A FAULT TOLERABLE LOAD BALANCING TECHNIQUE IN CLOUD COMPUTING

Dissertation submitted to the Central University of Punjab

For the award of

Masters of Technology

In

Centre for Computer Science and Technology

BY

Shiny

Supervisor

Er. Surinder Singh Khurana

Centre for Computer Science and Technology
School of Engineering and Technology
Central University of Punjab, Bathinda
November, 2014

DECLARATION

I declare that the dissertation entitled “A FAULT TOLERABLE LOAD BALANCING TECHNIQUE IN CLOUD COMPUTING” has been prepared by me under the guidance of Er. Surinder Singh Khurana, Assistant Professor, Centre for Computer Science and Technology, School of Engineering and Technology, Central University of Punjab.

No part of this dissertation has formed the basis for the award of any degree or fellowship previously.

Shiny
Centre for Computer Science and Technology
School of Engineering and Technology
Central University of Punjab
Bathinda -151001.

Date:

CERTIFICATE

I certify that Shiny has prepared her dissertation entitled "A FAULT TOLERABLE LOAD BALANCING TECHNIQUE IN CLOUD COMPUTING ", for the award of M.Tech. degree of the Central University of Punjab, under my guidance. She has carried out this work at the Centre for Computer Science and Technology, School of Engineering and Technology, Central University of Punjab.

Er. Surinder Singh Khurana
(Assistant Professor)
Centre for Computer Science and Technology
School of Engineering and Technology
Central University of Punjab
Bathinda - 151001.

Date:

ABSTRACT

A FAULT TOLERABLE LOAD BALANCING TECHNIQUE IN CLOUD COMPUTING

Name of student:	Shiny
Registration Number:	CUPB/MTECH/SET/CST/2012-13/20
Degree for which submitted:	M.Tech
Name of supervisor:	Er. Surinder Singh Khurana
Centre:	Computer Science and Technology
School of Studies:	Engineering and Technology

Key words:	Cloud Computing, Load Balancing, Fault Tolerance, Virtualization, Dynamic Load Balancing

As the IT industry is growing day by day, the need of computing and data storage is increasing rapidly. The process of this increasing mass of data requires more computer equipment to meet the various needs of the organizations. To better capitalize their investment, the over-equipped organizations open their infrastructures to others by exploiting the Internet and other important technologies such as virtualization by creating a new computing model: the cloud computing. Cloud computing is one of the significant milestones in recent times in the history of computers. However, there are number of technical challenges that need to be tackled which include reliability, resource provisioning, fault tolerance, load balancing and efficient mechanism to increase the service level agreement (SLA) and better use of the resources.

The main purpose of this dissertation report is to provide a preface of the topic and to work on the various issues involved in the field of load balancing and fault tolerance i.e. load computation and the distribution of load. Load balancing and fault tolerance in cloud computing have a great impact on the performance of the system. Good load balancing makes cloud computing more efficient by provisioning of resources to cloud users on demand basis in pay-as-you-say manner. Load balancers are used for assigning load to different virtual machines in such a way that none of the nodes gets loaded heavily or lightly. When many clients request the server simultaneously, server is overloaded which causes fault. There are many load balancing algorithms and fault tolerance techniques in order to settle down these issues, but these techniques further had some drawbacks. Das and Khilar (2013) discussed a load balancing technique for virtualization and fault tolerance in cloud computing (LBVFT) to assign the task to the virtual nodes depending upon the success rates (SR) and the previous load history. This

technique tolerates not only the faults but also reduce the chance of future faults by not assigning tasks to virtual nodes of physical servers whose success rates are very low and loads are very high. But there is still a need to provide an efficient load balancing, load migration, load calculation and fault handling technique to make the VFT model more effective.

Shiny Er. Surinder Singh Khurana

ACKNOWLEDEGEMENTS

Primarily, I would like to express my sincere gratitude to my supervisor Er. Surinder Singh Khurana, for his continuous support, patience, motivation, enthusiasm and intense knowledge. His guidance helped me in writing the thesis.

This dissertation would not have been possible without the help of Prof. Dr. A.K. Jain, CoC, Centre for Computer Science and Technology, Central University of Punjab and the other faculty members of the department (academic and technical) for their encouragement, insightful comments and hard questions. The library and the computer facilities of the university have been vital.

I would also like to show gratitude to Dr. Jai Rup Singh, former Vice-Chancellor, Dr. R.K. Kohli, present Vice-Chancellor of the university, Prof. Dr. P. Ramarao and Prof. Dr. R.C. Sharma, Dean of Academic Affairs and Dean of Examinations respectively. Without their constant help, support and encouragement, this dissertation would not have been possible.

In addition, a special thanks to my fellow mates for the stimulating discussions, exchanges of knowledge and skills, which helped me enriching my experience.

Last but not the least; I would like to thank my parents and my brother for their unconditional support, both financially and emotionally throughout.

Shiny

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Sr. No.	Full Form	Abbreviation
1.	Information Technology	IT
2.	Software as a Service	SaaS
3.	Platform as a Service	PaaS
4.	Infrastructure as a Service	IaaS
5.	National Institute of Standards and Technology	NIST
6.	Personal Computer	PC
7.	Quality of Service	QoS
8.	Service Level Agreement	SLA
9.	Personal Digital Assistant	PDA
10.	Load Balancing	LB
11.	Integrated Development Environment	IDE
12.	Load Balancing for Virtualization and Fault Tolerance	LBVFT
13.	Success Rate	SR
14.	Virtual Machine	VM
15.	Low Latency Fault Tolerance	LLFT
16.	Fault Tolerance	FT
17.	Central Processing Unit	CPU
18.	Operating System	OS
19.	Elastic Compute Cloud	EC2
20.	Simple Object Access Protocol	SOAP
21.	Representational State Transfer	REST
22.	Enterprise Resource Management	ERM
23.	Human Resource Management	HRM
24.	Customer Relationship Management	CRM
25.	Security Device Manager	SDM
26.	Google App Engine	GAE
27.	High Availability Proxy	HAProxy
28.	Index Name Server	INS
29.	Central Load Balancing Decision Model	CLBDM
30.	Load Balancing Min-Min	LBMM

31.	Dual Direction File Transfer Protocol	DDFTP
32.	Virtualization and Fault Tolerance	VFT
33.	Start Time	ST
34.	Finish Time	FT
35.	Standard Edition	SE
36.	Load Balancing strategy for Virtual Storage	LBVS
37.	Opportunistic Load Balancing	OLB
38.	Central Load Balancing Policy for Virtual Machines	CLBVM

CHAPTER 1

INTRODUCTION

1.1. Introduction to Research Area

Information Technology has become ubiquitous in organizations and an imminent key success factor in business. Organizations can create, communicate and collaborate faster, more efficient and reliable than ever before. Technology really has great impact on our society and our daily life in number of different ways. In today's high competitive world-a new paradigm of computing i.e. cloud computing appears to be one of the finest technology that grants permission for the IT world to use the resources comfortably and effortlessly. It provides a comfortable environment for the companies in comparison to the conventional computing. In the last few decades, advancement of cloud computing has transmute how information is stored, exchanged and protected. Cloud computing services are pliable, can smoothly readjust in any environment and curtail the expenditure required for the IT infrastructure. Advancements such as cloud computing, can allow buyers and venture to conveniently use applications, use remote servers to uphold data and operations without investiture and access the intimate files at any computer with Internet access.

Cloud computing is an emerging field of computer science and is a recent trend in IT that moves data away from the desktop and portable personal computers (PC's) into large data centers. The main advantage is that customers don't have to pay for infrastructures, its installation and required man power to handle such infrastructure and maintenance (Jadeja and Modi, 2012). Cloud computing is cheaper than other computing models; zero maintenance cost is involved since the service provider is responsible for the availability of services and clients are free from maintenance and management problems of resource machines. Due to this feature, cloud computing is also known as utility computing or "IT on demand" (Shaikh and Haider, 2011).

1.2. Motivation

Cloud computing comes into existence as one of the latest and biggest advancement in the history of computing world. Various fault tolerance and load balancing approaches has been suggested by number of authors. Due to virtualization there are still many issues that need to be resolved such as recovery, detection, dependability and robustness.The proposed work reduces the chance of faults occurred while balancing the load.

1.3. Problem Statement

Researchers have proposed many load balancing algorithms (Khiyaita et al., 2012) (Sran and Kaur, 2013) (Nuaimi et al., 2012) (Zenon et al., 2011) and fault tolerance techniques (Patra et al., 2013) (Bala and Chana, 2012) (Ganga and Karthik, 2013) but all these techniques still have some limitations such as response time, throughput, etc. In (Das and Khilar, 2013), a load balancing technique for virtualization and fault tolerance in cloud computing (LBVFT) is designed to assign the tasks to the virtual nodes depending on the success rates (SR) and the previous load history to reduce the service time and to increase the system availability. But the main drawback of this approach is that the load balancer (LB) will assign the same task to two virtual nodes having good SR value. Due to this the load is not distributed evenly to all available virtual nodes i.e. there is a huge difference between load of nodes with higher SR value and load of nodes with lower SR value. Hence there is need to improve the technique to make the model more effective.

1.4. Objectives

1) To study and analyze the technique used in (Patra et al., 2013).

2) To design and develop the virtualization and fault tolerance technique.

3) To work upon the performance of discussed approach using CloudSim.

1.5. Methodology

This research work mainly focuses on the reactive fault tolerance with virtualization to achieve high availability of nodes and fault tolerant system.

According to the existing technique, a set of virtual nodes are created from the resources of a physical server according to the demand and the hypervisor can keep the records of which virtual node is created from which server.

Step 1

- Study of existing load balancing algorithm.
- Simulation of existing algorithm on the simulator.

Step 2

- Simulation of the proposed work in order to remove the problem that arises in the existing work.

Step 3

- Statistical analysis of the parameters.

1.6. Dissertation Organization

The overall schema of the chapters is formulated in the following manner:

Chapter 2 gives the concise introduction about the term cloud computing in a nutshell, its advantages, disadvantages and the divergent models.

Chapter 3 reviews the load balancing and fault tolerance techniques used in the cloud computing.

Chapter 4 depicts the review of literature.

Chapter 5 gives the description of the proposed scheme used in the dissertation work.

Chapter 6 describes the various parameters used in the dissertation work and the computed results.

Chapter 7 finally concludes the overall work done in this dissertation.

CHAPTER 2

CLOUD COMPUTING IN A NUTSHELL

Cloud computing is an important landmark in the history of computing. The main objective behind cloud computing is to yield a podium for sharing of resources which incorporates software and infrastructure as a consequence of virtualization. The main fundamental of this concept is to handout storage, computing, and software as a service.

It is known as cloud computing because the data and applications both exist on a “cloud” of web servers. The fundamental aspect of cloud computing is to make a better use of distributed resources, combines them to accomplish immense throughput and to be able to solve large scale computation problems. It is a distributed computing prototype that emphases on providing a spacious range of users with distributed access to scalable virtualized hardware and/or software infrastructure over the internet. It contracts with the virtualization, scalability, interoperability, quality of service (QoS) and the delivery models of the cloud. The idea of cloud computing has notably changed the area of parallel and distributed computing systems (Ray and Sarkar, 2012).

There is no one particular definition of cloud computing. A lot of researchers defined the term cloud computing in their own different forms. Following are the most extensively quoted definitions (Magoules et al., 2013):

NIST: “Cloud Computing is a model for enabling 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.”

Foster: “A large scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.”

Gartner: “A style of computing where scalable and elastic IT capabilities are provided as a service to multiple external customers using Internet technologies.”

Buyya in (Buyya et al., 2011) has defined: “Cloud is parallel and distributed computing system consisting of a collection of inter-connected and virtualized computer that are dynamically provisioned and presented as a one or more consolidated computing resources based on service level agreements (SLA) established through negotiation between the service provider and consumers.

2.1. Cloud Architecture

Cloud computing architecture indicates the components that are required for cloud computing. Cloud computing generally encompasses multiple cloud components interconnecting with each other over a loose coupling mechanism such as messaging queue.

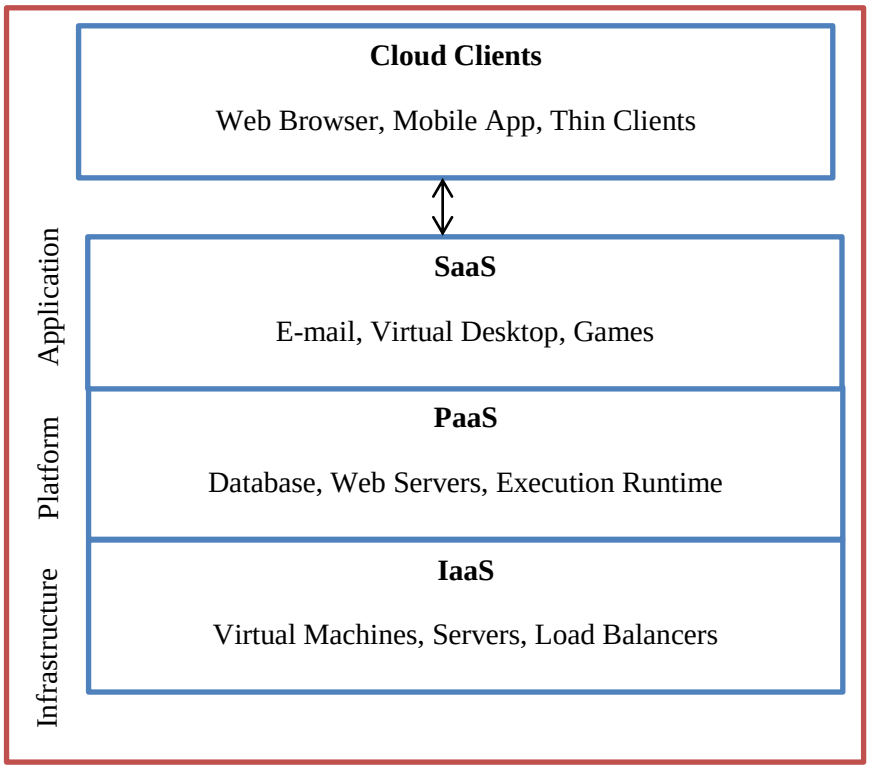

Figure2.1 Architecture of Cloud Computing

Cloud computing architecture can be categorized into two segments: front end and back end. The front end and back end, both are linked with each other over a group of network. The front end is one that the user (client) can examine while the back end is the cloud of the system. Front end subsists client’s computer and the application that is required to access the cloud and the back end has the cloud

computing services alike various computers, servers and data storage (Jadeja and Modi, 2012).

2.1.1. Components of Cloud

Cloud computing is the vigorous provisioning of information technology proficiencies (hardware, software or services) from third parties over a network (Ray and Sarkar, 2012). The core motive behind the attractiveness of cloud computing is due to the acceptance of businesses as the simpler way to implement business processes. A cloud computing model is broadly classified into three major components: clients, datacenter and distributed servers.

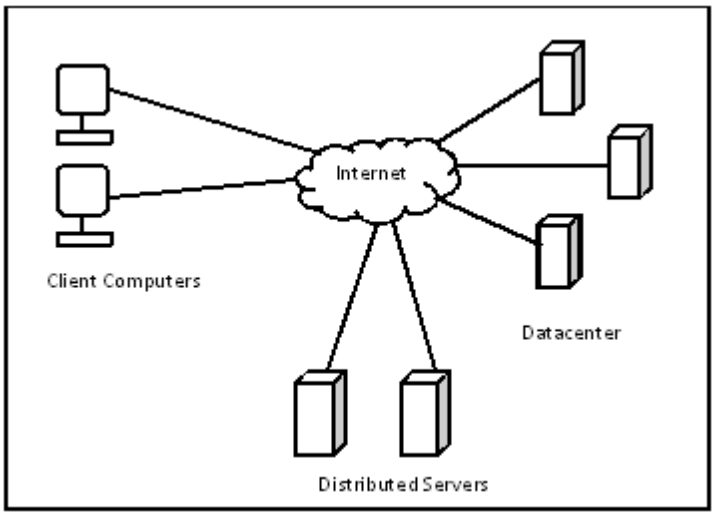

Figure2.2 Cloud Components

Clients

Cloud service consumers are the end users alias clients. Clients are the devices with the help of which the end users can collaborate to maintain their information on the cloud. Clients generally fall into three categories (Srinivas et al., 2012; Ray and Sarkar, 2012):

- **Mobile:** A mobile device encompasses Personal Digital Assistant (PDA) or smartphones, namely, Windows Mobile Smartphone, Blackberry or an iPhone. This category of clients requires high speed and immense level of security.

- **Thin:** Thin clients are computers that neither have hard drives nor have DVD ROM drives and largely depend on the server. They also ensure high level of security, because data cannot be stored in the thin client.

- **Thick:** In general, a thin client also known as lean, zero or slim client is a computer or computer program that bet massively on the other computer server to accomplish its computational aspect using a web browser like Firefox, Google Chrome or Internet Explorer to hook up to the cloud. The main assets of thin clients are listed below:

  1) Lower hardware costs

  2) Lower IT costs

  3) Data Security

  4) Limited power consumption

  5) Less noise

  6) Ease of repair or replacement

Datacenter

The datacenter is a hefty collection of networked computer servers frequently used by many corporations for processing, storage and distributing huge amount of data. It basically could consist of storage, network, and server. It can be a large room in the vault of the building or a room with lot of servers that one can access by dint of Internet (Srinivas et al., 2012). An end user connects to the datacenter to sign up distinct applications (Ray and Sarkar, 2012).

Distributed Servers

A distributed server is the area of computer science, also known as local or workgroup server which vigorously checks the services of their hosts. Distributed server is the component of a cloud that supports a specific group of users on the network. But when using the application from the cloud, the user felt that he/she is using this application from its own machine (Ray and Sarkar, 2012).

2.2. Deployment Models

Deployment model refers to the location and management of the cloud's infrastructure. Cloud services can be deployed in different ways depending upon the organizational structure and provisioning location. The different types of cloud deployment models are: public cloud, private cloud and hybrid cloud.

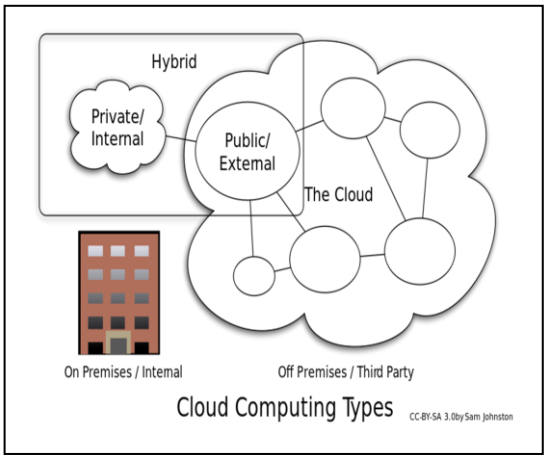

Figure 2.3 Cloud Types (Johnston 2009)

2.2.1. Public Cloud

Public cloud permits user to the cloud through interfacing using web browsers. Usually, public cloud providers like Amazon AWS, Microsoft and Google own and operate the infrastructure and bargain access over the Internet .Users want to pay only for the time period they use the service i.e. pay-per-use (Jadeja and Modi, 2012). Clients do not need to buy hardware to get service and can also scale their user on demand. Public clouds are less secure as compared to other cloud models as all the applications and data on the public cloud are more prone to malignant attacks. This type of cloud is also called external cloud computing and provides all the benefits like cost reduction, scalability and easy maintenance (Sanyal et al., 2013).

2.2.2. Private Cloud

Private cloud or internal cloud computing means using a cloud infrastructure solely by a customer or the whole organization (Nuaimi et al., 2012). Private cloud is build up in such a way whose services are lend by an IT department to those in its own organization without limitation of network bandwidth, security, software, and

also provides more control over the deployment and use. Organization retains the hardware and software infrastructure, copes cloud and restraint access to its resources. As compared to public clouds, where all resources and applications were managed by the service provider, in private cloud these services are pooled together and made available for users at the organizational level (Jadeja and Modi, 2012).

2.2.3. Hybrid Cloud

A hybrid cloud is a cloud computing environment in which an organization offers and accomplishes some remote resources and has others provided externally (Sran and Kaur, 2013). Hybrid cloud is a amalgamation of private and public clouds that interoperate together and enables high level transportability to data and application. In this, a private cloud is associated with one or more external cloud services. The goal is to combine services and data from a variety of cloud models to generate a unified automated and well-managed computing environment (Pathak et al., 2012). It is more secure way to access information over the internet (Roy and Dutta, 2013).

2.2.4. Community Cloud

A community cloud is a multi-tenant cloud service model that is shared among several organizations and that is governed, managed and secured universally by all the participating organizations or a third party service provider. Community clouds are designed for business and organizations working on mutual projects, applications or research which requires a central cloud computing facility for building, managing and executing such projects.

2.3. Service Models

Cloud computing is spreading through enterprises as it enables the agility sought by global organizations. The term services in cloud computing is the concept of being able to use reusable, fine-grained components across a vendor’s network (Kumar et al., 2012). Service delivery in cloud computing mainly comprises of three service models, namely SaaS, IaaS and PaaS.

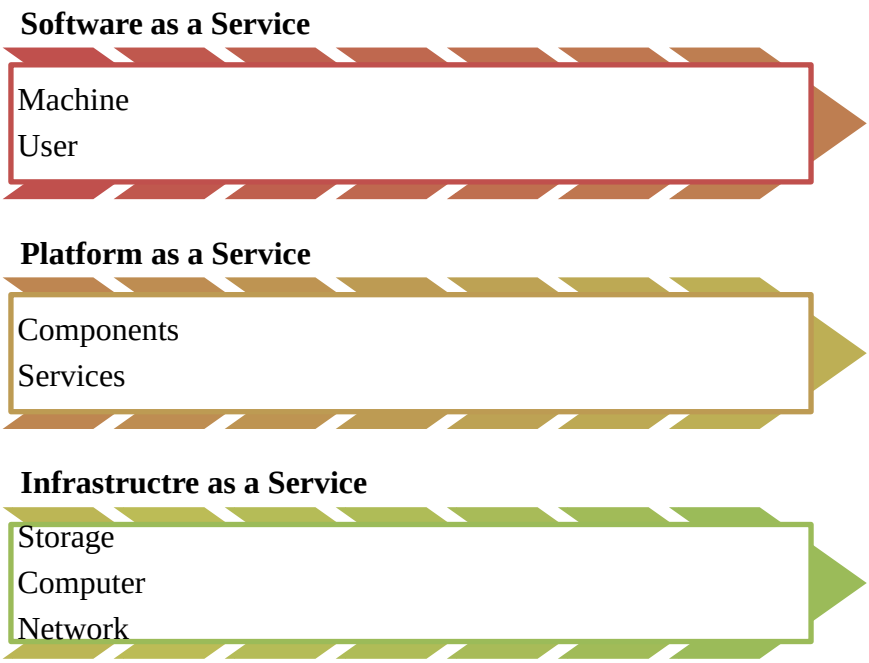

Figure2.4 Cloud Computing Service Models

2.3.1. Software as a Service (SaaS)

SaaS, sometimes referred to as “software on demand”. SaaS provides the software to the users and hence users do not need to install the software on their machines and they can use the software directly from the cloud (Sran and Kaur, 2013). It is the model in which an application is hosted as a service to customers who access it via the Internet (Khiyaita et al., 2012). SaaS is software that is owned, delivered and managed remotely by one or more providers and is offered in a pay-as-per-use manner. SaaS focuses on providing users with business specific capabilities such as e-mail or customer management (Ray and Sarkar, 2012). The typical user of SaaS offering usually has neither knowledge nor control about the underlying infrastructure. One of the example of SaaS provider is Google Apps that provides large suite of web based applications for many business applications including accounting, enterprise resource management (ERP), human resource management (HRM), customer relationship management (CRM) and security device manager (SDM).

2.3.2. Platform as a Service (PaaS)

PaaS is a service model cloud computing, provides a computing platform using the cloud infrastructure. PaaS solutions provide a collection of hardware and software resources that developers can use to build and deploy applications within the cloud. Depending upon their needs, developers may use a Windows-based PaaS solution or Linux-based PaaS (Jamsa, 2013). Platform-based cloud services deliver higher-level services than the infrastructure-based model offers. The client controls the applications that run in the environment, but does not control the operating system, hardware and network infrastructure on which they are running. The provider arranges the network, servers and storage. One of the example of PaaS is Google App Engine that provides clients to run their applications on Google's infrastructure (Ray and Sarkar, 2012). PaaS services include application design, development, testing, deployment and hosting. Other services include team collaboration, web service integration, database integration, security, scalability, storage, state management and versioning. PaaS also supports web development interfaces such as Simple Object Access Protocol (SOAP) and Representational State Transfer (REST), which allows the construction of multiple web services, sometimes called mashups. A downfall to PaaS is a lack of interoperability and portability among providers (Khiyaita et al., 2012). The key examples are GAE (Google App Engine), Microsoft‟s Azure.

2.3.3. Infrastructure as a Service (IaaS)

IaaS, also known as cloud infrastructure services, provides cloud users the infrastructure for various purposes like the storage system and computational resources. It is the process of providing the hardware that is necessary to run an application. IaaS model provides a virtual data center within the cloud. The client need not purchase the required servers, data center or the network resources. The key advantage is that customers need to pay only for the time duration they use the service (Jadeja and Modi, 2012). An IaaS provider offers customers-bandwidth, storage and compute power on an elastic, on-demand basis, over the Internet. The environment of IaaS differs depending upon the size of the organization and the nature of the business (Kumar et al., 2012). One of the examples of IaaS providers is Amazon Elastic Compute Cloud (EC2). It provides

users with a special virtual machine that can be deployed and run on EC2 infrastructure (Ray and Sarkar, 2012).

2.4. Characteristics of Cloud Computing

2.4.1. Technical Aspects

Technical characteristics are the foundation that ensures other functional and economic requirements. Not all technology is advanced, but might be enhanced to realize a specific feature, directly or as a pre-condition.

Virtualization

Virtualization is the scheme of partitioning or break down the resources of a single server into multiple segregated virtual machines. It provides substantial benefits for a computing system including increased utilization, energy saving, rapid deployment, improved maintenance capability, isolation and encapsulation. It is one of the key technologies that can consolidate different infrastructures, so the management of virtual machines needs to be further developed. Furthermore, virtualization enables applications to migrate from one server to another while they are still running without downtime, providing flexible workload management and high availability during planned maintenance or unplanned event.

Multi-Tenancy

Multi-tenancy is the vital technology that clouds use to share IT resources cost-efficiently and firmly. A tenant is any application either inside or outside the company that needs its own secure and exclusionary virtual computing environment. Multi-tenancy equips assistance to the resource providers, for instance, centralization of infrastructure in locations with lower costs and improvement of utilization and efficiency with high peak load capacity (Khiyaita et al., 2012). It is an architecture in which a single instance of a software application serves multiple customers.

Security

Cloud computing means that all of your stuff is on the web. Security and privacy are the biggest concerns about cloud computing which hamper the growth of

cloud. Security issues such as data loss, phishing, and botnet pose serious threats to organizations data and software. For example, data should be fully segregated from one to another and an efficient recovery and replication mechanism should be prepared if a failure occurs. In terms of complexity, the complexity of security is increased when data is distributed over a wider area in multi-tenant systems which are shared by unrelated users (Khiyaita et al., 2012). Moreover, the multi-tenancy model and the pooled computing resources in cloud computing has introduced new security challenges that require novel techniques to tackle with.

2.4.2. Qualitative Aspects

Qualitative aspect refers to the qualities or properties of cloud computing, rather than specific technological requirements. One subjective can be accomplished in multiple ways depending on different providers.

Reliability

Reliability represents the ability to ensure constant system operation without disruption. In simple terms, reliability means a system providing correct output, staying operational and being repairable in a timely fashion. Using redundant sites, the chances of losing data and code dramatically decreases so that cloud computing is suitable for business continuity and disaster recovery. Reliability is a particular QoS requirement, focusing on prevention of loss (Khiyaita et al., 2012).

Availability

Availability refers to a relevant ability that satisfies specific requirements of the outsourced services. In many cases, QoS metrics such as response time and throughput must be guaranteed, so as to ensure advance quality of cloud users.

Elasticity

Elasticity means that the provision of services is elastic and adaptable, which allows the user to request the service near real-time without engineering for peak loads. Cloud services can be rapidly and elastically provisioned, in some cases automatically, to quick scale out and speedily released to quickly scale in. The services are measured exquisitely, so that the amount of offering can perfectly match a consumer’s usage. Performance is examined and consistent.

2.4.3. Economic Aspects

Economic features make cloud computing distinct, compared with other computer paradigms. In a merchandising environment, service offerings are not limited to an exclusive technological perspective, but extend to a broader understanding of business needs.

Pay-as-you-go

The term “cloud computing” refers to the on-demand delivery of IT resources via the Internet with pay-as-you-go pricing. Pay-as-you-go is a utility computing billing method which means the users pay according to the actual consumption of resource. Cloud computing reduces the cost of infrastructure maintenance and acquisition, so it can help enterprises; especially small to medium sized, to reduce time to market and get return on investment. Conventionally, users have to equipped with all software and hardware infrastructure before computing starts, and maintain it during the computing process.

Energy Efficiency

Energy efficiency is due to the ability of clouds to reduce the consumption of unused resources. Computers are administrated centrally, so additional costs of energy consumption as well as carbon emission can be better controlled than in uncooperative cases. Additionally, green IT issues are subject to both software stack and hardware level.

Operational Expenditure

The infrastructure is consistently provided by a third-party and does not need to be purchased for one-time or unusual intensive computing tasks, so it’s easier for the users to enter the computing world. Nominal or no IT skills are required for the implementation.

2.5. Disadvantages

Migration

Migration problem is one of the big issue about cloud computing. If you want to move from one cloud to another i.e. from one hosting provider to another, have to

face more problems. It's not easy to move to another hosting provider because the migration process will take time to transfer files.

Requires a constant internet connection

It is the panic situation for business owner, when site goes offline for some time. This makes your business dependent on the reliability of your Internet connection. If you do not have an Internet connection, you cannot access anything even your on data. A dead Internet connection means no work. Web based applications often require a lot of bandwidth to download (Das and Khilar, 2013). Even Amazon, Google and Apple websites faced this problem. There are two ways to mitigate this risk (Kumar et al., 2013):

- Make sure that you are using an enterprise class Internet connection: Enterprise resource connections are more expensive but provide much better fault tolerance and repair service that consumer-class connections do.
- Provide redundant connections if you can: If one connection fails, traffic can be rerouted through alternative connections.

2.6. Virtualization

Virtualization simply means what is not real, but gives all the facilities of a real. It is a technique, which allows sharing single physical instance of an application or resource among multiple organizations or tenants. Virtualization is related to cloud, because using virtualization an end user can use different services of cloud. Creating a virtual machine over the existing operating system and hardware is referred as hardware virtualization. Computer virtualization refers to the abstraction of computer resources, such as the process of running two or more logical computer systems on one set of physical hardware. The machine on which the virtual machine is created is known as host machine and the virtual machine is referred to as guest machine. This virtual machine is managed by a software or firmware, which is known as hypervisor. With virtualization, a system administrator could combine several physical systems into virtual machines on one single, powerful system, thereby unplugging the original hardware and reducing power and cooling consumption.

2.6.1. Types of Virtualization

Full virtualization

Full virtualization uses hypervisor, which interacts with the physical servers CPU and disk space as shown in figure 2.5. The hypervisor keeps each virtual server completely independent of the other virtual servers running on the physical machine. Further there are two types of full virtualization:

- Native Virtualization- With the native or bare metal virtualization, the hypervisor runs directly on the intrinsic hardware, without a host operating system (OS).
- Hosted Virtualization- With hosted virtualization, the hypervisor runs on the top of the host OS. The host OS can be any usual operation system, such as Linux, Windows or MacOS.

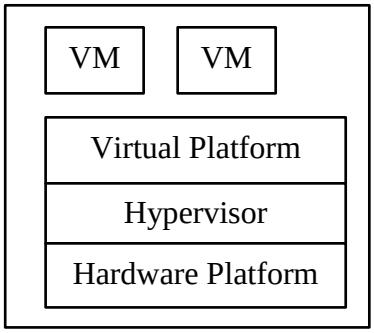

Figure2.5 Full Virtualization

Para virtualization

Para virtualization relates to communication between the guest OS and the hypervisor to improve performance and efficiency. It involves the modifying OS kernel to replace non-virtualizable instructions with hyper calls that communicate directly with the virtualization layer hypervisor. The main advantage of Para virtualization is lower virtualization overhead, but the key advantage of Para virtualization is that it varies substantially depending upon the workload.

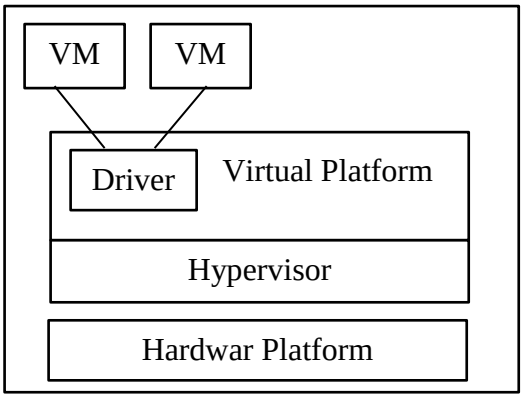

Figure2.6 Para Virtualization

CHAPTER 3

LOAD BALANCING AND FAULT TOLERANCE TECHNIQUE

3.1. Load Balancing

As cloud computing is growing rapidly and clients are demanding more results and better services, load balancing for cloud has become a very important and interesting concept. Load balancing is a new technique that facilitates networks and resources by providing a maximum throughput with minimum response time (Chackzo et al., 2011).

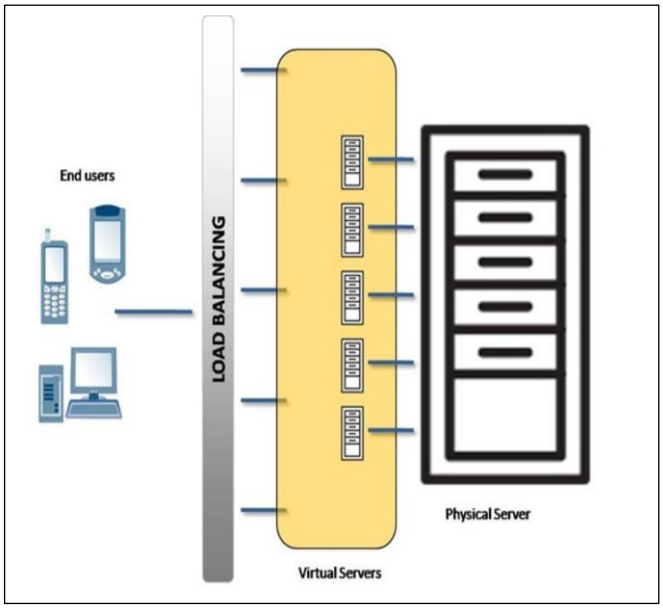

Figure3.1 Load Balancers

It is a generic term used for distributing the workload across one or more servers, network interfaces, hard drives or other computing resources to enhance both resource utilization and job response time. Load balancing ensures that all the processors in the system or every node in the network does approximately the equal amount of work at any instant of time. While balancing the load, certain types of information such as number of jobs waiting in queue, job arrival rate, CPU processing rate and so forth at each processors, may be exchanged among the processors for improving the overall performance (Jadeja and Modi, 2012).

Load balancing is one of the main issues associated with the concept of cloud computing. The load can be memory, CPU capacity, network or delay load. It is always required to share the load among the different virtual nodes in order to improve the resource utilization and better performance (Desai and Prajapati, 2013).

Load balancers are used for assigning load to different virtual machines in such a way that none of the nodes gets loaded heavily or lightly. The load balancing needs to be done properly because failure in any one node can lead to unavailability of data (Moharana et al., 2013). The load balancer accepts multiple requests from the client and distributes each of them across multiple computers or network devices based on how busy the network device is. Load balancing helps to prevent a server or network device from getting vanquish with requests and also helps in distribution of work. If the load balancer is not available the client can wait for long time and process their request in the particular server only where the client give request (Roy and Dutta, 2013).

3.1.1. Goals of load balancing

The main aim of load balancing is as follows:

- To maintain the firmness of the system
- To significantly improve the performance of the system
- To have a backup plan in case of any failure
- To encompass the future modification of the system

3.1.2. Types of Load Balancing Algorithms

The load balancing algorithms are classified into two categories: static and dynamic load balancing.

Static Load Balancing Algorithm

In static load balancing algorithms, the performance of the processors is determined at the beginning of the execution, it does not depend on current state of the system. The goal of static load balancing is to reduce the overall execution time of a synchronous program while minimizing the communication delays. These algorithms are mostly suitable for homogeneous and stable environments and can

produce very good results (Nuaimi et al., 2012).Some of the examples of static load balancing algorithms are: Round Robin algorithm, Randomized algorithm and Threshold algorithm.

Dynamic Load Balancing Algorithm

In dynamic load balancing algorithm, the decisions on load balancing are based on the current state of the system; no prior knowledge is needed. The main advantage of dynamic load balancing is that if any node fails, it will not stop the system; it will only affect the performance of the system. These algorithms are more flexible than static algorithms, can easily adapt to changes and provide better results in heterogeneous and dynamic environments (Nuaimi et al., 2012). Dynamic load balancer uses policies for keeping the track of updated information. There are four policies for dynamic load balancers: transfer policy, selection policy, location policy and information policy (Moharana et al., 2013).

3.2. Dynamic Load Balancing

A load balancing algorithm which is dynamic in nature does not consider the previous state or behavior of the system; it depends upon the current behavior of the system. The important things to examine while developing such algorithms are: load estimation, comparison of load, stability of different system, performance of the system, interaction between the nodes, nature of work to be transferred, selection of nodes, etc. (Roy and Dutta, 2013). The task of load balancing is shared among distributed nodes. In a distributed system, dynamic load balancing can be done in two different ways: distributed and non-distributed.

3.2.1. Distributed Dynamic Load Balancing Algorithm

The distributed dynamic load balancing algorithm is executed by all nodes present in the system and the task of scheduling is shared among them (Moharana et al., 2013). The interaction among the nodes to achieve load balancing can take two forms: cooperative and non-cooperative. In the cooperative, the nodes work together to achieve the common objective, for example to improve the response time, etc. whereas in non-cooperative, each node works independently towards a goal local to it, for example to improve the response time of a local task, etc. (Kumar 2013). In dynamic load balancing systems, the nodes can interact with

each other generating more messages as compared to non-distributed ones because each node in the system needs to interact with every other node (Moharana et al., 2013).

3.2.2. Non-Distributed Load Balancing Algorithm

In the non-distributed or undistributed, the nodes work individually in order to achieve a common goal. Non-distributed dynamic load balancing algorithms are further classified into two: centralized and semi-centralized.

Semi-distributed Dynamic Load Balancing

In semi-distributed dynamic load balancing, the nodes of the system are partitioned into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in each cluster by appropriate election technique which takes care of load balancing within that cluster. Therefore, the load balancing of whole system is done via the central nodes of each cluster (Kumar 2013).

Centralized Dynamic Load Balancing

In centralized dynamic load balancing, the algorithm is executed only by a single node in the whole system i.e. central node. This node is completely responsible for load balancing of the whole system and rest of the nodes interacts only with the central node (Kumar 2013).

3.3. FAULT TOLERANCE

Undoubtedly, cloud computing is a buzzword in the IT industry today. Due to rapid growth of cloud computing, the need of fault tolerance in cloud is an important key factor. Fault tolerance is a configuration that prevents a computer or network device from failing in the event of unexpected problem or error such as hardware failure, link failure, unauthorized access, variations in the configuration of different systems and system running out of memory or disk space. In order to minimize failure impact on the system and application execution, failures should be anticipated and proactively handled. There are various types of faults which can occur in the cloud computing. Based on fault tolerance policies, there are two fault tolerance techniques: reactive and proactive fault tolerance. The main benefits of

implementing fault tolerance in cloud computing includes failure recovery, lower cost, improved performance metrics etc. (Patra et al., 2013) (Ganga and Karthik, 2013) (Bala and Chana, 2012).

3.3.1. Categories of Fault Tolerance

3.3.1.1. Reactive Fault Tolerance

Reactive fault tolerance policies reduce the effect of failures on the application execution when the failure effectively occurs. Following are the various techniques based upon these policies (Patra et al. 2013) (Ganga and Karthik, 2013) (Bala and Chana, 2012):

Check pointing/Restart

It is an efficient task level fault tolerance technique for long running applications. In this scenario after doing every change in the system a check pointing is done. When a task fails, it is allowed to be restarted from the recently checked pointed state rather than from the beginning.

Job Migration

Sometimes it happened that due to some reason a job cannot be completely executed on a particular machine. During failure of any task, it can be migrated to another machine. This technique can be implicated by using HA Proxy.

Replication

Replication simply means copy. Various task replicas are run on different resources for the execution to succeed till the entire replicated task is not crashed. It can be implemented using tools like HA Proxy, Hadoop and Amazon EC2 etc.

3.3.1.2. Proactive Fault Tolerance

The main principle behind proactive fault tolerance technique is to avoid recovery from faults, errors and failures by predicting them and proactively replace the suspected components and other working components. Based upon these policies, the various proactive fault tolerance techniques are:

Preemptive migration

Preemptive migration relies on a feedback loop control mechanism where the applications are constantly monitored and analyzed.

Self-Healing

In self-healing, a big task can be divided into small parts. This multiplication can be done for better performance. When multiple instances of an application are running on various virtual machines, it automatically handles failure of application instances.

Rejuvenation

Software rejuvenation restarts the operating environment of a task in order to prevent failures. It is a technique that designs the system for periodic reboots. It restarts the system with clean state and helps to fresh start.

CHAPTER 4

REVIEW OF LITERATURE

Cloud computing (Natarajan 2013) is new way of delivering IT services. The cloud market is segmented into public cloud, private cloud and hybrid cloud. The public cloud is often segmented into IaaS (Infrastructure as a Service), SaaS (Software as a Service) and PaaS (Platform as a Service). A lot of work has been done in the field of load balancing and fault tolerance for cloud computing. But still there are number of issues in these areas to solve. A number of researchers have discussed various load balancing algorithms in (Ray and Sarkar, 2012), (Zenon et al., 2011) (Sran and Kaur, 2013) (Nuaimi et al., 2012) (Roy and Dutta, 2013) (Moharana et al., 2013) (Kumar 2013) and fault tolerance techniques in (Patra et al., 2013) (Ganga and Karthik, 2013) (Bala and Chana, 2012) (Das and Khilar, 2013).

Jadeja and Modi (2012), gives the brief history of cloud computing, its architecture, advantages and various categories of clouds. In order to deploy a cloud computing solution, the main effort is in determining the type of cloud to be enforced. There are three types of cloud namely public cloud, private cloud and hybrid cloud. In this work the researcher also discusses the issues of cloud computing-security and privacy. There are number of ways in order to solve these problems, one is the proper utilization of different authentication modes and the second one is authorization, so that the users does not face any kind of difficulty and can only choose the data that is compatible with their job. Reliability is also one of the major issues.

Nuaimi et al. (2012), discusses the multiple algorithms of load balancing for Cloud Computing like MapReduce algorithm Central Load Balancing Decision Model (CLBDM), Index Name Server (INS), Dual Direction FTP, Load Balancing Min-Min (LBMM) algorithm etc. and the challenges such as spatial distribution of the cloud nodes, algorithm complexity, point of failure etc. that must be addressed to provide the most suitable and efficient load balancing algorithms. The researcher also discusses the advantages and disadvantages of the algorithms. Then compared the existing algorithms based on the challenges discussed. In this,

DDFTP algorithm is used in order to provide efficient load balancing and better resource utilization. The main drawback of this algorithm is it relies on full replication of the files on multiple sites, which wastes storage resources.

Zenon et al. (2011), demonstrate various load balancing techniques to obtain measurable improvements in resource utilization and availability of cloud computing environment. Various models and rules can be applied to load balancers, however these should be based on the scenario the load balancer will be applied. While cThe network structure or topology should be taken into account when creating the logical rules for the load balancer. The load balancing in the distributed networks has been improved by using message oriented architecture.

Sran and Kaur (2013), demonstrates that the load balancing is one of the leading issue in the field of cloud computing. It distributes the dynamic workload among all the virtual nodes so as to attain high resource utilization ratio and makes the surety that every computing resource is distributed effortlessly. There are numerous algorithms of load balancing discussed by the author, but among all none of the algorithm is perfect and solves all the issues related to cloud computing. So there is a need to develop an adaptive and cost effective load balancing algorithm that is best suited for variegated environment.

Kumar et al. (2013), gives an overall description of load balancing, various types of load balancing algorithms used in the cloud computing especially the dynamic load balancing algorithm and the different policies related to it. But these techniques still needs some improvement.

Argha Roy et al. (2013), discussed the dynamic load balancing technique used in the cloud computing. The algorithm that is proposed by the author can axiomatically monitor the load with the help of load balancer. The data replication, job migration and the static load balancing is avoided. The CPU and Memory can be utilized properly and the reliable VM in the cloud pool can be identified. But till date there is no such algorithm that fully distributes the load of the system.

Desai and Prajapati (2013), describes the concept of virtualization, load balancing and their types. Some qualitative metrics can be improved for the betterment of load balancing in cloud computing such as throughput, fault tolerant,

overhead, response time, migration time, resource utilization, scalability and performance. The main purpose of load balancing is to satisfy the customer requirement by distributing load dynamically among the nodes and to make maximum resource utilization by reassigning the total load to individual node. This ensures that every resource is distributed efficiently and evenly. So the performance of the system is increased.

In **(Ganga and Karthik, 2013)** K. Ganga *et al.* presented a fault tolerance technique to prevent the loss due to faults. Fault tolerance is concerned as a setup or to enable a system to tolerate software faults in the system after its development. A Scientific Workflow System provides mechanism to gracefully handle the resource failure. The different fault tolerance techniques in cloud computing are discussed and focus mainly on scientific workflow based on task replication and simulation of cloud computing systems.

Anju Bala (2012), discuss the fault tolerance techniques, challenges and tools used for implementing fault tolerance techniques in cloud computing. Based on fault tolerance policies various fault tolerance techniques can be used that can either be task-level or workflow level such as checkpointing/restart, replication, job migration, SGuard, Retry, Software rejuvenation, Proactive fault tolerance using self-healing etc. Cloud virtualized system architecture is also proposed based on HAProxy. Autonomic fault tolerance is implemented dealing with various software faults for server applications in a cloud virtualized environment. Data replication technique is implemented on virtual environment. The experimental results are obtained that validate the system fault tolerance. But this technique still needs some improvement.

Kansal and Chana (2012) also discussed various existing load balancing approaches used in cloud computing namely VectorDot, Carton, LBVS, CLBVM, Event-driven, Scheduling Strategy on LB of VM resources, Compare and balance, Task Scheduling based on LB, Honeybee Foraging Behavior, Biased Random Sampling, Server-based LB for Internet distributed services, Join-Idle Queue and Lock-free multiprocessing solution for LB. The main emphasize of these techniques is on improving the performance, service response time and reducing

associated overhead. But none of these techniques has considered the energy consumption and carbon emission factors.

Randles et al., (2010) presents three algorithms viz. HoneyBee Foraging Behaviour, Biased Random Sampling and Active Clustering, for distributed load balancing in large scale Cloud systems. The main aim is to give the comparative study of the above three approaches, demonstrating distributed algorithms for load balancing.

Jhawar et al., (2012) designed a framework that allows the service provider to integrate its system with the existing Cloud infrastructure and provides the basis to generically realize the approach in delivering fault tolerance as a service.

Patra et al (2013), also discusses the various fault tolerance techniques that are used to predict the failures and take an appropriate action before failures actually occur. Various proposed models such as low latency fault tolerance (LLFT), Candy, Vega-Warden, FT-cloud, Magi-Cube etc. for fault tolerance are discussed and compared on the basis of metrics for fault tolerance in cloud. In the present scenario, there are number of fault tolerance models which provide different fault tolerance mechanisms to enhance the system. But there are some drawbacks that the researcher cannot fulfill. So there is a possibility to overcome the drawbacks of all previous models and try to make a compact model which will cover maximum fault tolerance aspect.

Das and Khilar (2013) discussed a load balancing technique for virtualization and fault tolerance in cloud computing (LBVFT) to assign the task to the virtual nodes depending upon the success rates (SR) and the previous load history. This technique tolerates not only the faults but also reduce the chance of future faults by not assigning tasks to virtual nodes of physical servers whose success rates are very low and loads are very high. But there is still need to provide an efficient load balancing, load migration, load calculation and fault handling technique to make the VFT model more effective.

In **(Wickremasinghe et al., 2010)**, the authors discussed the detailed explanation of how to use the CloudSim simulator. The comprehensive view of the layered

architecture of the CloudSim, design and implementation of the simulatordifferent classes used in the simulator.

Sharma and Banga (2013) proposed an efficient and enhanced algorithm for balancing the load in cloud computing that can maintain the load balancing and provide better improved strategies through efficient job scheduling and modified resource allocation techniques. They proposed an algorithm in which live migration of load is done in virtual machine to avoid the underutilization and hence improving the data transfer cost. The results were discussed based upon the existing Equally Spread Current Execution, Round Robin, Throttled and the new proposed algorithm.

Hung et al., (2012) proposed a LB3M scheduling algorithm which combines minimum completion time and load balancing strategies. LB3M can provide efficient utilization of computing resources, maintain the load balancing in cloud computing environment and can achieve better load balancing and performance than other algorithms, such as MM and LBMM.

Pathak et al., (2012) worked on load balancing using Honey Bee algorithm. They discussed various parameters like fault tolerance performance, response time, scalability, throughput, resource utilization, migration time and associated overhead, but for an energy-efficient load balancing, metrics like energy consumption and carbon emission should also be considered. The different techniques used for balancing the load can also be discussed in this paper such as Biased Random Sampling, Honey Foraging Behavior, Join-Idle-Queue, a Lock-free multiprocessing solution for LB, Decentralized Content Aware Load Balancing, Server Based Load Balancing for Internet Distributed Services, Scheduling Strategies on Load Balancing of Virtual Machine Resources, Central Load Balancing Policy for Virtual Machines, LBVS (Load Balancing strategy for Virtual Storage), 2-phase Load Balancing Algorithm, A Task Scheduling Algorithm Based on Load Balancing and Active Clustering.

Wang et al., (2010) advanced a two-phase scheduling algorithm incorporates OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms beneath the three-level cloud computing architecture that can freely balance the load of the system in a systematic manner. In the above study, LBMM

scheduling algorithm is modified from Min-Min scheduling algorithm that can make the minimum execution time of each task on cloud computing environment and the OLB scheduling algorithm is used to attempt each node keep busy in order to achieve the goal of load balancing.

Tchana et al., (2012) discusses two approaches of fault tolerance, first one consists in leaving the responsibility of FT management to one cloud provider and the second one consists in sharing the responsibility between the two cloud participants. They also mentioned the various situations in which there is the chance for the occurrence of fault in different levels of the cloud i.e.application level, virtualization level and hardware level.

Chandrakala and Sivaprakasam (2013) proposed a dynamic load balancing technique so as to evade the problem of fault tolerance. The technique checks the utilization of the CPU, if there is less CPU utilization, then the algorithm can respond to the client request otherwise the request is deviated to another server by means of load balancer. The algorithm used in this approach can automatically auditor the load with the use of load balancer. The main focus is to avoid data replication, job migration and the static load balancing.

CHAPTER 5

PROPOSED WORK

This section gives the detailed description of the proposed model and the simulation parameters used in dissertation work.

5.1. Proposed Scheme

The proposed technique i.e. discussed in this report namely, A Fault Tolerable Load Balancing Technique in Cloud Computing, tries to get the better results in comparison to the existing technique-Load Balancing Technique for Virtualization and Fault Tolerance.

1. Set SR=0.5, n1= 1,n2= 2;
[where SR= Success Rate
n1= number of times the virtual node of the server gives successful results
n2= number of times load balancer assigns the job to a virtual node]
2. Get request from the client.
3. Assign the job X to the virtual machines namely V1 and V2based upon the SR values of the nodes;
[where V1= virtual machine having higher SR value
V2= virtual machine having lower SR value]
4. if (job X successfully executed on VM1)
{
n1= n1+1
n2= n2+1
SR(VM1)= n1/n2
Upgrade table
}
else
{
n2= n2+1
SR(VM1)=n1/n2
Upgrade table
}
5. if (job X successfully executed on VM2)
{

```
                n1= n1+1
                n2= n2+1
                SR(VM2)= n1/n2
                Upgrade table
        }
        else
        {
                n2= n2+1
                SR(VM2)=n1/n2
                Upgrade table
        }
6. if (there is any other request)
        {
          Go to step 3.
        }
7. End of algorithm.
```

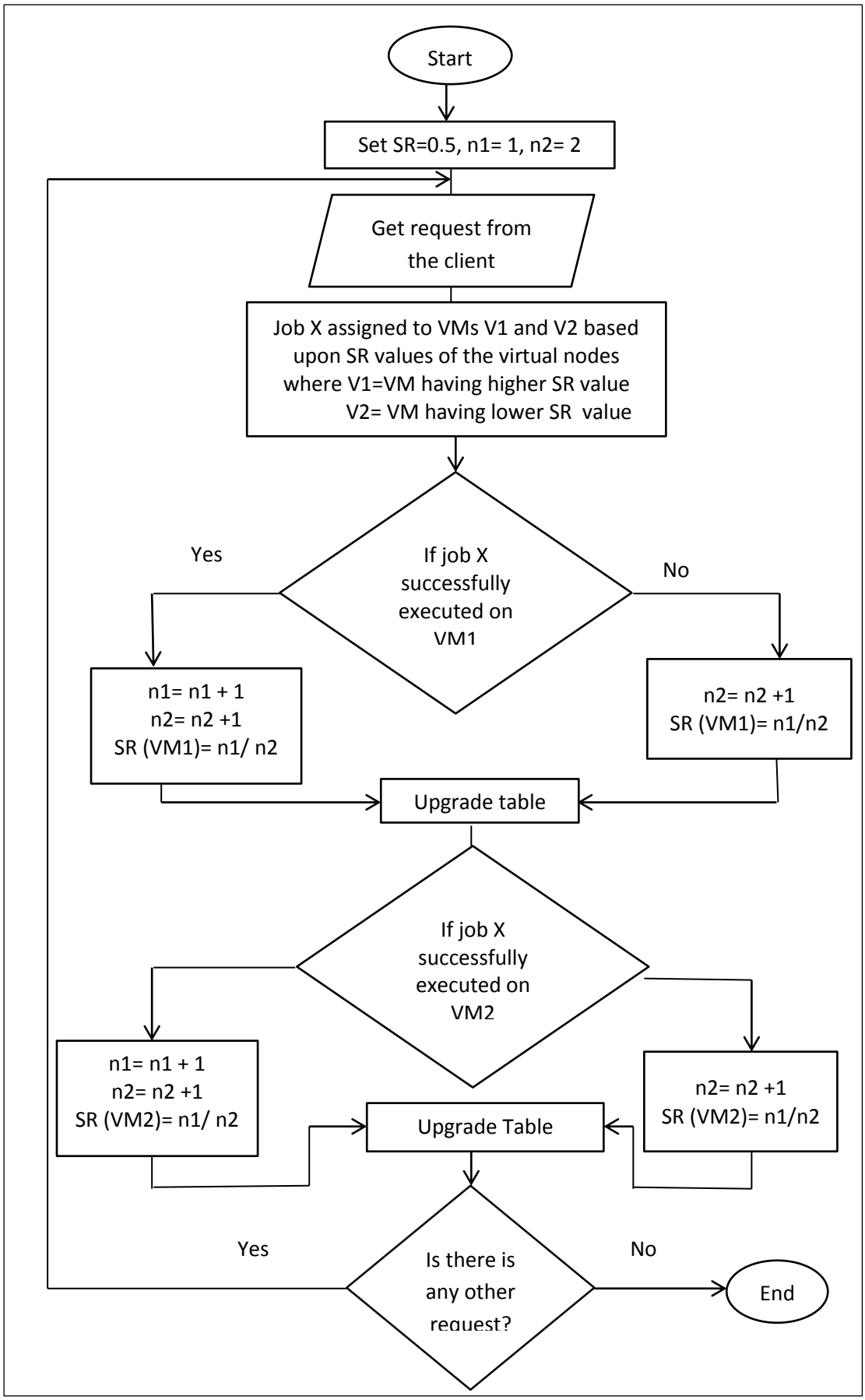

Figure 5.1: Flow Chart of Proposed Work

First of all, apply the success rate computation algorithm discussed in the dissertation work. Depending upon the calculated value of success rate, select the available nodes. After that submit the tasks to the selected nodes. In this new approach, the selection of nodes is based on two ways; first is the nodes with success rate in increasing order and secondly in their decreasing order.

In this approach, the client first submits the task. With the help of hypervisor, virtualization is performed. A hypervisor is an operating system, also known as virtual machine manager that allows multiple operating systems to share a single host. It controls the processor and allocates what is necessary for each operating system without disturbing the guest operating systems. A hypervisor also keeps the history of which virtual machine is linked to which physical machine and maintains the record. After that it is role of the load balancer to balance the load among all the available virtual nodes according to the success rate. And fault handler takes the responsibility if any node fails, the whole process repeats again and again until the status of the node is success.

CHAPTER 6

SIMULATION AND RESULTS

This chapter discusses about the various simulation parameters used in dissertation work and the results calculated after implementing the proposed technique and its comparison with the existing technique.

The simulation work is done in CloudSim 3.0.3 with Ecllipse IDE (Integrated Development Environment) as a Java SE platform. In totality three virtual nodes (VM) and their corresponding success rate (SR) values has been considered.

6.1. Simulation Parameters

The proposed approach is simulated in the CloudSim with Eclipse and the various parameters used for the simulation are discussed in Table 6.1.

Table 6.1: Simulation Parameters

Parameters	Values
VM RAM	512 MB
VM (MIPS)	250
VM Size	10000 MB
No. of CPUs	1
No. of servers	1
No. of VMs	8

6.2. Results

In the table 6.2, L1 to L88 are the loads of the virtual nodes. It is presumed that there are 8 virtual nodes and their corresponding SR values and loads are taken. Here it is examined that there are 10 cycles for each virtual node. The proposed approach will allocate the jobs only to those virtual nodes which are having good SR value and load history.

Table 6.2: Simulation Results

Cycles	VM1		VM2		VM3		VM4		VM5		VM6		VM7		VM8	
	Load	SR	Load	SR	Load	SR	Load	SR	Load	SR	Load	SR	Load	SR	Load	SR
Start	L1	0.5	L2	0.5	L3	0.5	L4	0.5	L5	0.5	L6	0.5	L7	0.5	L8	0.5
1	L9	0.7	L10	0.6	L11	0.5	L12	0.4	L13	0.3	L14	0.52	L15	0.42	L16	0.63
2	L17	0.66	L18	0.75	L19	0.66	L20	0.74	L21	0.68	L22	0.675	L23	0.78	L24	0.635
3	L25	0.62	L26	0.68	L27	0.64	L28	0.616	L29	0.64	L30	0.65	L31	0.626	L32	0.69
4	L33	0.667	L34	0.6	L35	0.76	L36	0.71	L37	0.618	L38	0.68	L39	0.72	L40	0.65
5	L41	0.699	L42	0.642	L43	0.76	L44	0.7	L45	0.69	L46	0.638	L47	0.71	L48	0.69
6	L49	0.7	L50	0.76	L51	0.615	L52	0.74	L53	0.72	L54	0.69	L55	0.7	L56	0.63
7	L57	0.8	L58	0.7	L59	0.614	L60	0.69	L61	0.629	L62	0.67	L63	0.72	L64	0.681
8	L65	0.81	L66	0.63	L67	0.59	L68	0.623	L69	0.4	L70	0.72	L71	0.63	L72	0.58
9	L73	0.83	L74	0.68	L75	0.641	L76	0.54	L77	0.51	L78	0.67	L79	0.59	L80	0.82
10	L81	0.83	L82	0.67	L83	0.61	L84	0.61	L85	0.58	L86	0.56	L87	0.61	L88	0.72

- **Fault Tolerance:** In simple terms, it is the ability of a system to operate ceaseless or reciprocate elegantly even in case of any hardware or software failure.

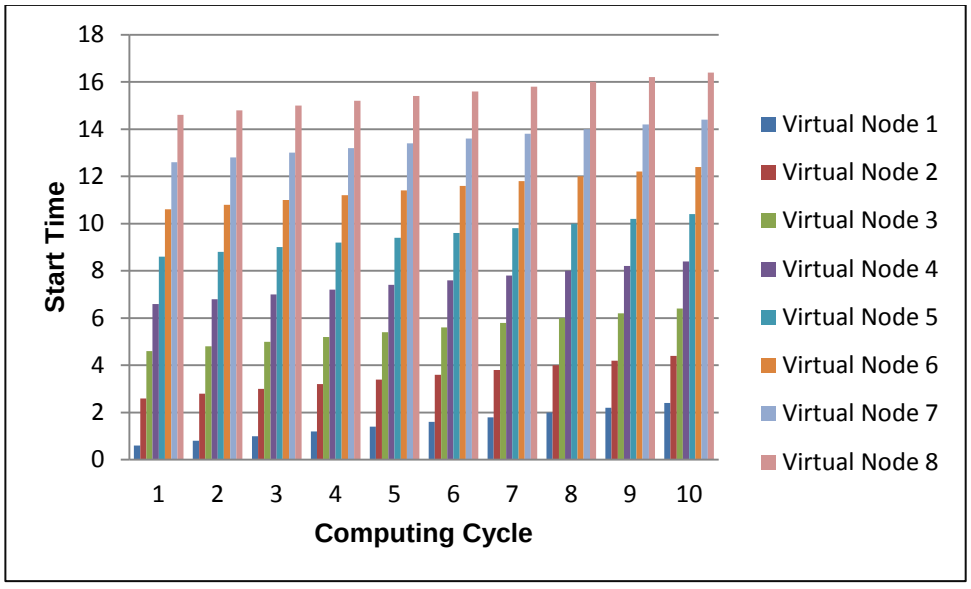

Figure 6.1: Start Time of the Virtual Nodes w.r.t. Computing Cycle

The figure 6.1 shows the start time of the different virtual nodes with respect to number of computing cycle. The start time is the time required to start a process or event.

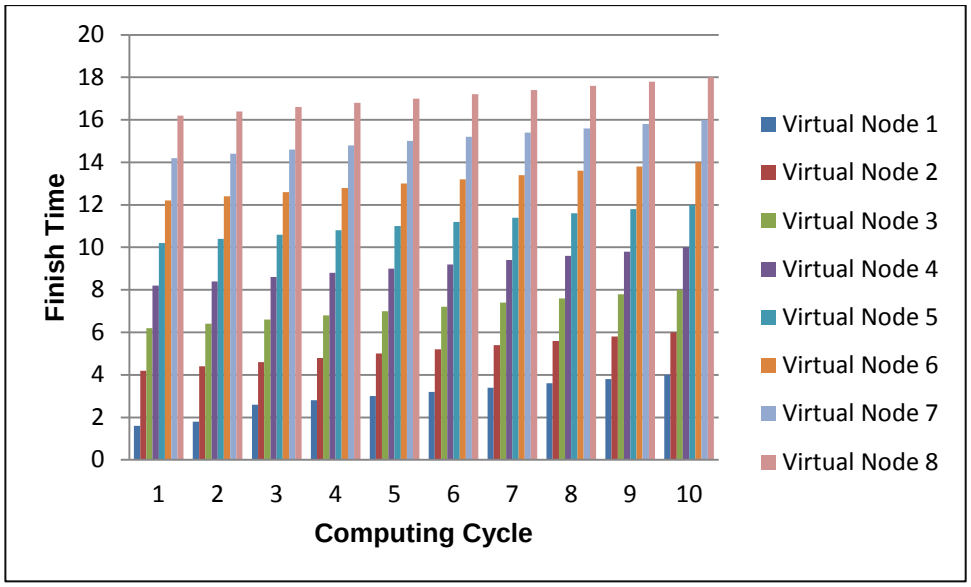

Figure 6.2: Finish Time of the 8 Virtual Nodes

The above graph shows the finish time of the 8 virtual nodes; where finish time is defined as the time required to complete a process or event in a given span of time.

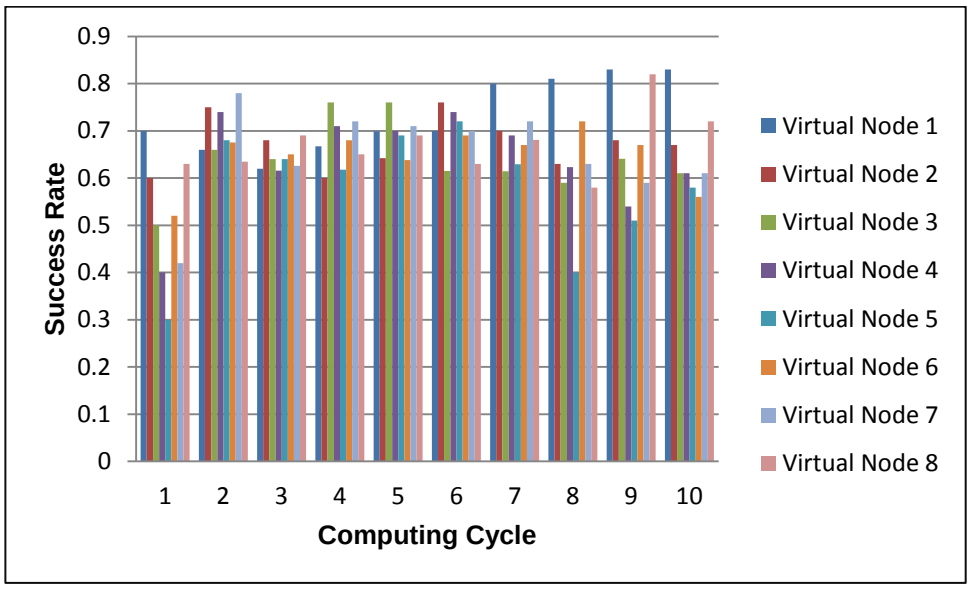

Figure 6.3: Success Rate of 8 Virtual Nodes

The success rate for 8 individual virtual nodes has been shown at given instant of time as appeared in the figure 6.3, where the term success rate is the percentage or fraction of success among a number of attempts.

From the above results it has been concluded that the proposed approach is relevant for balancing the load and efficient also for handling the faults. In the mentioned above, after getting the request from the client, the load can be distributed among the different virtual nodes using success rate computation algorithm in two different ways, one is assigning the job to the virtual node having higher SR value and another one is assigning the job to the virtual node with lower SR value. Accordingly, depending upon the SR value and the available load history, the virtual nodes has been selected.

CHAPTER 7

CONCLUSION

Cloud computing is found to be one of the colossal achievement by the IT developers crosswise globe. It enables the users to access extensible, distributed, virtualized, hardware or software infrastructure across the network. Load balancing is one of the leading issue of cloud computing. So there is need to set up a well ordered and economical fault tolerable load balancing approach for cloud computing.

The technique used for the dissertation work disburses the load by considering the success rate and the load history of the available virtual nodes. There are 10 computation cycles and for each cycle there are three virtual nodes. The main highlight of this work is to implement an efficient load balancing and fault handling technique to make the model more successful.

From the simulation of the results, it has been cleared that after calculating the success rate of the different virtual nodes, some nodes give the better results in case of existing work and some virtual nodes shows good results in case of proposed approach as compared to the existing one.

The future work is to provide an organized or efficient load balancing and fault tolerant technique in order to remove these issues permanently and to design the approach more cost-effective and successful.

REFERENCES

Ames, J. (2012). Dec., 18. Homepage, <http://blog.appcore.com/blog/bid/167543/Types-of-Cloud-Computing-Private-Public-and-Hybrid-Clouds>

Accessed 2014 July 4.

Bala, A. and I. Chana (2012). Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing. Computer Science and Engineering Department, Thapar University Patiala, Punjab, India.

Buyya, R., Broberg, J., and Goscinski, A. (2011). Cloud Computing: Principles and Paradigms. pp. 3-4. John Wiley & Sons, US.

Calheiros, R. N., Ranjan, R. and Beloglazov, A., Cesar, A.F. DeRose and Buyya, R. (2010). CloudSim: A Toolkit for Modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Wiley Online Library.

Chaczko, Z., Mahadevan, V., Aslanzadeh, S. and Medermid C. (2011). Availability and Load Balancing in Cloud Computing. Proceedings of International Conference on Computer and Software Modeling, pp. 134-140, Singapore.

Chandrakala, N. and Sivaprakasam, P.(2013). Analysis of Fault Tolerance Approaches in Dynamic Cloud Computing. International Journal of Advanced Research in Computer Science and Software Engineering, 3(2), 87-92.

Das, P. and P. M. Khilar (2013). LBVFT: A Load Balancing Technique for Virtualization and Fault Tolerance in Cloud Computing. International Journal of computer Applications, 69(28).

Desai, T. and Prajapati, J. (2013). A Survey of Various Load Balancing Techniques and Challenges In Cloud Computing. International Journal of Science and Technology, 2(11), 158-161.

Ganga, K. and S. Karthik, (2013). A Fault Tolerance Approach in Scientific Workflow Systems based on Cloud Computing. Pattern Recognition, Informatics and Mobile Engineering (PRIME), IEEE, 21-22.

Hung, C., Wang, H. and Hu, Y. (2012). Efficient Load Balancing Algorithm for Cloud Computing Network. Institute of Science and Technology, 251-253.

Jadeja, Y. and K. Modi (2012). Cloud Computing-Concepts, Architecture and Challenges. Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on, IEEE.

Jamsa, K. (2013). Cloud Computing. pp. 1-12. Jones & Bartlett Learning, USA.

Jhawar, R., Piuri, V. and Santambrogio, M. (2012). Fault Tolerance Management in Cloud Computing: A System-Level Perspective. IEEE, 1-10.

Kansal, N. J. and Chana, I. (2012). Existing Load Balancing Techniques in Cloud Computing: A Systematic Review. Journal of Information Systems and Communication, 3(1), 87-91.

Khiyaita, A., Zbakh, M., Bakkali, H.E. and Kettani, D.E. (2012). Load Balancing Cloud Computing: State of Art, IEEE, 106-109.

Kumar, D.K., Rao, G.V. and Rao, G.S. (2012). Cloud Computing: An Analysis of Its Challenges and Security Issues. International Journal of Computer Science and Network, 1(5).

Kumar, Y. R. (2013). Effective Distributive Dynamic Load Balancing For the Clouds. International Journal of Engineering, 2(2).

Magoulies, F., Pan, F. and Teng, F. (2012). Cloud Computing: Data Intensive Computing and Scheduling. pp. 1-6. CRC Press, UK.

Moharana, S.S., Ramesh, R.D. and Powar D. (2013). Analysis of Load Balancers in Cloud Computing. International Journal of Computer Science and Engineering, 2(2).

Natrajan, C. (2013). May 7. Homepage,
<http://chandrus.wordpress.com/2013/05/07/a-brief-history-of-cloud-computing/#> Accessed 2014 June, 12.

Nuaimi, K. A., Mohamed, N., Nuaimi M. A. and Jarrodi, J. A. (2012). A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms. Network Cloud Computing and Applications, 137-142.

Pathak, K. K., Yadav, P.S., Tiwari, R. and Gupta, T.K. (2012). A Modified for Load Balancing in Cloud Computing Using Extended Honey Bee Algorithm. International Journal of Research Review in Engineering Science and Technology, 1(3), 12-19.

Patra, P.K., Singh, H. and Singh, G. (2013). Fault Tolerance Techniques and Comparative Implementation in Cloud Computing. International Journal of Computer Applications, 64(14).

Randles, M., Lamb, D. and Taleb-Bendiab, A. (2010). A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing. International Conference on Advanced Networking and Applications Workshops, 551-556.

Ray, S. and Sarkar, A.D. (2012). Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment. International Journal on Cloud Computing Services and Architecture, 2(5).

Roy, A. and Dutta, D. (2013). Dynamic Load Balancing: Improve Efficiency in Cloud Computing. International Journal of Emerging Research in Management and Technology, 2(4).

Sanyal, M. K., Das, S. and Bhadra, S. (2013). Cloud Computing- A New Way to Roll Out E-Governance Projects in India. International Journal of Computer Engineering and Technology, 4(2), 61-72.

Shaikh, F.B. and Haider, S. (2011). Security Threats in Cloud Computing. Internet Technology and Secured Transactions, 214-219.

Sran, N. and Kaur, N. (2013). Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing. International Conference on Computer and Software Modeling, 2(1).

Sharma, T. and Banga, V.K. (2013). Efficient and Enhanced Algorithm in Cloud Computing. International Journal of Soft Computing and Engineering, 3(1), 385-390.

Tchana, A., Broto, L. and Hagimont, D. (2012). Fault Tolerant Approaches to Cloud Computing Infrastructures. The Eight International Conference on Automatic and Autonomous Systems, 42-48.

Velte, A.T., Veltey, T.J. and Elsenpeter, R. (2010). Cloud Computing: A Practical Approach. Tata McGraw-Hill Education Private Limited, New Delhi, India.

Verma, B. (2012). April 18. Homepage,
<http://www.techinmind.com/what-is-cloud-computing-what-are-its-advantages-and-disadvantages/>
Accessed 2014 June, 24.

Wang, S.C., Yan, K.Q., Liao, W.P. and Wang, S.S. (2010). Towards a Load Balancing in a Three-level Cloud Computing Network. IEEE, 108-113.

Wickremasinghe, B., Calheiros, R.N. and Buyya, R. (2010). CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications. Advanced Information Networking and Applications (AINA).

Zenon, C., Mahadevan, V., Aslanzadeh, S. and Mcdermid, C. (2011). Availability and Load Balancing in Cloud Computing. International Conference on Computer and Software Modeling, 14.