CHAPTER 1

INTRODUCTION

Cloud computing is a most widespread and popular form of computing, promising high reliability for customers and providers both at the same point of time from many fields of sciences or industry. Clients from the different field are served by datacenters in cloud environment geographically spread over the world. Cloud serves a large number of requests coming from various sources over datacenter with high power consumption. However, to provide such a large computing power required a huge power, leading to high power consumption and cost. Request types in cloud system also affect the services which are public and private requests whose proportion is random in nature. A survey in 2006 over the performance of cloud environment in the USA shows datacenter consumed 4.5 billion kWh units of power, which is 1.5% of total power consumed in the USA and this power requirement is increasing 18% every year [1]. In general, cloud computing deals with various issues live poor resource utilization and load balancing and many more. Some of the issues are discoursed as follows: 1) as cloud computing tools are used by industry and they have issues with the rapidly growing request and a number of servers deployed, increasing the power consumption. 2) Task allocation of request among datacenter without having knowledge of QoS provided by servers. 3) Current task allocation algorithms only focus on balancing the request and improve unitization of the system but not the failure probability of system. 4) High loaded data centers have high failure probability and due to high load, this may lead to slow down of datacenter and poor QoS (Quality of service) to the client and client provider. 5) While few of the servers are overloaded and some of them are idle or under loaded. 6) Some request needs to be computed with QoS but due to high load and fault rate they may the QoS promised which is not appropriate to the user and will be a critical
issue. 7) As per recent study [2-5], utilization of data centers is a major problem because 60% data centers are idle and most of 20% data centers are utilized and waste of the resources.

This shows the poor utilization of resources but this shows the importance of a new approach that has sufficient strategy to minimize wastes of resources and increasing reliability by allocating task over resources which in the case of Cloud is VM with low failure probability to provide high QoS to users. The existing algorithms only take into consideration cloud as non-faulty in nature and fail to provide specific QoS when a fault occurs. So to overcome these issues and improve the performance of the system, we have proposed approaches for resource load balancing and allocation. Figure 1.1 shows cloud computing features, type and various other properties [6]

Figure 1.1 Cloud system characteristics and properties

1.1. Characteristics of Cloud

Cloud is a distributed environment, where the servers are placed at various geographical locations but seems to a user as a single entity. Cloud computing provides better performance than any other distributed system line Grid computing or cluster computing and
many more. There are various characteristics of cloud computing which makes it superior than any other system which are as follows [6, 17];

a) High Availability
   One of the most important features of the cloud is all-time availability of resources in form of storage, computational capability and high network resources. This property also states that the resources are available in overloading conditions also.

b) Pay per use model
   This feature made cloud computing popular in the industry due to the affordable nature of cloud by an industry with high infrastructure or a business holder with the small requirement can easily manage and have its own infrastructure and high computing system at a low cost. Cloud computing allows a user to pay for only those resources, which are used by him for that specific period of time rather than purchasing a complete server or private infrastructure.

c) Elasticity
   Cloud is said to be flexible and scalable at the same time. This feature allows the cloud to scale its resources up or down based on the user or business needs for a period of time. This allows the cloud to have high availability under overloaded condition also and provided uninterrupted services to the user without failure and high quality of service.

d) Reliability
   Cloud computing ensures to provide high reliable computing services and resources to the user which means that the user will be provided with uninterrupted services with the quality of services as assured to the client.

1.2. Business Models
   Cloud computing provides various service-driven business models to provide a different level of computation to the users. Cloud computing provides 3 type of service models listed as:
   Software as a Service (SaaS), platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Our work focuses on improving the performance of cloud infrastructure as a service in a faulty cloud environment [17]. The fault is a behavior of every distributed system because fault may occur any time that may be due to system failure, network failure or disk failure.
1.3. Issues in Cloud Computing

Cloud computing deals with various issues to maintain above discoures characteristics and quality of serves assured to the user by cloud providers in term of high resource availability, computational capability [7-8]. Some or the issues dealing with resource management, resource scheduling, and managing system performance are discoures below.

- Resource allocation
- Load balancing
- Migration
- Power efficient resource allocation and load balancing algorithms
- Cost efficient resource allocation and load balancing algorithms
- Fault tolerant algorithms
- Behavior-based algorithms
- Trust management

1.3.1. Resource Allocation

Resource Allocation strategy (RAS) in the cloud is all about the scheduling of tasks or requests by cloud provider in such a manner to balance the load over all the servers and provide high Quality of Service to clients. It also includes the time required to allocate the resources and the resources available. The main aim is to improve the utilization of resources and complete all the request within the deadline and with least execution time [9].

An optimal RAS should avoid the following criteria as follows:

a) Resource contention situation arises when two applications try to access the same resource at the same time.

b) The scarcity of resources arises when there are limited resources.

c) Resource fragmentation situation arises when the resources are isolated.

d) Over-provisioning of resources arises when the application gets surplus resources than the demanded one.

e) Under-provisioning of resources occurs when the application is assigned with fewer numbers of resources than the demand.

Resource allocation algorithm can be categorized into three subcategories as from the literature review conducted over existing proposed algorithms.
Categorization is as follows:
1) Static
2) Dynamic
3) Learning-based.

Static scheduling algorithms are referred to algorithms which are not affected by system and behavior of cloud some of the algorithms line SJF, FCFS, Round robin etc [11]. On the other hand, dynamic algorithms are those whose objective function depends on the system parameters line deadline, available resources, resource utilization of host and many more example of these algorithms is a deadline-based algorithm, cost-based algorithm, utilization based algorithm [5-10]. The problem with these algorithms is that they do not take into consideration the previous performance of host and system as a whole. Moreover, the past faulty nature of the system is not taken into consideration and leads to large request failure. Dynamic algorithms deal with the issue of local minima these algorithms are not able to find a global best solution and stuck in local best solution.

1.3.2. Load Balancing

Load balancing aims to distribute load across multiple resources, such as server, a server cluster, central processing. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource

Goal of Load Balancing [12] are as follows:
1) To improve the performance substantially.
2) To improve system stability.
3) To have scalability in the system.
4) To improve the system condition under high load or request rate.

Types of Load balancing algorithms [13]

*Sender-Initiated:* When load balancing algorithm is triggered by the sender.
*Receiver Initiated:* When load balancing algorithm is triggered by the receiver.
*Symmetric:* It is the combination of both sender initiated and receiver initiated.
Load balancing is also used to manage the average utilization of the system as a whole to avoid creation of hot spots i.e. the request should not be clustered on a single datacenter rather should be spread over the servers. So it aims it find an underloaded server and move the requests to that selected server. This makes a requirement of a load balancing algorithm to fulfill these requirements taking into consideration system utilization and quality of service without failure.

1.3.3. Migration

Migration in cloud infrastructure plays an important role in cloud Infrastructure under system overloading condition. In cloud infrastructure when the server gets overloaded i.e., the utilization is beyond a threshold is considered to be overloaded, in such condition we need to migrate a virtual machine from overloaded server to an under loaded or neutral server [14]. This help to balance the load and prevent the server from any failure. So there is a requirement of an intelligent and efficient migration algorithm or balance the condition and improve the performance of the system.

1.3.4. Power efficient resource allocation and load balancing algorithms

The power efficiency of a cloud environment is an important issue for a green cloud environment. As 53% of the total expense of a datacenter is spend on cooling i.e. power consumption [15]. In a survey in 2006 on datacenters established U.S consumed more than 1.4% of total power generated during the year [16]. Therefore we require improving the power efficiency of infrastructure. The problem can be solved in various ways and various proposal are been made to solve and improve the performance. So to do this we need to design power-aware resource allocation and load balancing algorithm to improve the total power consumption of the system and any such algorithm will result in a reduction of overall power consumption.

1.3.5. Cost efficient resource allocation and load balancing algorithms

Cloud computing uses pay-per-use model to ensure least cost and payment only for the resources used. To maintain this feature cloud controller algorithms like resource allocation migration and load balancing are responsible for maintaining this characteristic by offering the resources which can complete the client request on time and within the budget of client and have least cost that can be offered. So we require cost aware algorithm which are cost efficient and can provide the best system performance by improving utilization and
power consumption all at the same time [10, 15]. These type of algorithm are referred to as multi-objective algorithms, there are many proposals made for improving the performance of the system but they only take into consideration either power or cost, so cannot guarantee the best performance.

1.3.6. **Fault tolerant algorithms**

Cloud computing environment is a type of distributed environment like grid computing and cluster computing. Existing algorithms consider cloud as nonfaulty but faults are a part of distributed environment which may be due to hardware or software failure at any point of time [93, 99, 100]. There are many fault aware and fault prediction algorithms been proposed for grid environment to improve the reliability of the system. So similarly we require fault aware algorithms to make system fault aware reduce the failure probability of the system and increase the reliability of the system.

1.3.7. **Behavior-based algorithms**

Most of the resource allocation and load balancing algorithm proposed for cloud infrastructure are dynamic algorithms like min-min, max min and many more. These algorithms take into consideration only the current behavior \ status or the server and system for selection of server. The problem with these algorithms is that they do not take into consideration the previous performance of the system for prediction of the better solution rather than stuck in local minima. Behavior-based algorithm lists genetic algorithm, ant colony, particle swan optimization, monkey search and many more. So there is a need of algorithms taking into consideration the previous and present performance of the system for decision making.

1.3.8. **Trust management**

Trust models are been used in all form of distributed environments ranging from MANETS (Mobile ad hoc network), Sensor network and Grid computing to validate the reliability of nodes over distributed network. In grid computing, various trust models are been proposed to ensure trust in term of security and reliability of the server or the node. Trust models are to resolve the problem of reliability in any heterogeneous environment, which contributed of nodes having different configuration spread over a network. There are many models being proposed in a cloud computing environment.
What is Trust?
Trust can be defined as an entity based on reliability and firm belief based on an attribute of the entity. Trust is the firm belief in the competence of an entity to act as expected, such that this firm belief is not a fixed value associated with the entity, but rather it is subjected to the entity’s behavior and applies only within a specific context at given time [18]. The definition simply means that trust is a variable changing believe, based on both static and dynamic parameters.

Trust can also be defined as “the subjective probability by which an individual expects that another individual performs a given action on which its welfare depends” [19-20].
Trust can be categorized into three major classifications which are as follows [21]:

a) **Blind trust:** This is the default trust before any event in the system, and which would include an agent to initiate a relationship with unknown entities.

b) **Conditional trust:** This is a classic state of trust during the life of the agent. This condition trust is likely to evolve, and can be subject to some sets of constraints or condition.

c) **Unconditional trust:** Such a trust is the probability be configured directly by an administrator, and would not be sensitive to successful/unsuccessful interaction and external recommendation of any other sources of evolution of the conditional trust.

1.4. Problem statement
The aim of this work is to make system fault tolerant and more reliable computing system with improved performance in cloud infrastructure environment. A number of algorithms have been worked out for long period of time but they assumed cloud as non-faulty. So in our work, we have proposed various fault tolerant algorithms to resolve various issues as follows:
1) To design a fault and deadline aware load balancing algorithms for private and hybrid cloud, which aim to improve QoS of load balancing algorithm and minimize the faults, resource utilization, minimize response time and avoiding overloading of any single resource in cloud.

2) To design learning based fault aware resource allocation algorithms, to provide a global best schedule with least scheduling time complexity.

3) Designing fault aware and power-efficient scheduling algorithms for improving power efficiency and request failure count in the cloud.

### 1.5. Parameters Used

**Fault rate:** defined as the total count of request failed over a period of time T

**Failure Probability:** as the probability of request to fail on a specific host or system.

**Reliability of a system:** This feature of a system can derived from the failure probability of system which can be defined as:

\[
\text{Reliability} \% = 100 - \text{Failure Probability}\%
\]

\[(1.1)\]

**Power Efficiency:** The ratio of the output power over the input power i.e. the percentage power consumed over a period of time.

**Utilization:** this is the capability of the host to be used out of total available resources.

**Average Resource Utilization:** This is an average of utilization of resources over the whole system i.e. all available hosts.

**Average start time:** Average waiting time of request before been scheduled or allocated.

**Average Finish time:** Average of finishing time of all the request executed by system.

**Scheduling Delay:** Total time to find a suitable resource for a set of tasks.

**MakeSpan:** Total execution time of system including scheduling delay for set of requests /Task

### 1.6. Performance Parameters

To study the performance of proposed algorithm over existing algorithm we require to compare these parameters listed below:
Average utilization: Average utilization is the average percentage of time during which the server is busy processing jobs during a simulation.

Power Utilization: Power utilization can be defined as the power consumed in kWh during the simulation.

Average queue length: This is the average size of the queue of a server during a simulation.

Request failure count: Total count of requests failed during a simulation.

Request completion count: Total count of requests completed during a simulation.

Average start time: This is the average of the start time of all the requests generated during the simulation.

Average finish time: This is the average of finish time of all the requests generated during the simulation.

Scheduling delay: The times taken to find a suitable server for a set of given requests.

Makespan: The time taken to complete all requests over a given cloud environment.

Failure probability: The probability of failure of each request is a given system.

Reliability: Reliability of a system can be defined as the probability of the system being reliable, which can be defined as (1 - failure probability).

1.7. Outline of the Thesis

The thesis has been organized into 6 chapters and out of that CHAPTER 1 presents an introduction comprises problem statement and various issues in cloud computing. CHAPTER 2 presents the existing proposed work to solve the problem of resource allocation and load balancing with different combinations of authors taking different performance matrices. This chapter also discussed the various methods power efficient, cost efficient algorithms and algorithms inspired by nature. The preliminary notations are introduced to keep the clarity of usage throughout the thesis. CHAPTER 3 presents three techniques for load balancing for cloud storage in faulty environment to improve the fault tolerant behavior and reliability of cloud. The approaches are proposed for private and hybrid cloud environment. CHAPTER 4 presents a set of learning based techniques for the faulty cloud to find global best solution and shows improvement in scheduling delay, failure count, failure probability and the reliability of the system. CHAPTER 5 presents a set of power efficient and fault aware approaches inspired from nature like honey bee and ant colony algorithms to find best suitable resource. Finally, followed by the conclusion and future scope of the research work for further research are provided in CHAPTER 6.