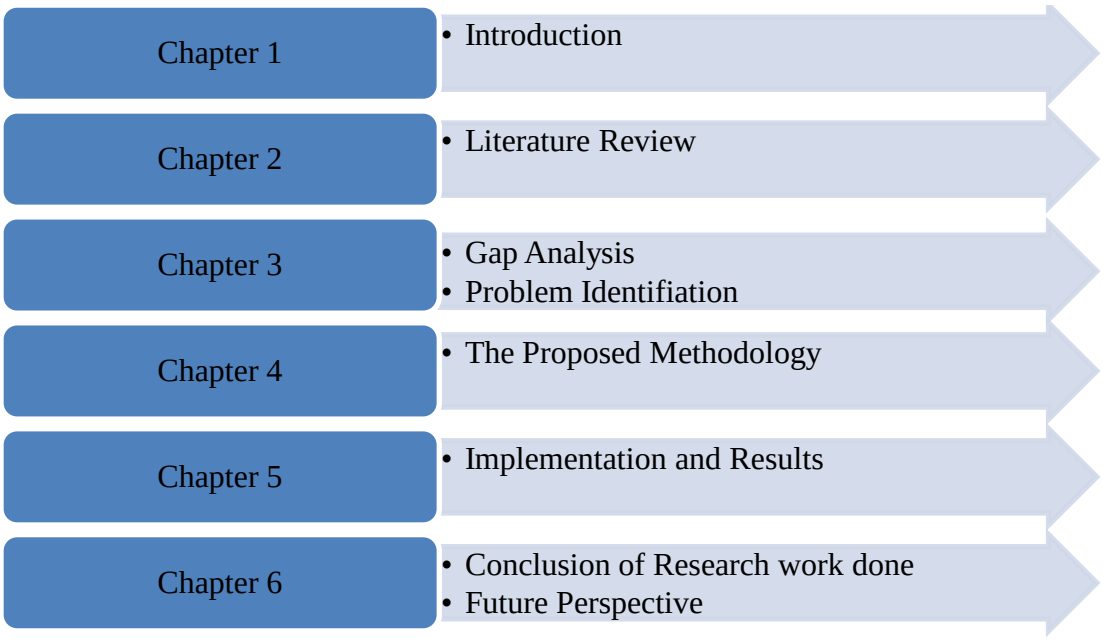

Figure 1.6: Thesis Outline

In this chapter Cloud Computing and Fault tolerance is discussed. Research motivation and thesis outline are also given. Conclusion of this chapter is that fault tolerance in Cloud Computing needs an efficient approach so that reliability of can be improved

Chapter 2

Literature Review

Types of fault and fault tolerance methodologies are discussed in context of Cloud Computing in this chapter. The literature related to fault tolerance in Cloud Computing is briefed and this study helps in determining the research gaps in this field and scope of improvements.

2.1 Types of faults and failure

Faults can occur anywhere in the entire system. In Cloud environment it can be at customer side or can be cloud provider side. A fault can completely damage the system or it can cause failure in some part of the system. Some components may be affected more than the other. Sometimes even on occurrence of fault application keeps on running but performance may be affected. Faults can be permanent, temporary or sometimes come-go-come type means come in the system then go away and then again come back due to any reason. Based on the timing and impact various categories of faults are there [18] [19]. Different faults are categorized as follows:

- Crash faults: shut down the system completely (hard disk failure).
- Byzantine faults: which affects the certain part of the system but application hosted on the system keeps on running however with performance degradation.
- Permanent faults: Permanent out of service state for a component.
- Transient faults: Temporary outage and system gets back to operation after sometime.
- Intermittent fault: Goes away temporarily and hence comes back.
- Malicious fault: Incorrect functioning of the component.

Fault can cause failure in any component of the system. It can affect hardware and software components. Fault can cause security and privacy damage also. Loss of data, inconsistency in content and theft of private information are the result of authentication

faults. Some of the faults and failures based on the faults are mentioned in Table 2.1.Fault at any point of time or at any level can be handled with proactive and reactive fault tolerance according to the scenario.

Table 2.1: Types of Faults [20] [21] [22]

Type of Faults	**Reason for Fault**	**Results in failure**	**Tolerance Technique**
Hardware Faults	Overheating, Incorrect voltage, External temperature,	Node failure, Processor, RAM, Hard disk, RAID Controller	Reactive/Proactive
Software Faults	Algorithmic Error, Missing Specifications	Node Failure, Service Failure, Process Failure	Reactive
Application Faults	SLA violation, Memory Leaks	Service Failure, Process Failure,	Reactive/Proactive
Network Faults	Overloading, Packet Loss	Link Error, Device Issue, Bandwidth Problems	Reactive
Security Faults	Authentication, Authorization, SLA violation	Theft, Loss of Data, Misuse	Proactive
Environmental Faults	Climatic Changes,	Cooling System Performance Issue	Reactive
Catastrophic Faults	Earthquake, Floods	Datacenter Issue	Reactive

Faults in Cloud Computing can be handled from any side customer side or provider side and also can be handled in a collaborative manner.

2.2 Fault Tolerance in Cloud Computing

Fault tolerance is one of the strategies to improve efficiency of Cloud Computing. Fault occurring in Cloud services are big threat to its acceptance in IT industry. Because of hierarchal architecture in Cloud Computing Fault at any one layer can be transformed to other and hence easily creates problem in functionality of layers above it [15] [16]. Hence, Fault tolerance can be used to resolve and handle the issues like reliability and availability in Cloud Computing. Fault tolerance can be achieved by following two basic things one is preventing errors to occur in the system and second is taking appropriate steps so that one fault could not cause entire system to shut down [19]. Before taking any preventive step we need to find when and how faults could occur in the system. Faults can be handled at different point in time of their occurrence and on this bases three types of fault tolerance is there [21].

- Proactive fault tolerance: Prevention is better than cure kind of strategy is followed here, fault conditions are detected earlier and remedial actions are taken to prevent failure of the system.

- Reactive fault tolerance: This method is applicable after the fault has occurred and is affecting application running on the cloud. A kind of on demand fault tolerance service is implemented to handle the error.

- Adaptive fault tolerance: Technological advancement is the basis of this approach; fault is handled as per the circumstances when fault occurs automatically and adaptively.

2.3 Fault Tolerance Methodologies

Methodology for achieving Fault Tolerance in Cloud Computing includes Redundancy of resources, Migration of Virtual Machine, Load balancing, and Replication.

- Redundancy**:** Redundancy can be done for every resource involved in the Cloud Computing, Hardware Redundancy, and Software Redundancy. Hardware

redundancy, keeping additional hardware components either to detect wrong results from the single component or to prevent loss if one component fails [20]. Software redundancy, running one application on separate but similar software components at the same time helps in getting efficient and correct results and hence beneficial for tolerating faults.

- Migration: Physical server sometimes faces some faulty circumstances which affects the execution of application in virtual machine. Therefore, the particular virtual machine is moved from one physical server to the other without affecting the execution of the application. It helps in fault tolerance by allowing continuous execution of the application even if some fault occurs in the physical server [23].

- Replication: Replicating the operations executing on one virtual machine on some other virtual machine at the simultaneously so that if primary machine face any issue the secondary could replace it and fault tolerance can be easily achieved [24].

- Load Balancing: Allocating jobs more than its capability leads to fault occurrence in a node and may cause entire service failure. Therefore to prevent such faulty conditions load balancing is a good practice. Optimize use of resources and continuous monitoring of the performance for dynamic load balancing helps in hurdle free execution of the applications. Load balancing algorithms can be used for equally dividing the load among the nodes [25].

Fault tolerance involves three major processes which begin with monitoring of the resources, predicting fault occurrences by analyzing the collected information and then finally taking remedial steps accordingly. Proactive fault tolerance is the result of predicting faults and autonomic self-healing in Cloud Computing. Prediction requires continuous monitoring of the system so that it can be prevented from failure. In [44] [39] existing probability distribution models are used for doing performance analysis. This type of approach helps in improving reliability of Cloud environment. Artificial intelligence is another of predicting faults using rule based applications for analysis and prediction. Although a number of efficient techniques are available for handling faults in cloud computing system, but there are many challenges in implementing these

technologies [26] [27] [28]. Taking benefits from the resources more than for which they are actually meant causes hindrance in solving any issue. Some techniques are customers satisfactory some are service provide based, hence a technique which satisfies both is hard to be implemented. Carefully designed data centers are prone to the big faults because of the size of the data, location of the resources which is segregated all over the world. Increase in number of users and rising demand of high computation power is another prominent thing to be kept in mind while applying any of the fault tolerance technique. As cloud computing is based on virtualization of the resources so it becomes difficult to handle faults occurring in the virtual system and also migration without affecting user is not so easy. Algorithm designed for one type of system may or may not be suitable for other type of system.

2.4 Literature Analyzed for Research Purpose

Existing research work has been studied to get relevant information in context of fault tolerance. Various authors have worked on fault tolerance and their contribution is reviewed for understanding the fault occurrence in Cloud environment. The summary of information collected is given here. Various approaches have been studied and effort is made in the direction of getting an idea to solve fault tolerance issue of Cloud Computing.

Jhawar and Piuri [17] addressed the Reliability and Availability of Cloud by suggesting solution for different categories of fault occurring in Cloud. A conceptual framework is proposed in which fault tolerance for user's application is implemented as a service. A statistical approach is followed and Markov model is considered for analyzing the behavior of the system.

Chao-Tung Yang et al. [32] have used technology and tools available for virtualization and implemented virtualization based fault tolerance for improving scalability and reliability in Cloud environment. Cluster based application like Hadoop has been used for experimental purpose and observation resulted in decreased downtime of the application.

Jing Deng et al [30] focused on fault tolerance and reliability by working on a particular problem *i.e.* Matrix multiplication. The approach followed to improve reliability is

simply the selection of appropriate cloud and designing the system carefully. By following this reliable results can be obtained in a scientific application running on cloud.

Bala and Chana in [20] worked on handling software faults occurring in virtualized cloud environment. Availability and reliability are focused while implementing the fault tolerant architecture. Autonomic fault tolerance is implemented using HAProxy and data replication technique for handling the faulty situations.

Tchana et al. [31] suggested a collaborative solution for fault tolerance in Cloud Computing. The proposed scenario is implemented in an autonomic Cloud environment and faults at different level are considered. Fault management by single party and by sharing between the customer and the provider is compared.

Ward and Barker [33] proposed a solution based on semantic web technologies similar to existing monitoring tools for collecting relevant information and detecting faults. The collected information generates machine readable content and hence additional monitoring tool is not required. The proposed approach supports dynamic and complex behavior of Cloud Computing.

Yu. He.et al. [34] designed a monitoring framework which is for analyzing simulation of Cloud. The hierarchical designed based on simulation cloud is proposed which helps in collecting the performance data and analysis of the same for detecting abnormalities in the behavior. Hence is supporting fault tolerance as well with the monitoring system as the middleware.

Pervila [35] studied Nagios capabilities for detecting faults and also its self-healing attributes to determine its ability to work in self-healing environment. Nagios plug-in is used for connecting with other application and an approach for self-healing is designed.

Gogouvitis et al. [36] considered QoS as important factor in Cloud Computing environment and proposed architecture with the purpose of monitoring considering the VISION cloud project. An interface is designed for provisioning monitored information to the client related to performance analysis of the Cloud environment.

Nezih Yigitbasi et al. [37] proposed a framework called C-Meter for analyzing performance of cloud implemented in python. Performance metrics are derived for evaluating the computing and network resources.

Chanon Wang et al. [38] followed a statistical approach for handling faults in the cloud based multimedia system. The existing fault tolerance techniques like retry is examined and compared with check-pointing. The parameters for performance evaluation involved response time and arrival rate. Failure rate and processing time are also investigated before and after implementing the retrying technique for fault recovery.

Bo Yang et al. [39] implemented queuing model for understanding the cloud performance overhead when fault tolerance technique is used for fault tolerance. Distribution model is used for evaluating the reliability of Cloud when number of jobs arrived for scheduling. Service response time is taken as the parameter for cloud performance with and without implementing recovery technique.

Zheng and Liyu [40] designed a framework for web services in which fault tolerance is achieved at runtime. A collaborative approach is followed for reconfiguring web service dynamically and dependability is evaluated.

Song Fu [41] utilized the virtualization concept efficiently for reconfiguring virtual machine so that availability can be improved. Predicting failure helps in selecting appropriate node and hence results in better performance. A metric called capacity-reliability is used to find the best fit node in cluster computing environment.

Polze et al. [42] followed a migration approach for preventing failure of the application. A proactive approach is used for moving a virtual machine from susceptible physical server. A probability based reliability measure is done which triggers the migration. This approach helps in keeping the Service Level Agreement (SLA) conditions for providing service.

Approaches followed by the various authors for fault tolerance have been studied. Based on the analysis of existing work an improved approach can be proposed for fault prediction.

Chapter 3

Research Gaps and Problem Formulation

This chapter is focused on analyzing the research gaps and defining the problem statement based on the gap analysis.

3.1 Gap Analysis and Problem Identification

Gap Analyzed and Problem Identification steps are given in Figure 3.1.

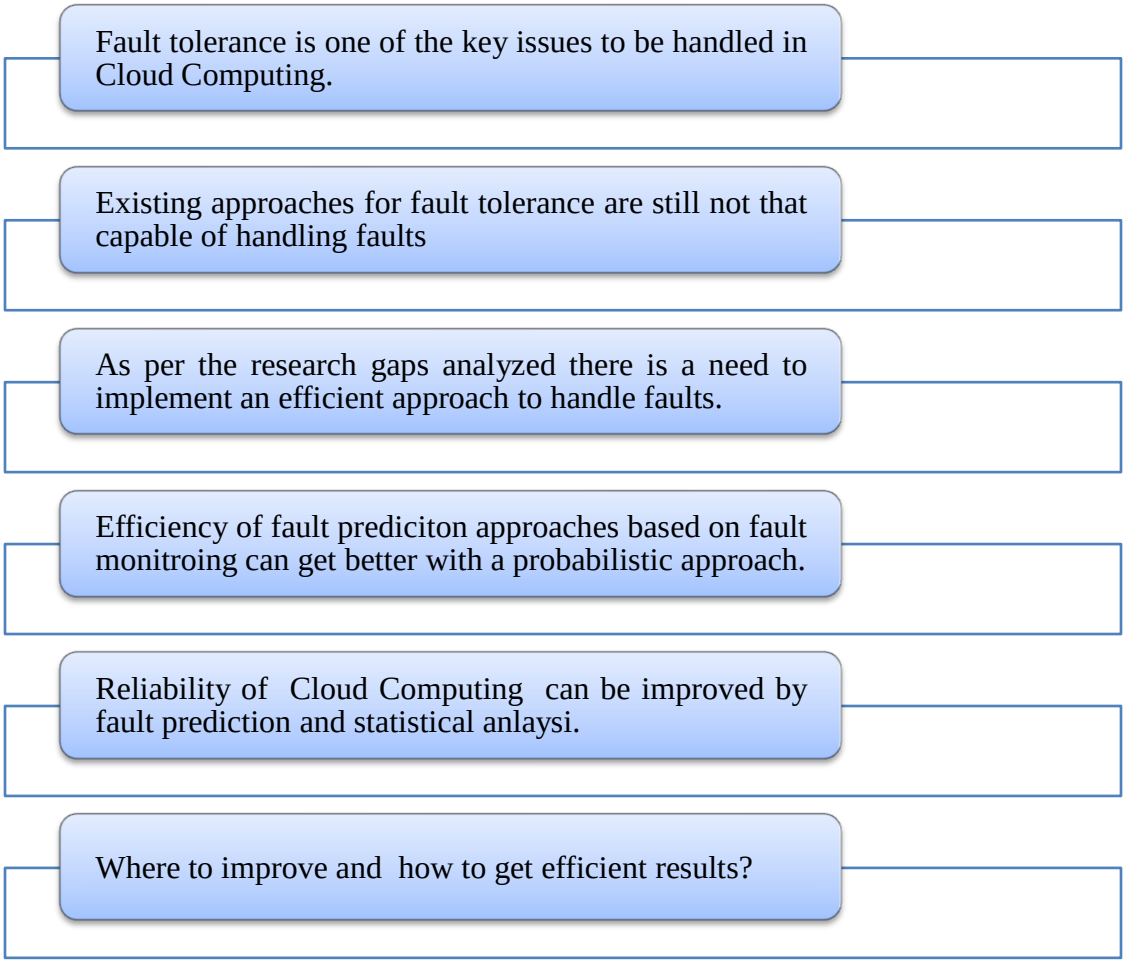

Figure 3.1: Gap Analysis and Problem Identification

As Cloud Computing is an emerging practice which is improving efficiency of computing services. As per the research analysis it has been demonstrated that there is a need to

improve fault tolerance approaches in Cloud Computing. To fill this research gap statistical analysis approach for fault prediction can be a good alternative. This approach can predict and handle faults. Another problem in cloud environment exits is of reliability which can be improved by following probabilistic approaches for fault prediction.

3.2 Research Objectives

The gap analysis made it easy to understand the existing approaches being followed for handling faults. Aim of this thesis work is to propose an approach that can be used for solving the identified problem.

- To study state of art of fault tolerance in Cloud Computing and approaches for improving reliability.
- To analyze monitoring information generated from the analysis of continuous monitoring and finding parameters for fault prediction.
- To propose a probabilistic approach for fault prediction and statistical analysis that can improve reliability of Cloud environment.

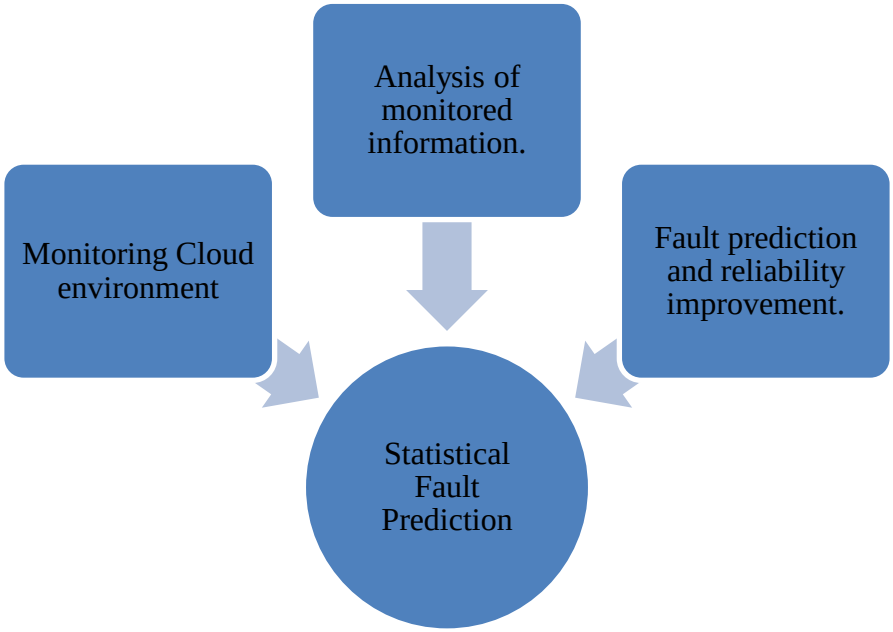

Figure 3.2: Research Objective

Chapter4

The Proposed Methodology

In this chapter the proposed methodology for solving the problem identified during gap analysis is described. Fault monitoring and statistical analysis for finding fault occurrence probability is also discussed.

4.1The Proposed Approach for Fault Prediction

Fault prediction is an approach that allows handling a faulty scenario well in advance so that loss of data and resources can be prevented. Accuracy in prediction results in reducing the post development maintenance efforts [43]. The proposed approach of fault prediction involves fault monitoring and statistical analysis of virtual machine is shown in Figure. 4.1.

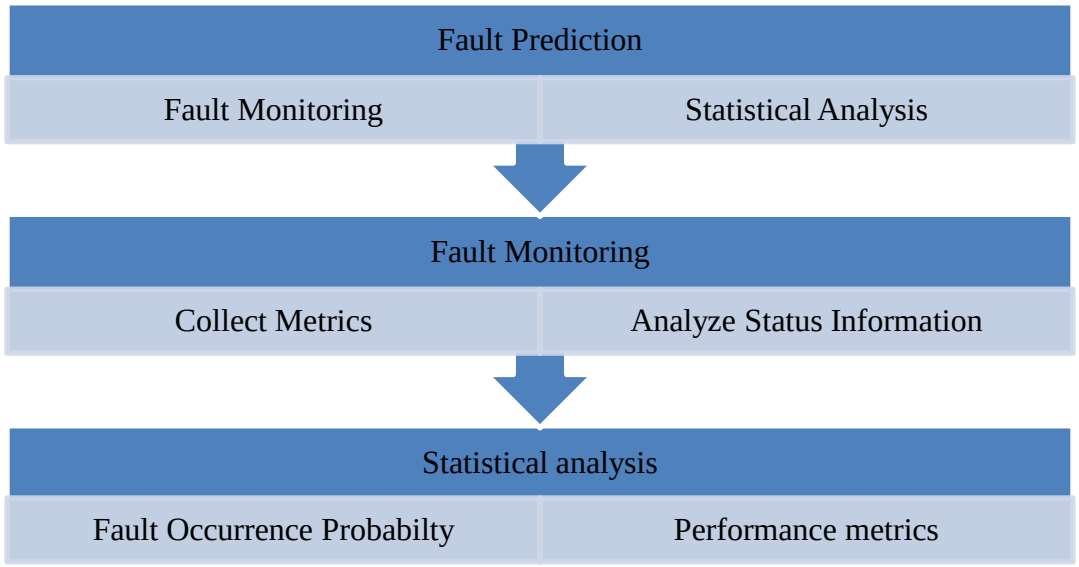

Figure 4.1: Fault Prediction Approach

A statistical approach based on fault monitoring for predicting fault occurrence probability in virtual machine has been proposed. Fault monitoring is analyzing the machine behavior to detect any kind of unexpected and unknown states of services running on the machine. It can be done by a monitoring system which is capable of collecting metrics and analyzing machine state based on them. Uptime and downtime are

taken as parameters from monitoring results to find the availability status on the basis of which fault proneness for a virtual machine is further estimated using distribution model. For statistical analysis the properties of analyzed data from monitoring results are compared with existing distribution models and accordingly Poisson distribution [44] is applied for predicting future fault occurrences. Prediction requires relevant data and appropriate methods to get efficient results.

The methodology followed for finding the solution is briefed here. First step is to monitor and collect the metrics for analysis of virtual machine scenarios of fault occurrence now, find corresponding types of failure caused due to deviation in system behavior. Then, Analyzing the fault occurrences, availability, criticality and failure status of the virtual machine on the basis of metrics collected. After analysis the distribution model that can be used for predicting fault probability is decided. For implementing distribution model, an application that is capable of predicting fault occurrence probability for the virtual machine is designed.

A standard approach as used by Simon and Dorband in [44] Poisson distribution model has been adopted for solving the identified problem of Fault Prediction. Fault occurrence behavior of the system can be compared with existing distribution model properties. From the comparison done the fault occurrence probability can be mapped with Poisson distribution model properties. The probability is constant at any time and also occurring faults are independent similar to events in Poisson distribution [39]. Using uptime and downtime of a virtual machin availability of the node can be calculated .From this availability information, it can be easily observed that m out of total n nodes are showing less availability. Hence, these nodes are needed to be considered for further observation for predicting faults considering a threshold value of availability. Once the node(s) with less availability percentage is detected, then the criticality status of the node is considered which helps in determining the fault occurrence probability

4.2 Fault Monitoring

Fault Monitoring means observing the system performance so that any of the suspected change in system's state can be detected. Monitoring helps in detecting the root cause of an event by collecting the metrics and analyzing them. A fault cannot be eliminated

without knowing the root cause of its occurrence and also it is not possible to handle system failures without having any idea about system's performance. One of the available monitoring systems has been selected for this purpose called NAGIOS [35]. Its main purpose is to check nodes, time to time or for specified time to detect any of the uncertain and suspicious behavior. Finally, it notifies the status to administrator or associated contact group so that preventive method can be followed for avoiding system failure. Monitoring is done by monitoring agents on the basis of metric values obtained from the monitored components [45]. Monitoring process is shown in Figure 4.2. Fault tolerance can be achieved by following two basic things one is preventing errors to occur in the system and second is taking appropriate steps so that one particular fault cannot cause the entire system to shut down. Before taking any preventive step, it is needed to find when and how faults could occur in the system [35].

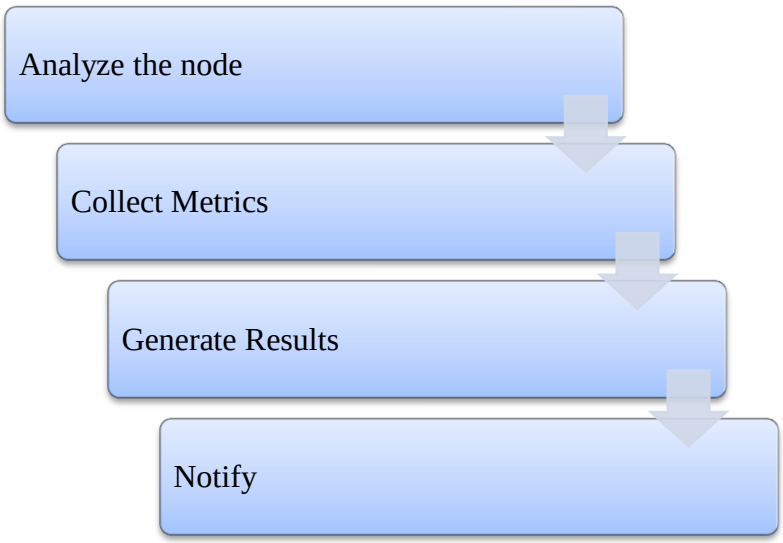

Figure 4.2: Fault Monitoring Process

Nagios is used for tracking faults in the system. It is an open source tool for monitoring of physical and virtual machines. It tells the current status of the hosts and services which are added to its configuration files. Nagios configuration files can be edited easily as per the requirements as shown in Figure 4.3. Service and host entries are made in configuration file for which monitoring is to be done. Configuration file for Nagios is /etc/Nagios/nrpe.cfg. After the entire configuration is done and required changes are made, restart the Nagios service and check on the browser if responding correctly. Using

Nagios it is easy to write own scripts for monitoring hosts and associated services. After configuring Nagios core add number of users and services as shown in Figure 4.5 to its configuration file [35].

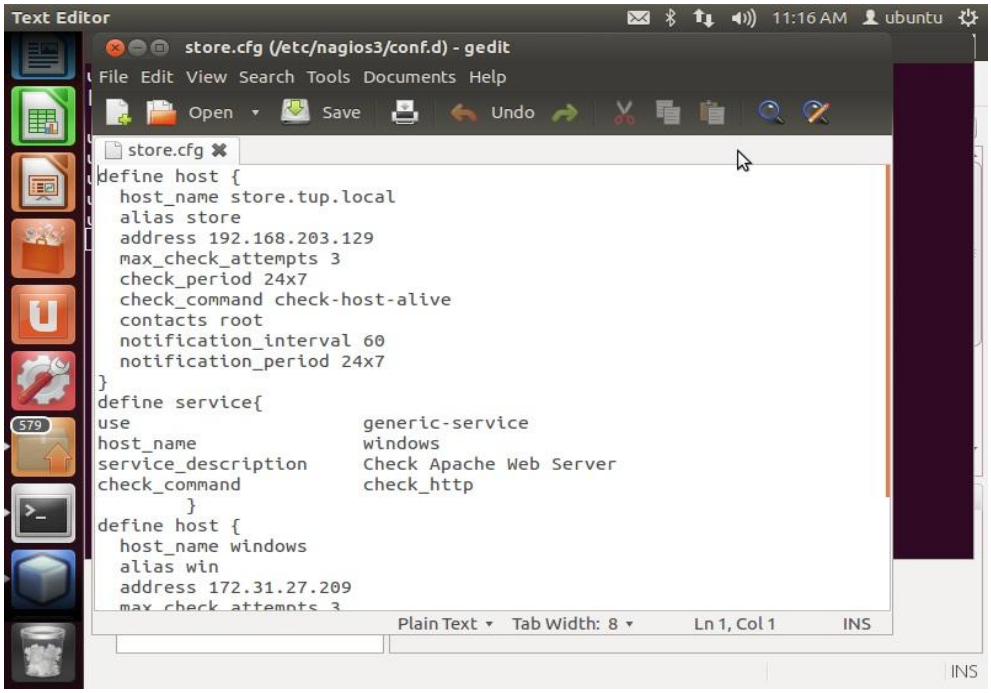

Figure 4.3: Host entry

In Nagios configuration file the structure to define a service or host is given, own service check script can be written following that structure. For example following scripts define a service check for windows host:

define service{

use generic service

host_name windows

service_description name of the service to be tracked

check_command command to track service

}

A separate host file is created for storing all the relevant information for the host. The services to be checked, time interval and notification present are given in this file. During installation setting the username, password, postfix configuration and altering Nagios configuration files are some of the tasks needed to be done for better results as shown in Figure 4.4.

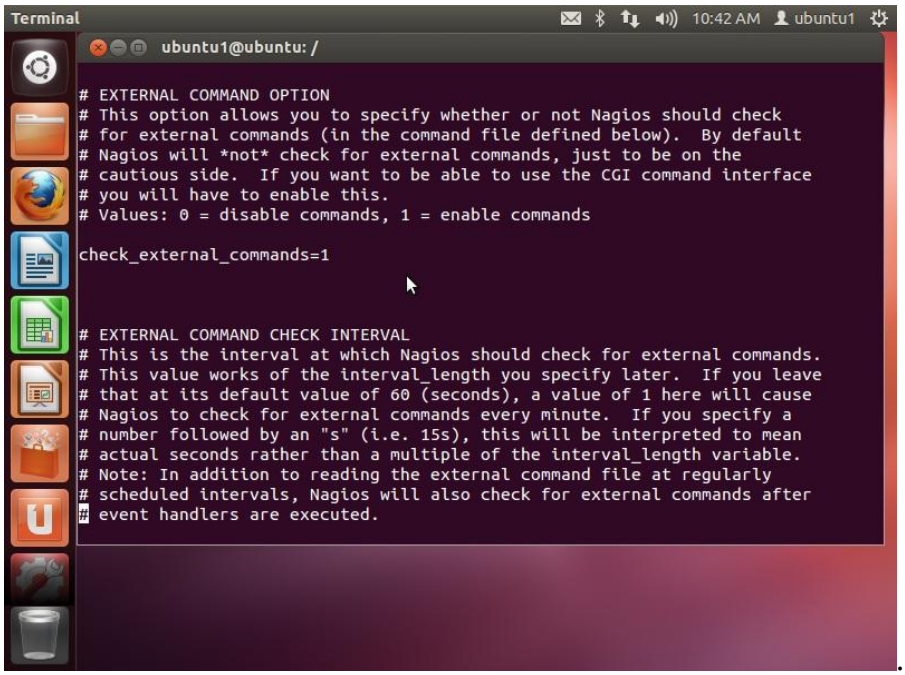

Figure 4. 4: Configuring Nagios

The status symbols used in Nagios are shown in Table 4.1. Each state has its own significance.

Table 4.1: Result interpretation by Nagios

Code	State
0	Up
1	Up or Down
2	Down/Unreachable
3	Down/Unreachable

Up state shows that the host is available and if state is down that means host is out of service. The analysis of results generated from monitoring is done to get the required parameters for fault prediction. The status report tells the virtual machine status information. The uptime and downtime for the monitored servers are taken which is used to find availability. The criticality status is also analyzed for finding fault occurrence probability. For fault prediction purpose availability is taken as the criteria to determine which machine is more prone to faults. Figure 4.5 determines the host status checked for particular time duration and uptime and downtime of the machines are taken for finding availability using application interface developed. One of the Nagios plug-in is used here called NRPE. It tracks service on remote system like disk space or CPU load. It allows Nagios server to query remote system as if it is local and to check services on the remote system.

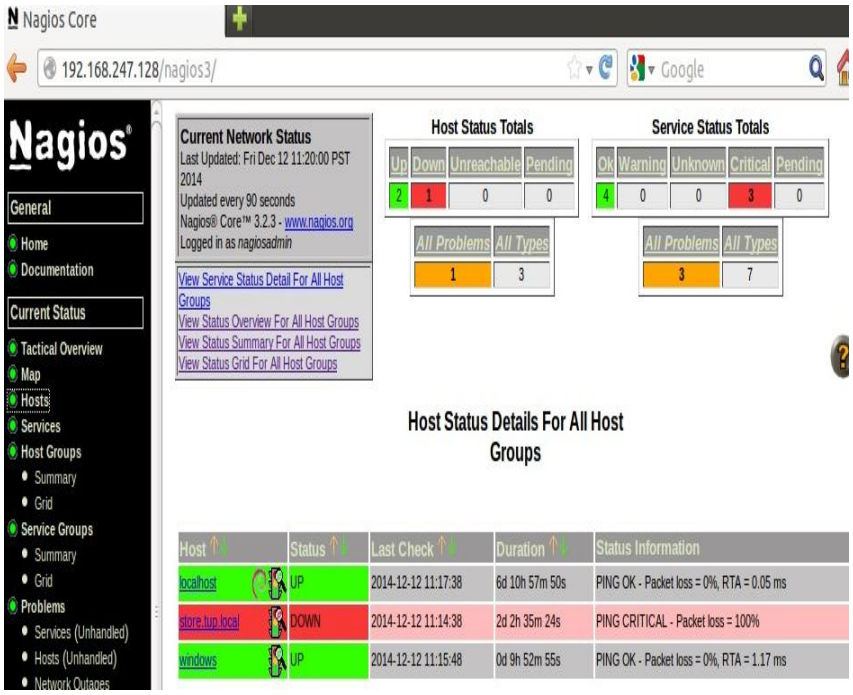

Figure 4. 5: Uptime and Downtime Statistics

Results from the analysis are generated in different forms reports, summary, histogram, graphs and maps. Fault tolerance involves three major processes which begin with monitoring of the resources, predicting fault occurrences by analyzing the collected

information and then finally taking remedial steps accordingly. Fault monitoring is results are now used for statistical analysis.

4.3 Statistical Analysis

Following a standard approach as used by Simon and Dorband in [44] Poisson distribution model has been adopted for solving the identified problem of Fault Prediction. Fault occurrence behavior of the system can be compared with existing distribution model properties. From the comparison done the fault occurrence probability can be mapped with Poisson distribution model properties. The probability is constant at any time and also occurring faults are independent similar to events in Poisson distribution [39]. A fault is nothing but deviation of system functionality from the expected behavior. Lack of resources, long downtime, incompatibility between job arrival rate and scheduling of the jobs to the nodes *etc.* can be the reason for this deviation. Considering these key points, an approach for calculating availability of the nodes and fault occurrence probability has been proposed and implemented.

Using uptime and downtime of a virtual node find the availability .From this availability information, it can be easily observed that m out of total n nodes are showing less availability. Hence, these nodes are needed to be considered for further observation for predicting faults considering a threshold value of availability. Once the node(s) with less availability percentage is detected, next task is to identify the reasons for downtime. Therefore, for this purpose criticality status of the node is considered which helps in determining the fault occurrence probability. On the basis of results obtained from the analysis of the nodes (virtual machines on VM box), availability has been calculated. Further based on availability the fault prediction is considered to be done for node with less availability and reliability of particular node for hosting an application is measured. The experiment is conducted on 3 different virtual machines for 10 hours a day and for 10 days that is for a total of 100 hours. Fault is injected for 25 times in the total experiment duration. For improving reliability the number of fault injections is reduced in so that application can be executed completely. The basic method of finding availability [45] is considered here as shown in equation (1):

Virtual machines are manually kept up and down for the experimental purpose. The collected information is used to determine the availability of individual servers. Uptime and downtime for individual nodes are determined by Nagios. Experiment conducted gives the results for availability of three virtual machines being monitored.

$$\text{Availability} = \frac{\text{uptime}}{\text{uptime}+\text{downtime}} \quad (1)$$

For a particular instant of time availability for virtual machines is calculated and the result shows there availability is 97.16%, 97.66% and 98.99% at the time of observation. From this availability information, it can be easily observed that m out of total n nodes are showing less availability. Hence, these nodes are needed to be considered for further observation for predicting faults considering a threshold value of availability. Once the node(s) with less availability percentage is detected, next task is to identify the reasons for downtime. Therefore, for this purpose criticality status of the node is considered which helps in determining the fault occurrence probability. Here, the disk space is kept critical for injecting faults and evaluating the proposed statistical method for fault prediction. The strategy followed for fault prediction is a combination of fault monitoring and statistical model for prediction. The monitoring results helped in knowing virtual machine availability status, the duration for which machines was in fault prone state *i.e* .critical status of the machine and also complete crash down situations. Further, the prediction model gives the fault occurrence probability statistics. Failure rate of the system can be mapped with one of the distribution model according to its properties. Every virtual machine node is examined for the status and time for which any virtual machine node remains in critical condition due to any of its service criticality state in the total experiment duration is recorded. The average criticality of all the nodes is used to calculate mean value for Poisson distribution as shown in equation (2).The probability of fault occurrence as observed is found to be constant at any time and also fault occurrences are independent similar to events in Poisson distribution [39]. For a random variable x which follows Poisson distribution with parameter μ*i.e.* the average number of events in the given time interval formula for probability mass function is [44] [48]:

$$P(x) = \frac{e^{-\mu}\mu^{x}}{x!} \quad (2)$$

Where x = 0,1,2,3.....; μ>0.

Using Poisson distribution, chances of fault occurrence in critical system state are calculated to fault prone and hence there are chances of fault occurrence. Also, the Probability of x task failures in time t can be calculated as shown in equation (3) [39]:

$$F_x(t) = \frac{e^{-\mu t}(\mu t)^{x}}{x!} \quad (3)$$

Suppose server is working for 50 hours and as calculated on an average it is fault prone for 0.1 hours then probability for the node to be critical and hence fault prone for 2 hours is 0.0842. Reliability and fault occurrence are contrary to each other. From the fault occurrence probability rate, Reliability, probability that system will not fail for time interval t, can be calculated as shown in equation (5) [48]:

$$F(t) + R(t) = 1 \quad (4)$$

F(t) : Probability of fault occurrence

R(t) : Probability of fault free execution *i.e.* Reliability

Therefore,

$$R(t) = 1 - F(t) \quad (5)$$

Also, to find the sustainability of the machine for particular type of faults, a sustainability index can be calculated. It takes the mean time to failure and total runtime of the machine as parameters. A high value for Sustainability index depicts that the machine is in good state and can perform even better in faulty scenarios. Sustainability Index can be calculated as shown in equation (6) [44]:

$$S = \frac{F}{T} (7)$$

Where,

S: Sustainability Index

F: Meant Time to Failure

T: Total Runtime

A low value of sustainability index indicates resilience of machine for faults is less and high value indicated high sustainability. Hence, reliability of machine can be improved using this value. Repairable faults can be corrected which decreases the fault occurrence probability and hence increases the reliability. Improvement in reliability can be validated from an increased value of sustainability index.

In this chapter the proposed approach for fault tolerance in Cloud Computing has been discussed. Fault monitoring system configuration information and its results are given. Statistical analysis is done on the basis of parameters taken from monitoring information.

Chapter 5

Implementation and Results

In this chapter the implementation of the proposed approach is given and the results obtained are also discussed.

5.1 Implementation

The fault prediction model is implemented as an application interface called "Statistical Scrutinizer" .A graphical user interface is designed using Python 2.7.10. The application is a solution for the fault prediction problem in Cloud Computing. User just needs to enter the relevant metrics for finding availability, fault occurrence probability, reliability, sustainability index and the application gives the corresponding statistics for the machine. Using monitored metrics and parameters, statistical prediction of fault occurrences in cloud environment has been done. The interface for Statistical Scrutinizer is shown in Figure 5.1.

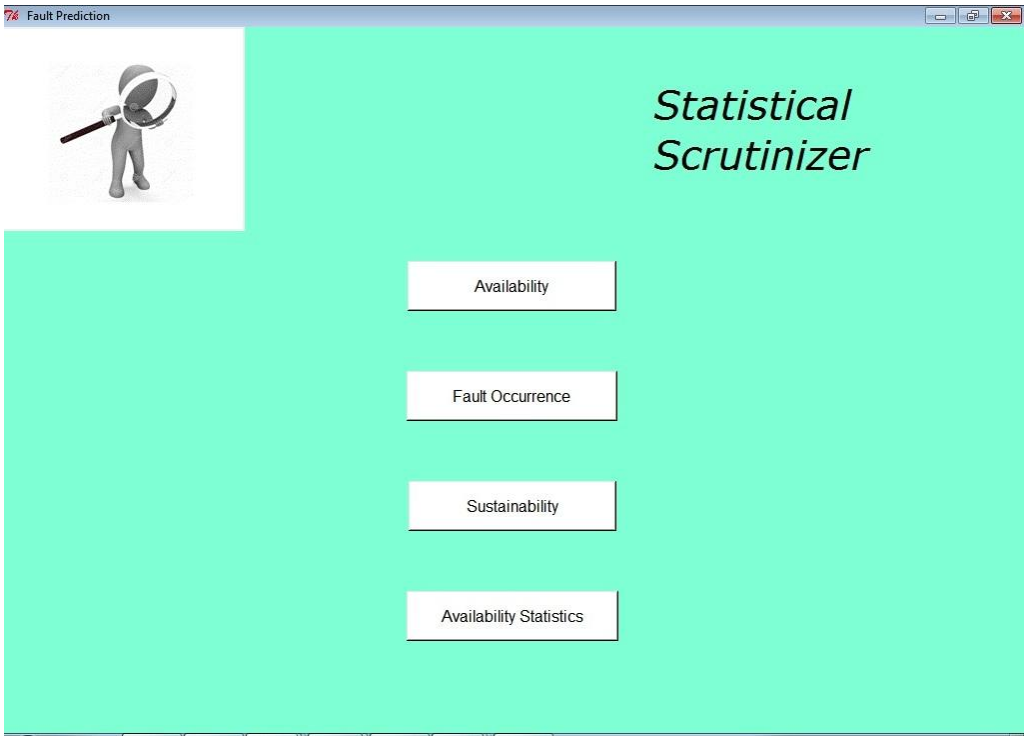

Figure 5.1: Application Interface

The interface gives the availability statistics and fault occurrence probability that can be plotted for better understanding. Experiment is performed on virtual machine using virtual box to get the parameters for statistical analysis. Uptime and downtime of the machine are used to calculate its availability. Disk criticality is taken as criteria for showing the system state critical. A particular machine is kept critical for certain time per day and the total duration is taken as the interval for which a fault can occur in the machine due to critical disk space. This can be done for any service and for any particular interval of time. The probability for fault occurrence is calculated based on the Poisson distribution model with taking a mean value and a random number for number of fault occurrences. Results are based on the analysis of the machine behavior for the experiment duration.

5.1.1 Availability Statistics

Uptime and downtime of the machine are used to calculate its availability. Based on the availability of the machine it can be predicted that this particular machine is prone to failure or not. Availability uptime and downtime values are entered as shown in Figure 5.2.

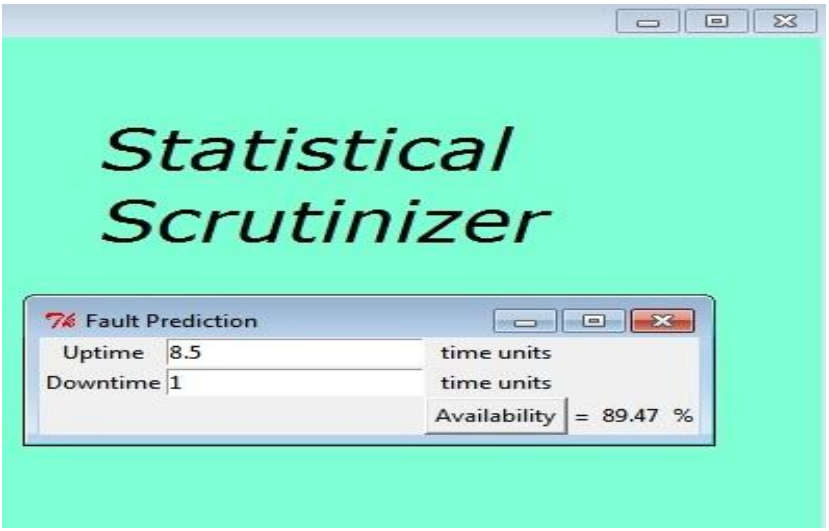

Figure 5.2: Availability Window

For calculating the availability user needs to enter the "Availability" button on the graphical user interface. A new window will open as shown in Figure 5.3. The availability calculations are based on general availability formula based on up and down states of a machine.

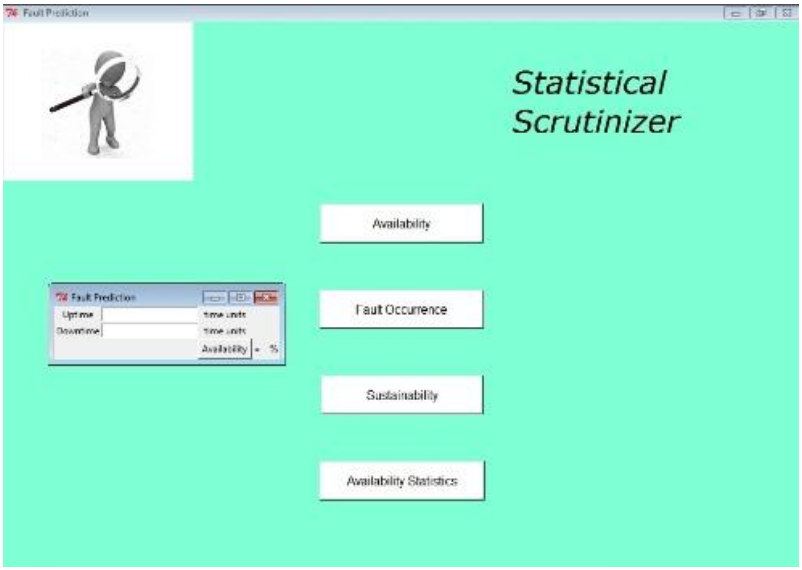

Figure 5.3: Calculate availability

The uptime and downtime values are taken from the monitoring analysis as shown in Figure 5.4

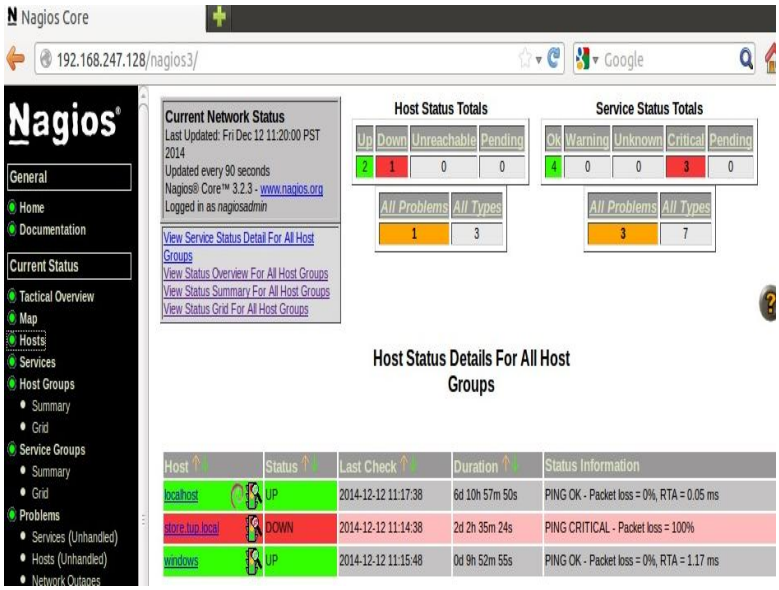

Figure 5.4: Virtual machine status

In the experiment performed virtual machines are monitored and their status is checked periodically. This periodic check gives the uptime and downtime for the machine which are given as input to find availability as shown in Figure 5.5

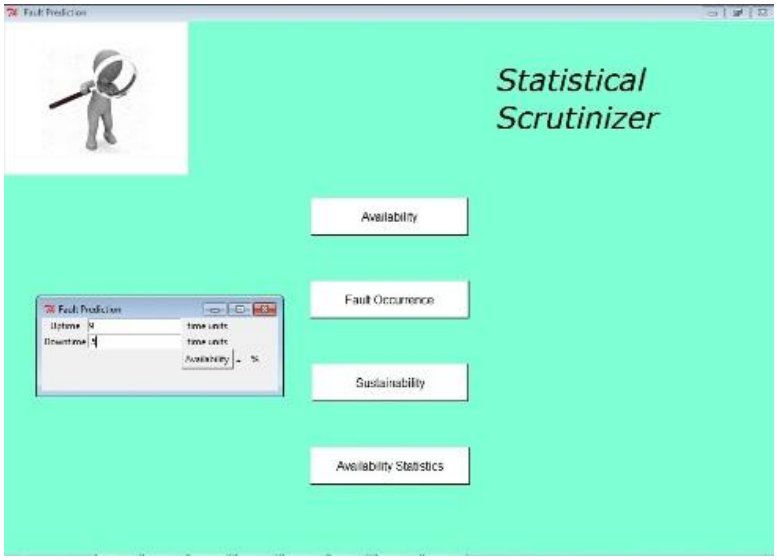

Figure 5.5: Uptime downtime entries

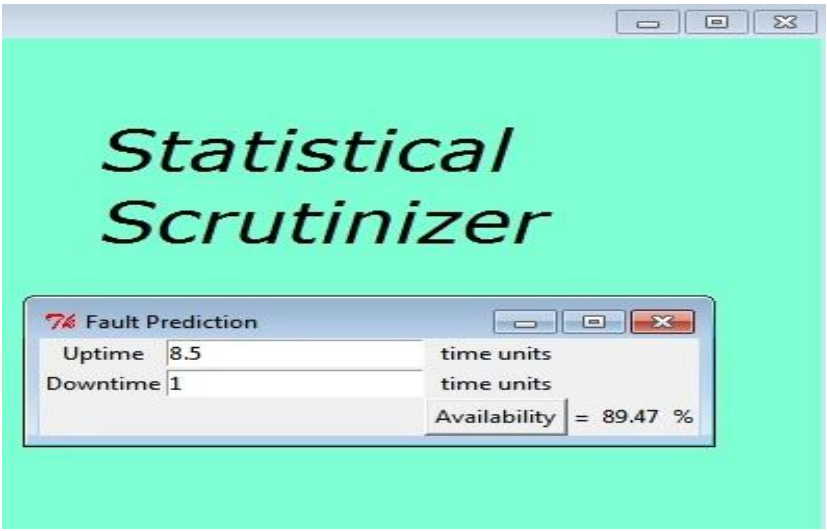

Figure 5.6: Calculated Availability

The calculated availability percentage is obtained by clicking calculate button as shown in Figure 5.6. The availability status of a machine can be analyzed for a particular duration and from that analysis a threshold value can be decided and set as criterion for deciding the fault analysis requirement of the machine. If availability is more than

threshold value than machine is less prone to faults and if availability of machine is decreasing as per the threshold value then it requires further inspection.

5.2.2 Fault Occurrence Probability

From the fault occurrence option number of faults occurring in a machine can be predicted. It gives probability of 'x' number of faults for a given machine whose average fault occurrence per unit time is known. Also, using the mean time for fault occurrence the probability of fault occurrence at any time 'x' can be calculated. The mean value is calculated from the criticality status of the machine. In the entire duration of experiment a total of 25 faults have been introduced in 100 hours. Accordingly, average fault occurrence per hour is 0.4 which is the mean value to predict probability. Hence, from this number of fault occurrence can be predicted. Window for fault prediction to find probability is shown in Figure 5.7.

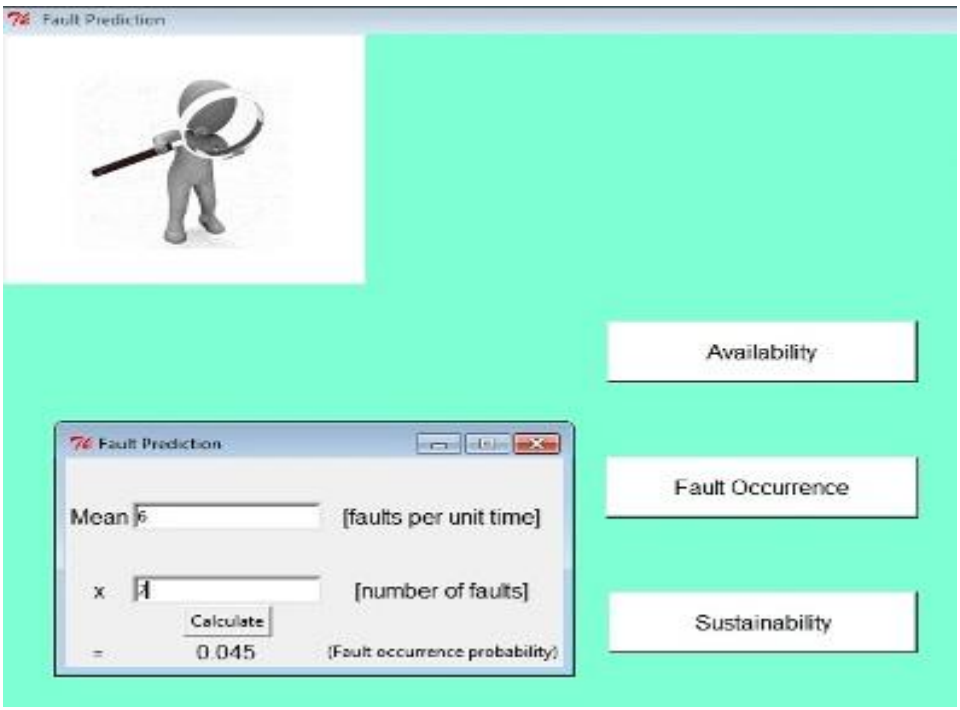

Figure 5.7: Fault Prediction.

Disk criticality is taken as criteria for showing the system state critical. A particular machine is kept critical for certain time per day and the total duration is taken as the interval for which a fault can occur in the machine due to critical disk space as shown in Figure 5.8.

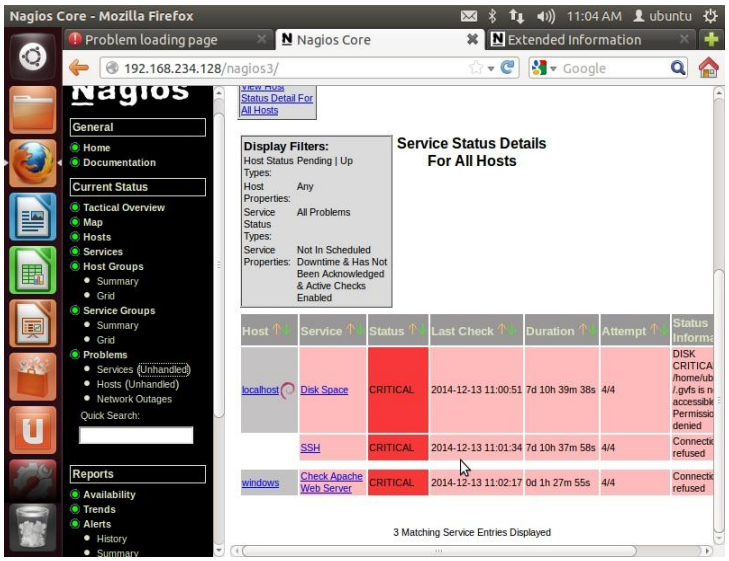

Figure 5.8: Criticality status

The value for mean and the random variable x which is representing the number of faults are entered as shown in Figure 5.9.

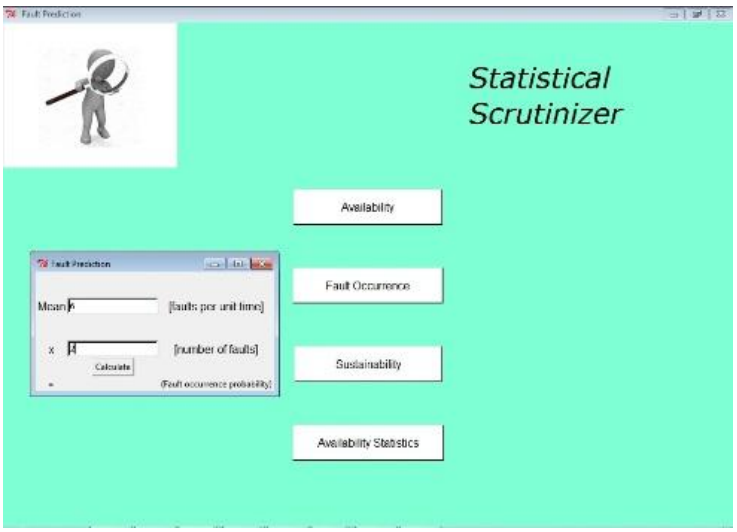

Figure 5.9: Mean and Number of Faults

The values entered for mean and number of faults is the parameters used for fault prediction. The calculate button when clicked gives the probability as shown in Figure 5.10. The calculations are done by implementing Poisson distribution and it gives the probability of 'x' number of fault occurrence whose mean value is known.

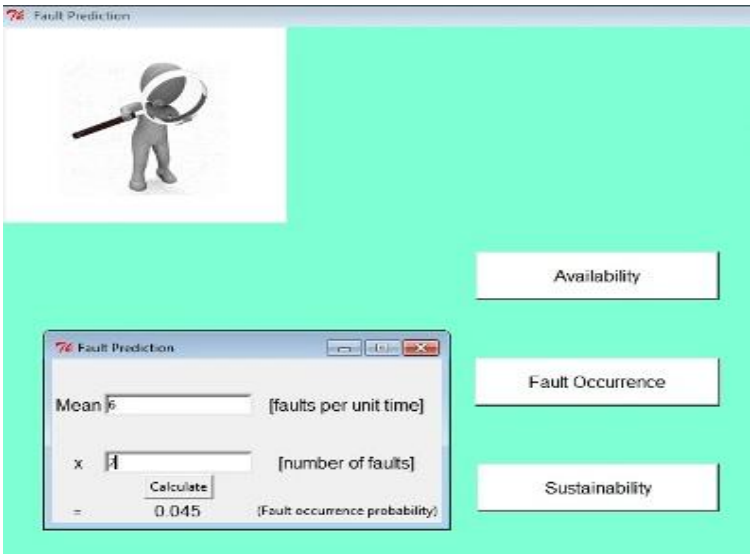

Figure 5.10: Fault occurrence probability

Reliability and fault occurrence are contrary to each other. From the fault occurrence probability rate, Reliability, probability that system will not fail for time interval t, can be calculated as shown in equation (1) and (2) [48]:

$$F(t) + R(t) = 1 \quad (1)$$

F(t) : Probability of fault occurrence

R(t) : Probability of fault free execution *i.e.* Reliability

Therefore,

$$R(t) = 1 - F(t) \quad (2)$$

Also, to find the sustainability of the machine for particular type of faults, a sustainability index can be calculated. It helps in improving availability of machine as repairable failures are chosen and corrected and then sustainability index improves.

5.2.3 Sustainability Index

To find sustainability index the criteria is the criticality status of the machine. This index value indicates the ability of machine to sustain number of faults for a given interval of time. For this mean time to failure is taken as input and the total runtime of the machine is the time interval for which the ability to sustain is calculated. A high index value

depicts high sustainability and low index value depicts that the machine needs immediate maintenance. As disk space is taken as the criteria to calculate mean time to failure of the machine, therefore low index value means disk load needed to be balanced. For this balancing the application running on the machine whose sustainability index is less can be prevented from failure by handling the repairable faults. The application can be resubmitted to a new machine with enough disk space. Figure 5.11 shows the sustainability calculation window.

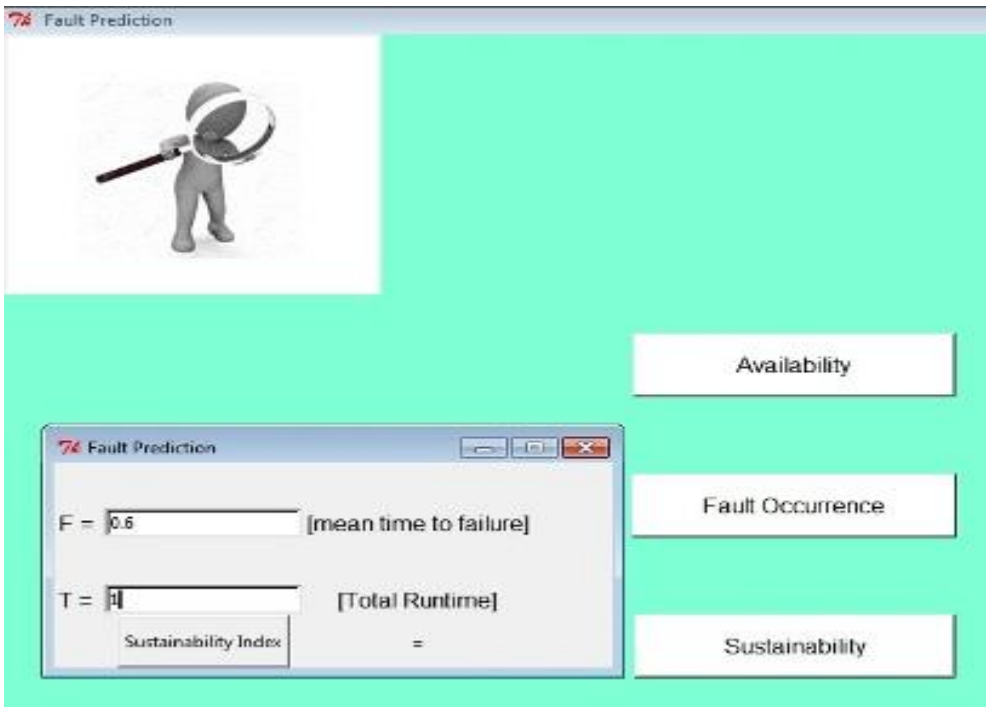

Figure 5.11: Sustainability Index Window

To find sustainability index the sustainability option available on the main window and is clicked then enter the required input values and click on the calculate button the required sustainability index will be displayed. From the calculate value it can be estimated that an application will run smoothly on that machine or not. The sustainability index value will be displayed as shown in Figure 5.13

Sustainability index calculated determines the ability of machine to sustain faults and hence to complete application running on it. Fault prediction helps in finding the faults that can be repaired hence reliability and availability can be improved.

5.3 Results

The interface gives the availability statistics and fault occurrence probability that can be plotted for better understanding as shown in Figure 5.12. The availability graph represents the per day availability status of virtual machine calculated for 10 days. Availability statistics of a machine for particular time duration can be calculated and a threshold value can be decided from the availability and set as criterion for deciding the fault analysis requirement of the machine.

Figure 5.12: Availability statistics

If availability is more than threshold value than machine is less prone to faults and if availability of machine is decreasing as per the threshold value then it requires further inspection. The graph for fault occurrence probability is the representation of number of fault occurrence probability. Fault occurrence distribution is calculated based on the Poisson distribution model with mean value 0.4. From this number of fault occurrence for a machine on some specific day can be predicted. Results are based on the analysis of the machine behavior for the experiment duration. Based on these fault prediction can be done in an efficient manner.

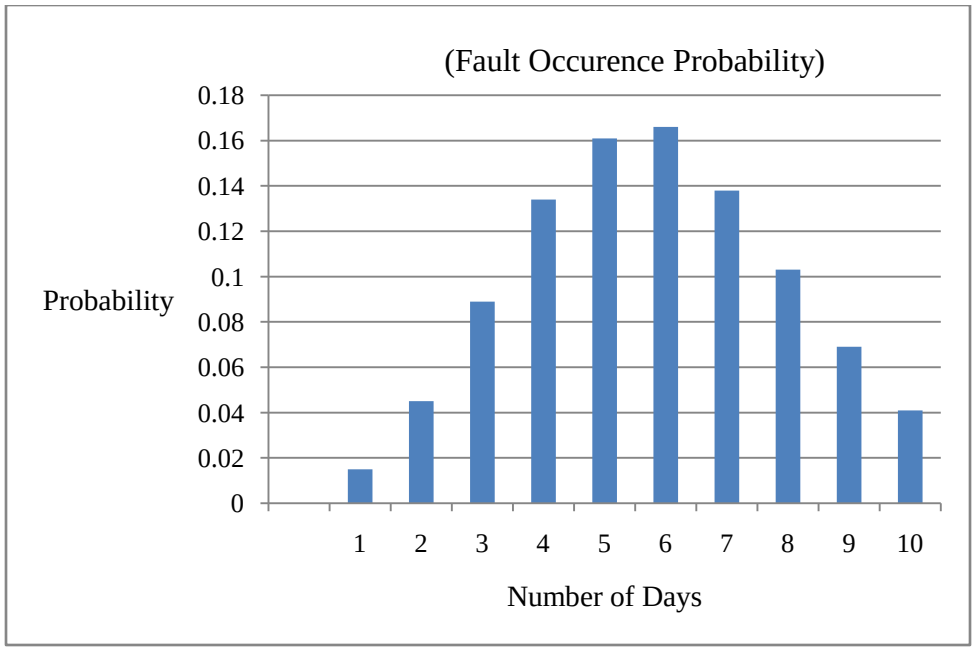

Figure 5.13: Fault Occurrence Probabilities

For the machine whose average fault occurrence is 0.4 per hour probability of 3 faults can be calculated using probability mass function of Poisson distribution.

$$P(3)=\frac{e^{-0.4}0.4^{3}}{3!}$$

$$P(3)=0.089$$

The graph in Figure5.13 is representing the fault occurrence probabilities for a virtual machine with average fault occurrence per unit time equals to 0.4. High value of probability indicated more number of fault occurrence which helps in preventing failure of machine by handling repairable faults in the machine. Hence, the performance of the machine can be improved. Form average fault occurrence meant time to failure can be calculated. When meant time to failure decreases the sustainability index increases and results in improved performance. By reducing the fault injections from 25 to 20 for 100 hour duration mean time to failure also improves from 4 hours to 5 hours and hence sustainability index value increases. Table 5.1 shows the sustainability index, reliability with mean time to failure 4 and 5. Results show improvement in sustainability index value.

Table 5.1: Sustainability Index and Reliability

F	R	Sustainability Index	R%	F	R	Sustainability Index	R%
4	2	2.0	96.0%	5	2	2.5	96.6%
4	3	1.33	91.0%	5	3	1.66	91.6%
4	4	1.0	85.50%	5	4	1.25	86.0%
4	6	0.67	84.80%	5	6	0.82	85.4%
4	8	0.5	89.0%	5	8	0.62	89.6%

Reliability percentage has been improved with number fault occurrence decreases from 25 to 20 in total experiment duration. Value for average fault occurrence per day is decreases form 6 to 5 and hence reliability is improved. As the probability of fault occurrence decrease probability of fault not occurring increases which means Reliability increased. From the experiment performed on the virtual machine there statistical analysis can be done using this statistical analyzer. Fault predictions results in better performance of the machines.

Therefore, the statistical approach followed for fault prediction gives the estimation of fault occurrences and hence proactive fault tolerance can be performed. On knowing the fault occurrence probabilities the steps to prevent application failure and data loss can be taken. Depending on the nature of fault appropriate remedial steps can be performed. Overloading can be prevented with migration, loss of data can be prevented with replication, and resubmission for the application can be done on less fault prone machine.

Chapter 6
Conclusion and Future Scope

This chapter discusses conclusion and future scope of the research initiative taken for doing statistical analysis as to improve the reliability in Cloud environment. The contribution of the proposed approach is also given.

6.1 Conclusion

This thesis work is based on statistical analysis which is an efficient and dependable method for fault predictions in Cloud Computing services. The interface designed for statistical analysis is capable of finding performance statistics for a machine. Virtual machines are monitored periodically and their status information is analyzed for calculating fault probabilities. Machine uptime, downtime and criticality status is used as parameters to find availability statistics. Application running on a machine, with high failure probability and low availability can be prevented from failure by handling the faulty scenarios predicted. Thus, improving the reliability of Cloud Computing and preventing failures in virtual machine. Sustainability index values are improved by decreasing the fault occurrence rate and mean time to failure.

6.2 Thesis Contribution

Contribution made by the proposed thesis work is summarized

- A probabilistic approach for fault prediction and statistical analysis is proposed.
- Fault monitoring for virtual machine is done periodically and parameters for fault prediction are obtained from status information of machine.
- Statistical analysis is done to find fault probabilities, availability, reliability and sustainability index
- A graphical user interface is designed for the proposed approach capable of doing fault analysis based on statistical approach.

6.3 Future Scope

Statistical fault prediction helps in preventing failure. More precision can be added to this approach by implementing the best fit distribution model for fault prediction according to the nature of faults occurring in the system. Real time applications can be benefited with this kind of continuous monitoring and statistical analysis for performance evaluation. The presented approach can be used for statistical analysis of virtual instances provided in public cloud such as Amazon and Microsoft Azure with the help of platform as a service.

References

[1] Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08* (pp. 1-10). IEEE.

[2] Mell, P., &Grance, T. (2011). The NIST definition of cloud computing.

[3] Zissis, D., &Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation computer systems*, *28*(3), 583-592.

[4] Laprie, J. C. (1995). Dependability—its attributes, impairments and means. In *Predictably Dependable Computing Systems* (pp. 3-18). Springer Berlin Heidelberg.

[5] B. Rimal, E. Choi and I. Lumb, "A Taxonomy and Survey of cloud computing Systems," in Fifth International Joint Conference on INC, IMS and IDC, 2009.

[6] "Clouds in IT history - Cloud lounge" [online] Available: http://www.cloud-lounge.org/clouds-in-IT-history.html, [Accessed: 15 May, 2014].

[7] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58.

[8] Höfer, C. N., &Karagiannis, G. (2011). Cloud computing services: taxonomy and comparison. *Journal of Internet Services and Applications*, *2*(2), 81-94.

[9] Sultan, N. (2010). Cloud computing for education: A new dawn. *International Journal of Information Management*, *30*(2), 109-116.

[10] Tsai, W. T., Sun, X., &Balasooriya, J. (2010, April). Service-oriented cloud computing architecture. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on* (pp. 684-689). IEEE.

[11] Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, *28*(1), 75-86.

[12] Dillon, T., Wu, C., & Chang, E. (2010, April). Cloud computing: issues and challenges. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on* (pp. 27-33). IEEE.

[13] Chellappa, R. (1997, October). Intermediaries in cloud-computing: A new computing paradigm. In *Presentation at the INFORMS conference, Dallas, elektronischveröffentlicht: URL: http://meetings2. informs. org/Dallas97/TALKS/MD19. html [Datum des Abrufs: 06.08. 2010]*.

[14] Kaushal, V., & Bala, A. (2011). Autonomic fault tolerance using haproxy in cloud environment. *International Journal of Advanced Engineering Sciences and Technologies*, *7*(2), 222-227.

[15] Bala, A. (2015). Scrutinize: Fault Monitoring for Preventing System Failure in Cloud Computing.

[16] Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, *1*(1), 7-18.

[17] Jhawar, R., Piuri, V., & Santambrogio, M. (2012, March). A comprehensive conceptual system-level approach to fault tolerance in cloud computing. In*Systems Conference (SysCon), 2012 IEEE International* (pp. 1-5). IEEE.

[18] Mili, A., Cukic, B., Xia, T., & Ben Ayed, R. (1999, October). Combining fault avoidance, fault removal and fault tolerance: An integrated model. In *Automated Software Engineering, 1999. 14th IEEE International Conference on.*(pp. 137-146). IEEE.

[19] Latchoumy, P., & Khader, P. S. A. (2011). Survey on fault tolerance in grid computing. *International Journal of Computer Science & Engineering Survey (IJCSES) Vol*, *2*(2011).

[20] Bala, A., & Chana, I. (2012). Fault tolerance-challenges, techniques and implementation in cloud computing. *IJCSI International Journal of Computer Science Issues*, *9*(1), 1694-0814.

[21] Ko, S. R., Lee, S., &Rajan, V. (2013). Cloud computing vulnerability incidents: A statistical overview. *Cloud Security Alliance*.

[22] Leelipushpam, P. G. J., &Sharmila, J. (2013, April). Live VM migration techniques in cloud environment—a survey. In *Information & Communication Technologies (ICT), 2013 IEEE Conference on* (pp. 408-413). IEEE.

[23] Yu, H., & Vahdat, A. (2005). Consistent and automatic replica regeneration.*ACM Transactions on Storage (TOS)*, *1*(1), 3-37.

[24] Khiyaita, A., Zbakh, M., El Bakkali, H., & El Kettani, D. (2012, April). Load balancing cloud computing: state of art. In *Network Security and Systems (JNS2), 2012 National Days of* (pp. 106-109). IEEE.

[25] Li, W., Zhang, P., & Yang, Z. (2012, June). A framework for self-healing service compositions in cloud computing environments. In *Web Services (ICWS), 2012 IEEE 19th International Conference on* (pp. 690-691). IEEE.

[26] Khanghahi, N., & Ravanmehr, R. (2013). CLOUD COMPUTING PERFORMANCE EVALUATION: ISSUES AND CHALLENGES. *Computing*, *1*, 5.

[27] Chen, G., Jin, H., Zou, D., Zhou, B. B., Qiang, W., & Hu, G. (2010, September). Shelp: Automatic self-healing for multiple application instances in a virtual machine environment. In *Cluster Computing (CLUSTER), 2010 IEEE International Conference on* (pp. 97-106). IEEE.

[28] Hwang, S., & Kesselman, C. (2003, June). Grid workflow: a flexible failure handling framework for the grid. In *High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on* (pp. 126-137). IEEE.

[29] Zhao, W., Melliar-Smith, P. M., & Moser, L. E. (2010, July). Fault tolerance middleware for cloud computing. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on* (pp. 67-74). IEEE.

[30] Deng, J., Huang, S. C. H., Han, Y. S., & Deng, J. H. (2010, December). Fault-tolerant and reliable computation in cloud computing. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* (pp. 1601-1605). IEEE.

[31] Tchana, A., Broto, L., & Hagimont, D. (2012, May). Approaches to cloud computing fault tolerance. In *Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on* (pp. 1-6). IEEE.

[32] Yang, C. T., Liu, J. C., Hsu, C. H., & Chou, W. L. (2014). On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *The Journal of Supercomputing*, *69*(3), 1103-1122.

[33] Ward, J. S., & Barker, A. (2012, June). Semantic based data collection for large scale cloud systems. In *Proceedings of the fifth international workshop on Data-Intensive Distributed Computing Date* (pp. 13-22). ACM.

[34] He, Y., Wang, X., Chen, Y., Du, Z., Huang, W., & Chai, X. (2013). A simulation cloud monitoring framework and its evaluation model. *Simulation Modeling Practice and Theory*, *38*, 20-37.

[35] Pervilä, M. A. (2007). Using Nagios to monitor faults in a self-healing environment. In *Seminar on Self-Healing Systems, University of Helsinki* (pp. 1-6).

[36] Gogouvitis, S. V., Alexandrou, V., Mavrogeorgi, N., Koutsoutos, S., Kyriazis, D., &Varvarigou, T. (2012, November). A monitoring mechanism for storage clouds. In *Cloud and Green Computing (CGC), 2012 Second International Conference on* (pp. 153-159). IEEE.

[37] Yigitbasi, N., Iosup, A., Epema, D., &Ostermann, S. (2009, May). C-meter: A framework for performance analysis of computing clouds. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid* (pp. 472-477). IEEE Computer Society.

[38] Wang, C., Xing, L., Wang, H., Dai, Y., & Zhang, Z. (2014). Performance Analysis of Media Cloud-Based Multimedia Systems with Retrying Fault-Tolerance Technique. *Systems Journal, IEEE*, *8*(1), 313-321.

[39] Yang, B., Tan, F., & Dai, Y. S. (2013). Performance evaluation of cloud service considering fault recovery. *The Journal of Supercomputing*, *65*(1), 426-444.

[40] Zheng, Z., & Lyu, M. R. (2009, June). A QoS-aware fault tolerant middleware for dependable service composition. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on* (pp. 239-248). IEEE.

[41] Fu, S. (2009, May). Failure-aware construction and reconfiguration of distributed virtual machines for high availability computing. In *Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on* (pp. 372-379). IEEE.

[42] Polze, A., Troger, P., & Salfner, F. (2011, March). Timely virtual machine migration for pro-active fault tolerance. In *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2011 14th IEEE International Symposium on* (pp. 234-243). IEEE.

[43] Hall, T., Beecham, S., Bowes, D., Gray, D., & Counsel, S. (2012). A systematic literature review on fault prediction performance in software engineering. *Software Engineering, IEEE Transactions on*, *38*(6), 1276-1304.

[44] Simon, T. A., &Dorband, J. Improving Application Resilience through Probabilistic Task Replication.

[45] Fatema, K., Emeakaroha, V. C., Healy, P. D., Morrison, J. P., & Lynn, T. (2014). A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives. *Journal of Parallel and Distributed Computing*, *74*(10), 2918-2933.

[46] Souza, D., Matos, R., Araujo, J., Alves, V., & Maciel, P. (2013). A tool for automatic dependability test in eucalyptus cloud computing infrastructures. *Computer and Information Science*, *6*(3), p57.

[47] Nita, M. C., Pop, F., Mocanu, M., & Cristea, V. (2014). FIM-SIM: Fault Injection Module for CloudSim Based on Statistical Distributions. *Journal of Telecommunications and Information Technology*, (4), 14-23.

[48] Geist, R., & Trivedi, K. (1990). Reliability estimation of fault-tolerant systems: Tools and techniques. *Computer*, *23*(7), 52-61.

List of Publications

Paper Published

Anu and Anju Bala, "Scrutinize: Fault Monitoring for Preventing System Failure in Cloud Computing" in *International Conference on Recent Development in Engineering, Science and Management* (ICRDESM-15) YMCA, New Delhi, India (May 2015) in *International Journal of Innovations & Advancement in Computer Science* IJIACS ISSN 2347–8616 Volume 4, Special Issue

Paper Accepted

Anu and Anju Bala, "Preventing Faults: Fault Monitoring and Proactive Fault Tolerance in Cloud Computing" in *International Conference on Information and Communication Technology for Sustainable Development*, ICT4SD-2015, Ahmedabad, India (July 2015) in Springer AISC series.

YouTube Link

Link to the video of this topic: https://youtu.be./f0YKrzz-7XE