CHAPTER 2

LITERATURE SURVEY

This chapter briefly introduces the cloud computing environment and the technologies used to facilitate various cloud services viz. Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Various approaches, technologies, methodologies and strategies adopted by major Cloud Service Providers (CSPs) are also summarized. The main focus of this thesis work is related to resource allocation techniques using virtualization for Cloud Computing environment. This work is a contribution to Infrastructure as a Service. A literature review is presented here along with the background and reasons for choosing these issues and the solution approach adopted. Finally, specific identified resource allocation issues are listed that need to be addressed followed by the objectives for these issues are presented.

2.1 INTRODUCTION

Today the world of computation is moving towards pay-as-per use model due to the numerous benefits provided by this model. Hence cloud computing services are predicted as the best option for future computational world [1]. Cloud computing is compared with other previous synonymous existing services and underlying technologies viz. utility computing, services provided via the internet using the web browsers, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), Grid Computing, and data centers etc [2]. The reasons for the advances in cloud computing mainly include the recent advances in Internet backbone, high performance, and scalable infrastructure in the web technologies and data centers. As a consequence of the inherent advantages and the enabling technological support systems, cloud computing is sure to become the key computing paradigm within the next few years. It may redefine and revolutionize the IT world with its efficient, easy to use, and economical new model [3].

Wide range of literature and methods [2][3][4][5] have been developed and designed for emerging cloud environments including the server connections, client connections and the
session management between the clients and the servers. Various systems and techniques have been developed and abstracted in different layers to practically facilitate the cloud services for using the resources placed onto the cloud. These abstract layers are patented by different organizations [147]. Various techniques are developed to predict the future utilization of cloud resources that may be requested by the users. Based on such predictions, more accurate allocations can be implemented to reduce the processing time [7][8]. As the cloud resources are requested and utilized on pay-as-per-use basis, so, some well-defined billing technologies are developed to measure the usage in an accurate and precise way [149]. Managing the resources on the cloud is a major concern. This is because the servers are deployed in a distributed fashion at various locations. Therefore, the complexity of coordination to access the resources requested by users in an efficient way with minimum complexity is an inherent requirement. Various resource deployment strategies are developed and adopted for the cloud environment [150].

Cloud Computing is the fastest growing technology due to continuous growth of internet users. It has three major features as follows:

- **Pay as per use:** Usually, business associations have to setup their own IT infrastructure to meet their business requirements of storage, network, services etc. which requires huge capital investment. Through cloud, they can get these infrastructures as a service, which saves huge upfront investment and have to pay for only what they use.

- **Flexible Allocation:** The resources are allocated only temporarily to a user and not permanently. The resources are allocated to user on demand from shared pool and returned back to pool after completion of its use. Then they can be re-allocated.

- **Virtualization:** It provides resources in a scalable manner which makes it cost effective and hence becomes very critical to cloud computing. It has three characteristics which makes it ideal for cloud computing- partitioning, isolation and encapsulation.

Cloud based services can be divided into three types: (i) Infrastructure as a Service (IaaS) (ii) Platform as a Service (PaaS) and (iii) Software as a Service (SaaS).
Most Service providers do not take into consideration the Multidimensional resource usage like, CPU usage, Memory Consumption and network bandwidth used, while placing the Virtual Machines (VMs) onto the physical machines. The major reason for this is that a lot of pre-computations are required to find the resource usage in varying and heterogeneous nature of workloads [10][12].

In Cloud Computing model, computing and data moves in between the portable PCs and the data centers. Most of the applications are delivered as a service over the Internet on to the cloud infrastructures, viz. the computing hardware, system software and other application software, which are the services provided by cloud computing model. The backbone of cloud computing infrastructure and services are result of advances in virtualization technologies, development of service oriented software, advancements in grid technologies, managed services and efficient power utilization innovations [13]. In cloud model, all arenas of knowledge and applications are required. viz., financial issues, legal issues, administrative issues, last but not the least the technical issues. These fields of knowledge and implementations are an integral part of cloud architecture. All these arenas are required to support the operations of cloud model for leveraging the services. The architectural models suggested for the clouds can be divided into two parts according to the services provided.
• To provide computing power on demand as per requirement and uses the instances to provide services as PaaS and SaaS.

• To provide data and scalable intensive computing resources on-demand.

These services are leveraged by pay-per-use model which is compatible to clusters of loosely coupled processors. Advances in virtualization technologies have given birth to highly growing commercial and freeware infrastructure based technology known as cloud computing. The key challenge for developing such applications lies in handling the heterogeneity in terms of tasks, resources, applications and infrastructures which inherently belong to grid and cloud resources. Cloud computing model is based on customized environment created on top of the physical infrastructure using Virtual Machine technologies. VMM (Virtual Machine Manager) plays an important role as a link between the grid resources and the gateway using virtualization technology [10][11].

A taxonomical classification is presented here for better understanding the cloud model and its functionalities. The design of cloud architecture is to support the services to the end users irrespective of their geographical locations. It appears to the end users as if there is single point of access to all services. Cloud services are categorized as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) as defined by National Institute of Standards and Technology (NIST). These categories of services are facilitated by many small and large service providers. For example, Software as a Service (SaaS) that inherently supports multi-tenancy property is provided by Salesforce.com, Microsoft, Oracle, IBM etc. A cloud based Platform as a Service (PaaS) is provided for software developers. The developers can choose any platform of their choice of work for developing the complete software life cycle programs, i.e., Software Development Life Cycle (SDLC). Their programs can be developed, tested, deployed and hosted from the cloud. Some of the major PaaS providers are Microsoft's Azure, Google App Engine, Amazon's Web Services etc. The cost of execution, operational complexities, administrative issues and development time is highly reduced if these services are used from the Clouds, as compared to the traditional ways of executions. Another service type provided from the cloud is Infrastructure as a Service (IaaS). These services facilitate to provide the computer infrastructures in a flexible and scalable way on pay-as-per-use basis. Some of the major infrastructure service providers are RackSpace,
FlexiScale, Joyent, GoGrid, AppNexus etc. Three service models are given by NIST, namely Private, Public and Hybrid. Private Cloud is set up in the premises of the organization. It offers services from within the organization. These cloud services are costlier as compared to the other types of services, but data and processes are more secure, as they are managed within the organization. Hence this model supports prevention from security exposures and reduces the complexity of legal hassles in Service Level Agreements as compared to Public and Hybrid Clouds. The examples of private clouds are Eucalyptus, VPC from Amazon, VMWare, Enomaly, Redplaid and Intalio etc [12].

Public Cloud provides open access to all public and other organizations. This is highly used and will be in demand for future cloud models, where provisioning is done dynamically in a fine-grained, self-service manner using web technologies and services over the internet. Examples of Public Cloud are FlexiScale, GigaSpaces, Zimory, Amazon’s Elastic Cloud Compute (EC2), Windows Azure, SymmetriQ and SunCloud etc. Hybrid Cloud uses both Private and Public Cloud to optimally provide services for different applications and their requirements. Private Cloud services are used for secure data and processes, whereas the Public Cloud is used where less or minimal security is required. For example, Elastra, Cloud Backup from Aisgra, Carpathia and Right Scales etc. [15][20].

2.2 INFRASTRUCTURE AND TECHNOLOGIES ADOPTED BY MAJOR CSPs

Various computing solutions, infrastructures and techniques are adopted by major CSPs, namely AWS, GoGrid, FlexiScale and RackSpace. Some technical details for these CSPs are mentioned below [12]:

2.2.1 Computing Architectures: The computing architecture of AWS known as Elastic Cloud Compute (EC2) uses Xen VMM to upload the VM images and their instances. GoGrid uses dedicated computer system resources that are based on grid architecture. FlexiScale uses scalable auto-reconfigurable infrastructure at data centers to cater the varying and fluctuating demands. RackSpace uses the idea of merging the servers on to the cloud environment to share resources with end users [43][44].
2.2.2 Virtualization Management: AWS, GoGrid and FlexiScale uses Xen hypervisor for virtualization management whereas, RackSpace uses the VMWare ESX Server for the same [45][46][47][48].

2.2.3 Services: AWS, GoGrid, FlexiScale and RackSpace, they all provide Infrastructure based services. Additionally AWS provides Xen Images as IaaS.

2.2.4 Load Balancing: AWS, GoGrid and FlexiScale uses various algorithms like, Round-Robin, request-balancing, migration of virtual servers in between the physical servers by scaling the physical servers horizontally and vertically to balance the loads. RackSpace uses simple software load balancing technique to balance the loads horizontally and vertically both [49][50][51].

2.2.5 Fault Tolerance: For handling fault tolerance, AWS system environments are designed in such a way that it automatically generates alert, failure messages and may re-synchronize the previous known states. Fault tolerance in GoGrid is implemented by instantly backing up the files in a reliable and scalable way. FlexiScale uses virtual dedicated servers and automates the operational services like start, stop and delete which can change the memory locations CPUs and may switch to other operational servers. Rackspace shares the IP between two servers in a Master and Slave mode to run the applications even in presence of faults [52][53].

2.2.6 Interoperability: AWS adopts interoperability in a horizontal way, i.e., it allows interoperable instances between EC2, Eucalyptus and other similar infrastructures. GoGrid is interoperable with various other Cloud infrastructure service providers like Gigaspaces. FlexiScale is interoperable with Amazon’s EC2, Mosso and GoGrid. e.g., applications deployed on FlexiScale can be managed and run onto above mentioned Cloud infrastructure too. Rackspace provides open specifications and manifesto for Cloud APIs, Cloud Server Files/API etc. that can be operated and run onto other Cloud Infrastructures.

2.2.7 Storage: AWS uses Simple Storage Server (S3) and Simple Data Base (SimpleDB). GoGrid uses private network services for connecting each of their servers by using the transfer protocols like SAMBA, FTP, RSYNC, SCP etc. to move data to and from cloud storages. FlexiScale uses SAN as a high-end devices and NAS as back-end devices. Their maximum
storage capacity is over 160 TB that is spread over more than 325 drives. Rackspace uses Cloud Files for storing data through limelight network [54] [55] [56].

2.2.8 Security: AWS protects their APIs by incorporating Secure Socket Layers. It also uses firewall certification using X.509 and using Secret Access Key (SAS70). GoGrid secures data by hosting private cloud service known as PrimeCloud. It prohibits customers from sharing such resources. FlexiScale adopts strategy of using dedicated Virtual Private Servers for data privacy. Rackspace uses encryption strategy in the communication channel itself - Encrypted Access Key for APIs and token generation strategy for authenticating sessions.

2.2.9 Programming Framework: AWS uses the MapReduce framework known as Amazon Elastic Compute. It supports multiple high level-languages like Java, PHP, Ruby, CGI, Perl etc. GoGrid supports Java, Python, Ruby and PHP. FlexiScale supports C, C++, C#, Java, PHP, Ruby and Perl. Rackspace supports Java, DOT.NET, Ruby, and Python.

2.3 CLOUD COMPUTING: PaaS AND SaaS SERVICE PROVIDERS

Some of the major PaaS and SaaS providers are mentioned below. They adopt different techniques to implement their features. Some important features are computing architecture, virtualization management, services provided, load balancing techniques adopted, implementation of fault tolerance techniques, storage management, interoperability, handling security issues and programming framework used. Major PasS and SaaS service providers are Google Application Engine (GAE), Gigaspaces, Azure, RightScale, SunCloud and SalesForce.com. Features adopted by these CSPs and their implementation details are given below:

2.3.1 Computing Architectures: Google Application Engine (GAE) uses geographically distributed architecture, GigaSpaces uses Space based architecture, Microsoft Azure’s data centres supports scalable platforms for cloud and developer services by providing individual OS environments for each set of services. RightScale provides clusters of virtual private servers for system monitoring. SunCloud uses Solaris OS and dynamic strategy for infrastructure management using Zetta-byte File System (ZFS). SalesForce uses Multitenant metadata driven architecture [43][44].

25
2.3.2 Virtualization Management: GAE has multitenant architecture, Gigaspace provides GigaSpace based service virtualization, Azure uses Hyper-V hypervisor, RightScale uses Xen hypervisor. SunCloud uses Sun VM hypervisor and uses ZFS and COMSTAR based storages. SunCloud supports Java and Glassfish based applications. SalesForce uses multitenant architecture that supports better isolation between private and shared data and logics [45][46][47][48].

2.3.3 Services: GAE, GigaSpaces, Azure, RighScale and SunCloud provide PaaS whereas SalesForce.com provides SaaS.

2.3.4 Load Balancing: GAE uses auto load balancing and scaling technique. Load balancing in GigaSpaces is done by high-performance communication protocols on EC2. Azure has in-built load balancing support in the hardware itself. Load balancing in RightScale is implemented by high availability proxies in the cloud. SunCloud uses Horizontal and vertical scalability for load balancing. Salesfore.com directly balances the load among their tenants [49][50][51].

2.3.5 Fault Tolerance: GAE automatically pushes the requests to fault tolerant servers. It has Cron Service in its Application Engine. GigaSpaces uses Service Virtualization Framework (SVF) for handling failures. Azure supports automated shifting to the replicas of SQL data servers in case of failures. RightScale uses all types of architectures including the advanced ones for handling failures by using Elastic IPs. Failure handling in SunCloud is achieved by scheduling the requests based on the resource requirements. SalesForce.com achieves the same by automated self tuning and management in case of failures [52][53].

2.3.6 Interoperability: GAE supports interoperability between the platforms of different service providers. It also supports multiple programming languages. Windows Azure uses interoperable platform for building new applications and also for enhancement of existing applications. RightScale uses integrated dashboard for managing the applications across clouds by deploying it only once. SunCloud uses open source java and interoperability for the large-scale computational resources across the clouds.
2.4 OPEN SOURCE CLOUD COMPUTING PLATFORMS

In addition to the above mentioned commercial cloud service providers, there are open source cloud service platforms available and are highly used by different organizations, including academia and Research and Development organizations. The advantages of using these platforms are that it can be customized as per their application requirements. Some important features of different open source platforms, namely Eucalyptus, OpenNebula, Nimbus, Enomaly Elastic Computing are mentioned below:

2.4.1 Computing Architecture: Eucalyptus supports multiple clusters that can be configured with individual network address and all clusters can be run onto a single cloud platform. OpenNebula provides efficient and dynamic scalability feature for managing the VMs in data centers. It uses Haizea based scheduling technique. Nimbus architecture consists of TeraPort clusters that are Globus based and is used for interfacing with cloud clients. Several virtual machines are combined together using Context Broker to create virtual clusters. Enomaly platform supports hosting of virtual clusters, distributed and remote storage system. It uses Content Delivery Networks for private and high-speed data communication [57] [58].

2.4.2 Virtualization Management: Eucalyptus, OpenNebula, Nimbus AND Enomaly Elastic Compute, use Xen hypervisor for virtualization management. Additionally, OpenNebula uses Amazon’s EC2 on demand basis. It also uses KVM. Enomaly uses VirtualBox and OpenVZ that is supported by KVM and Xen [47][57] [58]

2.4.3 Load Balancing: Eucalyptus incorporates cloud controller for load balancing. OpenNebula uses round-robin and other weight-based algorithms for load-balancing. A dedicated server is configured as load balancer. Nimbus uses context broker which acts as a self-configuring agent for virtual clusters. Enomaly elastic computer uses random, round-robin, hash and least resource usage based algorithms for load balancing [60].

2.4.4 Fault Tolerance: Eucalyptus adopts the concept of creating separate clusters within the cloud. This reduces the probability of interdependency and correlated failures. OpenNebula incorporates a daemon process which can be started when there are VMM failures. This daemon can recover all failures in currently running VMs. Nimbus uses periodic check and
recovery algorithms on currently active nodes. Enomaly incorporates algorithms and mechanisms for disaster, overflow and failover services.

2.4.5 Interoperability: Eucalyptus supports multiple interfaces while using the same cloud computing infrastructure in the back-end. OpenNebula supports the services of many intra-clouds e.g. EC2, elastic hosts etc. using the plug-ins. Nimbus supports inter-operations between various CSPs that have standard working codes. Enomaly provides portability of clouds to different cloud vendors to support interoperability.

2.5 IaaS PLATFORM SELECTION

InterGrid system is an execution platform for running applications on interconnected infrastructure which resembles the cloud computing model. The underlying technology used is virtualization which has been in use from the mainframe era to the new cloud computing era. These execution environments can be used in cloud infrastructures like Amazon’s EC2 (Elastic Compute Cloud). This architecture is proposed along with its implementation and experimental details with the results showing the benefits of using this virtualized distributed computing infrastructure. Recent advances in virtualization technologies have given birth to highly growing commercial infrastructure based technology known as cloud computing. The key challenge for developing such applications lies in handling the heterogeneity in terms of tasks, resources, applications and infrastructure which inherently belongs to grid and cloud resources. InterGrid resource sharing architecture is developed, the communication of which is based on InterGrid gateways (IGGs) which act like middleware brokers and facilitate all grid to share resources belonging to other grids. VMM (Virtual Machine Manager) plays an important role as a link between grid resources and gateways using virtualization technology. VMMs allow to run several guest operating systems concurrently onto the same host, which give privileged access to the hardware. VM template consists of the description for the VM with certain information like number of cores or processors to be assigned, the amount of memory required, requirement of kernel to boot VM’s operating systems, disk images consisting of VM’s file systems and price for using VM, if required. These information are static in nature and are used later. VM templates can be added, deleted and updated voluntarily by the administrators [61] [62].
Several cloud IaaS frameworks that provide cloud computing and virtualization services to users like Eucalyptus, OpenNode, CloudStack, CloudSigma, Archipel and EMOTIVE (Elastic Management of Tasks in Virtualized Environments) are available. A major problem with these frameworks is that they do not take into account properties of datacenters, viz. fault rate and processing speed etc. Datacenters are categorized on the basis of their calculated trust values are based on parameters that are taken into account. The task of these models is to categorize the data centers on the basis of multiple parameters collaboratively rather than single parameter.

**Open Source VMM: XEN**

Xen virtualization technology is available for Linux kernel. Xen is designed to consolidate multiple operating systems to run on a single server, normalize hardware accessed by the operating systems, isolate misbehaving applications, and migrate running OS instances from one physical server to another. Xen is an open source virtualization software based on Paravirtualization technology. Xen virtualization software abstracts the hardware by VM creations to virtualize resources like CPUs, memory, networks etc., [63]. Multiplexing of physical resources between the VMs is enforced by a VM monitor, which is also designed to provide required translations of operations from VMs to the physical hardware. On Xen hypervisor, the VM migration feature can easily be run and its effects analysed with xmmigrate command. This command has been frequently used and emphasized as it can be modified with variety of attributes that goes with the command such as whether we want the migration to be live or cold. However, the real problem is of detection of when to fire the VM migration mechanism and which VM is the causing trouble. System Virtual Machines (VMs) [18][54][64] are widely used on personnel computers as well as organizations having LAN based setups. System virtualization acts as powerful means of abstraction for upcoming applications. Resources are provided according to need on pay as per use principle in Cloud Computing. Service Level Agreements (SLAs) specify subscriber requirements, methods of maintaining balance between cost, quality and resources. Cloud Service Providers guarantee a level and quality of service to users as per the terms and conditions of SLAs. Many challenges are faced by CSPs while catering to needs of users and at the same time making efficient use of underlying heterogeneous resources in a dynamic way, as inherently expected from Cloud
Services in terms of Infrastructures, platforms and software.

2.6 RESEARCH ISSUES AND DIRECTIONS IN CLOUD COMPUTING

Based on the previous study about different CSPs and the techniques adopted by them for leveraging the cloud services, some major technical research issues related to cloud environments are identified as under:

2.6.1 Scheduling Algorithms in Cloud Computing

Cloud services are provisioned in such a way that it can dynamically be serviced without manual interference. This feature hides complexity of cloud’s internal systems from application developers and end users.

Cloud Computing is a vast field spanning various aspects on which its efficiency is dependent, an important one being scheduling. The stipulation to maintain QoS delivered to users makes the job of scheduling tasks on resources even more complex. Algorithms such as Hyper-heuristic scheduling algorithm aim at reducing the execution time of tasks [65]. Moreover the parameters to be taken into consideration change with a change in the type of application running on cloud, the type of resources provided by cloud and the presence of private infrastructure.

Scheduling becomes a key aspect due to the pay-as-per-use nature of the Cloud. The factors affecting technique of scheduling used change with change in scenarios. For scheduling in hybrid clouds, the data transfer speed has to be taken into consideration whereas for mobile environments scheduling becomes dependent on context change. Moreover scheduling can be improvised on many fronts such as energy efficiency, cost minimization, Maximization of resource utilization, etc., [66].

Marcos D. A. et al. [67] proposed that devices running applications such as voice assistants and shopping assistants should have components which can find out the user context specific to that application. This would save many resources but the results become worthless if the user context changes and new results have to be generated for the new user context. Thus computation spent to process information with respect to users old context is wasted. It was inferred that for a result to be useful, the execution time of the task has to be less than or equal
to the window of opportunity thus tasks getting completed outside the window of opportunity are aborted reducing wastage of resources.

Fang, Y. et al. [68] proposed a scheduling algorithm for computational tasks. The algorithm uses load balancing to increase the resource utilization and at the same time meets the user's requirements. This algorithm calculates an index which evaluates the load. This index is based upon the execution time of the running tasks, number of virtual machines and number of hosts. If it is greater than the maximum acceptable value then the load is balanced by migrating virtual machines to the host having less load and thus striking a balance in the whole system. This algorithm took into consideration the feature of variable user demand in Cloud Computing and tests proved that it is successful in increasing the resource utilization.

Li, J., et. al [68] compared various scheduling algorithms to find out an efficient solution for the allocation of servers to fulfill users' demand and at the same time taking into consideration the energy consumption. The key feature of this algorithm was the study of the relationship between the user demand and available servers. Tests revealed that for a small load, least first policy works out the best in terms of energy and heat emission without a very significant rejection rate of user requests. For medium loads, priority policy for servers work better wherein, the larger the server, the higher is its priority. Also the order in which the servers are set, plays an important role. The servers with maximum amount of resources should be given the highest priority.

Bossche, R.V.D., et al. [69] proposed an algorithm to schedule deadline constrained tasks in hybrid clouds within the minimum cost while maintaining QoS. Options are available for IT companies to use their pre-existing infrastructure along with public cloud services in the form of a hybrid cloud. Four scheduling policies were put forward by combining two queue policies EDF and FCFS with two cases-sending the cheapest task to public cloud and sending the unfeasible tasks to public cloud. They were compared with each other and with two other cases, first being the case where all tasks are scheduled on the public cloud and second a cost oriented scheduling policy introduced by the authors in a previous work [70]. Tests show that sending cheapest task using Earliest deadline first [71] stood out in terms of performance as in this case less data intensive tasks are sent to the public cloud due to reduction of data transfer cost and this also helps in meeting deadlines and data transfer time is saved.
Li J., et al. [72] proposed two algorithms named Dynamic Cloud List Scheduling (DCLS) and Dynamic Cloud Min-Min Scheduling (DCMMS) in which tasks are categorized in two types namely Advanced Reservation (AR) and Best Effort. The AR tasks are given a higher preference and to run them Best effort tasks are preempted. The two algorithms were proposed for scenarios where workload is heavy and applications tend to have conflicts over resources. These algorithm increased system utilization and decreased energy wastage in such situations. The algorithms accommodate any changes in scheduling which may occur when the task is actually executed and this property is successful in reducing a considerable amount of conflicts in case of heavy traffic. Tests showed that DCMMS gives better performance as compared to DCLS.

Abrishami, S., et al. [73] proposed two scheduling algorithms which were a modification of a grid computing resource allocation algorithm named Partial Critical Paths. Three main differences between Cloud Computing and Grid Computing were taken into consideration. They are:

a) In Cloud Computing, the users are given the freedom to demand limitless resources.

b) The concept of pay-as-per-use.

c) Use of heterogeneous bandwidths.

Two modified algorithms were proposed, first being IaaS Cloud Critical Partial Paths (ICPCP) which is a one phase algorithm where each partial critical path is directly scheduled. Second algorithm IaaS Cloud Partial Critical Path with Deadline Distribution (ICPCPD2) which is a two phase algorithm in which in the first phase tasks are given sub deadlines depending on the deadline for the application and in second phase each task is allocated a server. Tests revealed that ICPCP proved to be more efficient as compared to ICPCPD2.

Rodriguez, A., et al. [74] proposed a heuristic algorithm which used Particle Swarm Optimization to schedule tasks on virtual machines. The algorithm took into consideration all the key features of scheduling in IaaS such as variability in user demand, heterogeneity, meeting of QoS, and change in virtual machine characteristics. In Cloud Computing the resources on which the tasks are to be mapped are variable and are not known in advance, unlike in Grid Computing. Thus the task scheduling becomes a two-step problem in case of
Cloud Computing because prior to allocation of resources to the tasks the resources have to be assessed. This algorithm merges these two steps into single step and finds an optimal solution wherein both the tasks are handled by the algorithm. Tests have proved that the algorithm is successful and overpowers other algorithms such as IC-PCP.

Tawfeek, M.A., et al. [75] proposed a low-level heuristics from which the algorithm chooses are particle swarm optimization, ant colony optimization, Simulated annealing [76] and genetic algorithm [77]. Tests show that this algorithm outperforms general heuristic algorithms and also the computation time of the algorithm is not large as in every iteration only one heuristic algorithm is run unlike in hybrid-heuristic algorithms [88] which combine two or more heuristic algorithms to find the optimal result.

Previous researches state that Artificial Bee Colony [78] [79] [80] method is better than other algorithms of the same category such as Particle Swarm Optimization and Genetic algorithm. It was also tested and compared with HEFT and HEFT/LOSS [81] and proved to be better than them in terms of meeting multiple objectives.

Scheduling algorithms being proposed are based on the algorithm used in operating system like round-robin, first come first serve (FCFS) and so on. These algorithms are being implemented in Eucalyptus and OpenNebula. In Eucalyptus, Greedy and Round-robin algorithms are implemented and scheduling algorithm on power saving for datacenters [82]. In OpenNebula, ranking scheduling policy is being implemented and rank is based on the free CPU’s [83]. However, both the scheduling algorithms do not meet the practical scenarios i.e., they do not take into consideration the cost of the datacenter, properties of the datacenters, their fault rate and VMM characteristics. Yang, Z., et. al., [84] proposed a cost based scheduling algorithm for cloud IaaS, which takes into consideration the cost of resources of datacenter which are cost of CPU, memory and number of cores. Based on these, they calculated the cost according to the user requests. Costs were calculated for different datacenters and the datacenter is allocated to user in the increasing order of cost. There are other algorithms which are based on previous history, genetic algorithm etc. [85] [86] [87] [88].
Luo, L., et al. [89] put forward a strategy to save energy in case of tasks which don't use complete hardware allocated to them. Allocation of unnecessary components is prevented by this strategy. The components can be categorized as CPU, storage, memory and network. The various tasks come under IO intensive, CPU intensive and network intensive tasks. Tests showed that IO intensive tasks consume very less of CPU and majorly work using memory and storage whereas IO tasks over a network use a small percentage of CPU and network apart from memory and storage. Thus according to the specific needs of the task types component policies are generated. Allocation of components is done according to the task type using predefined policy.

(a) Analysis of scheduling algorithms

Each algorithm addresses one or two parameters on the basis of which scheduling is improved. Some target cost optimization, whereas others address shortening the makespan, some work at minimizing the overall energy consumption. At the same time these algorithms struggle to meet the Service Level Agreement and satisfy the users by maintaining the QoS. Different Cloud Structures demand different scheduling strategies. The details regarding the various testing parameters and the criterion for testing for the algorithms has been surveyed are given in Table 2.1.

(b) Conclusion drawn about scheduling algorithms

Scheduling can be optimized on the basis of many factors such as energy efficiency, cost reduction, makespan reduction. Constraints such as maintaining QoS [90] and Service Level Agreement by the Cloud Provider makes scheduling of task a complex job. In different scenarios different scheduling algorithms are applicable. For obtaining energy efficiency in mobile environments such as shopping assistants and voice assistants Context Aware Job Scheduling can be applied. Various Frameworks can be introduced for energy efficient scheduling of resources such as EnaCloud, Green predict and energy optimization resource allocation wherein resources are allocated based on the type of task at hand. These frameworks can also be tested in combination. Green Predict foretells the amount of servers to be kept in an
Table 2.1: Critical Analysis of existing Cloud Algorithms.

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Algorithm</th>
<th>Target environment</th>
<th>Criterion</th>
<th>Simulation environment</th>
<th>Testing Workload</th>
<th>Load Size</th>
<th>Comparison and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Context Aware Scheduling</td>
<td>Mobile application requests running on cloud</td>
<td>QoS improvement and reduction of resource wastage</td>
<td>Event Driven Simulator</td>
<td>Requests for applications such as shopping assistants and voice assistants.</td>
<td>Peak load-keeps tipping during work hours</td>
<td>Successful for a normal workload as well but works best for peak days.</td>
</tr>
<tr>
<td>2</td>
<td>Energy Aware Fault Tolerant Framework</td>
<td>Public Cloud</td>
<td>Reduction in soft errors and maintenance of energy efficiency</td>
<td>Runtime Simulation Engine</td>
<td>Deadline Sensitive Workloads.</td>
<td>Large</td>
<td>Better in terms of energy efficiency as compared to Triple Modular Redundant System.</td>
</tr>
<tr>
<td>3</td>
<td>Green Scheduling Algorithm with neural predictor</td>
<td>Public Cloud</td>
<td>Improve Energy Efficiency and Minimize Drop Rate.</td>
<td>CloudSim and GridSim</td>
<td>Generated Workloads same as requests to NASA and ClarkNet web servers</td>
<td></td>
<td>Four modes are introduced among which prediction along with additional servers is most successful for all workload tested.</td>
</tr>
<tr>
<td>4</td>
<td>EnaCloud</td>
<td>Public Cloud</td>
<td>Improve Energy Efficiency</td>
<td>iVic and Xen hypervisor</td>
<td>Web server and Data Servers ,Compute-Intensive ,Common applications</td>
<td>NA</td>
<td>10 % more energy efficient as compared to FCFS and 13% to Best Fit</td>
</tr>
<tr>
<td>5</td>
<td>Energy Efficient Optimization Method</td>
<td>Public Cloud</td>
<td>Improve Energy Efficiency</td>
<td>Real System</td>
<td>Tasks which don't use the whole of the hardware</td>
<td>NA</td>
<td>It works efficiently in case of I/O intensive tasks wherein CPU usage is negligible.</td>
</tr>
<tr>
<td>6</td>
<td>Least first</td>
<td>Public Cloud</td>
<td>Energy efficiency keeping QoS in mind</td>
<td>Numerical Study</td>
<td>NA</td>
<td>small</td>
<td>A higher priority is given to servers with a larger capacity.</td>
</tr>
<tr>
<td>7</td>
<td>Priority</td>
<td>Public Cloud</td>
<td></td>
<td></td>
<td>NA</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>IC-PCP</td>
<td></td>
<td>CloudSim</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>PSO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>SCS</td>
<td>Public Cloud</td>
<td>Cost Minimization within Deadline</td>
<td>Scientific Workflows</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Hyper-heuristic Scheduling Algorithm</td>
<td>Public Cloud</td>
<td>Reduction of make span</td>
<td>CloudSim and Hadoop Workflow and hadoop map-task</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>Load Balancing Scheduling</td>
<td>Public Cloud</td>
<td>Maximize resource utilization and meet QoS</td>
<td>CloudSim</td>
<td>Computationa l Tasks</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>Earliest Deadline First (EDF)</td>
<td>Hybrid Cloud</td>
<td>Cost Minimization with Deadline Constraints</td>
<td>Java Based Discrete Time Simulator</td>
<td>Batch Type Workloads with Deadlines</td>
<td>Large</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>FCFS</td>
<td>Multiple Clouds with Heterogeneous resource</td>
<td>Cloud Simulation Environment(developed by the author)</td>
<td>Workloads same as requests in</td>
<td>Large</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>Dynamic min-min Scheduling</td>
<td></td>
<td></td>
<td></td>
<td>DCMMES handles situations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SCS and PSO are almost equally efficient in meeting deadlines but PSO incurs a smaller cost as compared to SCS. Both of them perform better when deadlines are relaxed. ICPCP misses out on deadlines in most cases.

Better as compared to hybrid heuristic algorithm due to less computation and case specific algorithm.

Attains maximal utilization of available resources.

EDF overpowers FCFS as it successfully does the task within Deadlines taking into consideration the data transfer speeds and also handles runtime estimation errors successfully.
active mode such that number of idle servers is reduced whereas EnaCloud places the tasks such that minimum number of servers are kept ON. Thus the neural predictor of Green Predict can be used in collaboration with the scheduling algorithm of EnaCloud. Also energy optimization resource scheduling can be introduced in the above approach preventing the allocation of unnecessary resources to the tasks.

For tasks with deadline constraints, Earliest Deadline first [91] can be chosen for batch workloads whereas Particle Swarm Optimization can be chosen for scientific workloads. Make span of the tasks can be reduced using hyper heuristic algorithm which proves to be better than hybrid heuristic algorithm. For a multicloud system Dynamic Cloud Min-Min Scheduling may be applied. Fault Tolerant Scheduling with moderated energy consumption can be achieved using Energy Aware Fault Tolerant Scheduling Framework [92][93][94]. Cloud Computing has many parameters and the scheduling to be applied depends upon user requirements. The aim of any scheduling algorithm is to meet user demand with minimum overheads.

### 2.6.2 VM Migration and Load Balancing

Another important issue of Cloud Environment is VM Migration. Optimization of VM migration is an active area of research as it is frequently required in cloud model. To optimize resource usage and data durability of their users, migrations are adopted [95][96].

By mapping the services onto the Virtual Machines (VMs), where multiple VMs can run onto a single physical server, the problems related to heterogeneity in hardware, software and platforms could easily be solved. Consequently the terms and conditions for the SLAs can be agreed upon between the Cloud Service Providers and the service users [97][98].
The need to migrate the VMs arises because of the overloading of one physical machine that runs multiple VMs onto it. Consequently, there is overheating which may degrade the performance and may also lead to faulty operations and system crashes. Hence a conditional load balancing is required for better manageability of the cluster of servers [99].

The narrow interface between a virtualized OS and the virtual machine monitor (VMM) avoids the problem of residual dependencies. There are various ways to migrate a VM from one physical node to another. “Pure stop-and-copy” or “Cold” migration technique halts the VM and copies all its associated memory pages to another pre-identified destination node and then resumes the VM on it. On the contrary, few selected hypervisors like Xen and VMWare do a “Live” or “Hot” migration. The advantage of the latter method is that even the concerned applications and processes are unaware of the VM migration [100].

VM migration supersedes Process Migration except in some cases that occur due to the narrow interface between a virtualized OS and the virtual machine monitor (VMM) where it avoids the problem of residual dependencies. VM migration has the advantage of transferring internal memory states in a very consistent and efficient way. Process migration demonstrates a functionality of transferring a process running on one machine to the other. But, there is an inherent difference in the operating concepts of virtual machine migration and process migration [15]. Though in practice, migrating the process is difficult and quite complex as it should take care of legacy applications and at the same time it should also leverage the currently installed and related large databases of operating systems and maintain independence on different machines. These can be overcome by using a VMM based migration. VMMs such as VMware use Hardware abstraction to encapsulate the complete OS environment in such a way that it can be suspended from one machine and resumed onto the other one provided there are inherent similarities in the system architectures of the operating systems [96].

VM Load Balancing is crucial characteristics of system virtualization, allocate and shift the running applications dynamically to other physical machines as and when the load increases on any particular machine [101]. As a consequence of complexity aroused due to the vast heterogeneity in terms of underlying hardware, operating systems environment, platforms and communication technologies, and that too at the run time, it is inevitably important to address the performance and issues related to smooth delivery of services to the end users.
performance optimization and cross platform operational issues are the need of the hour for Cloud environments, and that too with a guaranteed level of services [102].

Load balancing is the capability of the system that allows the VM hosted applications to be transparently allocated a VM which has least load so that optimal resource utilization is achieved.

There are many approaches to load balancing viz. Static and dynamic etc. [103]. In addition, some hybrid approaches are also adopted. Dynamic Load Balancing, decision is taken at runtime according to the existing situation, whereas in static it is not so. Neither of them is superior or inferior but the selection of algorithm depends on the application requirements [104].

(a) Static load balancing

Static Load Balancing (SLB) refers to the load balancing algorithm that distributes the load strictly on the basis of certain predefined rules relating to the nature of input loads. It does not consider which node is receiving more or less load. In all static algorithms final selection of the virtual machine is done immediately after creation of application. Further it cannot be changed while in execution. These static load balancing techniques are suitable for a system in which load is limited and request of clients are also limited. Nowadays load on cloud servers is increasing and also the load is not static so we need more efficient algorithms a static load balancing algorithms. Some basic algorithms for static load balancing are:

- **Round Robin Algorithm**
  Whenever a new application comes it is assigned a virtual machine in a round robin fashion. In general, basic idea of round robin [105] is to reduce message passing between various virtual machines and reduce communication delay. Thus it is independent of the state of the system. When coming applications are of similar load then round robin works very well as it reduces the communication delay due to inter-process communication. Thus round-robin has best performance for this special purpose application of similar load, but does not give a good performance for general cases.
• **Randomised Algorithm**
  Random numbers are distributed on a basis of a statistical distribution and assigned to virtual machines. Incoming applications are distributed according to these randomly generated numbers. This algorithm is applicable when we have many virtual machines as compared to processes.

• **Central Manager Algorithm**
  In this algorithm [87], there is a master virtual machine and others are slave virtual machines which are assigned applications to be executed. Master virtual machine’s task is to gather load information of all the slave virtual machines and assign the coming application to the least loaded slave virtual machine.

![figure 2.2: Manager collecting information of slaves](image)

• **Threshold Algorithm**
  In this algorithm the current virtual machine is decided on the basis of two values of upper (t_upper) and lower (t_lower) threshold. Virtual machine is assigned a state depending on its current load compared to these threshold values. If current load is less than the t_lower virtual machine is assigned a Under Loaded State, if its greater than t_upper the state is overloaded, if it is between the two threshold values then the state is Medium. Initially all virtual machines have under loaded state. But as the system advances the load level limit of a virtual machine may change and its state may change. If the state changes then it sends message to all other virtual machines notifying the change, so that they can maintain the load state of entire system. When a process arrives and the local virtual machine is under
loaded then it executes the application else calls for the remote virtual machine. If no under loaded virtual machine exists then application is executed locally only.

(b) Dynamic load balancing

Dynamic Load Balancing (DLB) techniques [49][50][51] provide a method to dynamically allocate load based on self-adapting distribution and intelligent distribution. Here, it is distributed at runtime based on the new information collected. Mainly these techniques are based on greedy algorithmic approaches. Basic algorithms for dynamic load balancing are:

- **Central Queue Algorithm**
  The host virtual machine maintains a central queue of all the applications. This queue is shared by all the processes. New applications are added and pending applications are maintained in a cyclic FIFO order in the queue. When a virtual machine is free it will request for application for executing to the host and host assigns the application next in queue to the virtual machine which is demanding the request. If there is no application for execution in the queue then the request is buffered in queue form. And request is answered when new application arrives.

- **Load Queue Algorithm**
  Here the applications are assigned virtual machines similar to the static algorithm but a virtual machine here can initiate application migration. Initially all under loaded virtual machines are assigned applications. The applications are assigned following some static algorithm. Then if a virtual machine’s load goes below the lower threshold value then it asks for load from other machines and initiates migration process. Several scheduling algorithms have been re-invented and tested for cloud environment, namely intelligent scheduling algorithms, autonomous scheduling algorithms, agent-based negotiable scheduling algorithm, centralized scheduling algorithms. These algorithms are used by many popular Cloud Service Providers for balancing the loads on their virtual machines; a popular example could be VM Ware Distributed Resource Scheduler (VMDRS).

2.6.3 Trust, Fault and Energy awareness

A major issue in distributed and cloud environment is security and reliability [106] viz. how to decide that which datacenter is trust worthy and reliable as per the application requirement.
For example, in grid computing, trust can either be decided on the basis of its capability to compute multiple tasks in parallel or on the basis of its failure rate. Reliability and trust Models [107][108][109] are used to enhance security, reliability and QoS in Grid and Distributed environment. One major issue in cloud computing adoption is security of data. This is because data is frequently required to be moved from data centres to end user and vice versa. Developing and building a trust on the CSPs is a major design challenge. This inherently involves performance, system design, data governance and risk management.

In general, trust management is one of a major concern for CSPs. Users are skeptical about the data centers for using their services. To avoid malicious data centers, users prefer to use the datacenter that are more reliable based on the previous service history. So, trust management plays an important role and has influence in the decision making process of user. Various trust models are developed [110][111][112][113]. Trust models and their types are enlisted below:

(a) **Trust Aware Model**

Trust can be defined as the firm belief with some associated attributes. A default trust value can be assigned to an entity based on its previous behavior. It can be applied only within a specific time period as it may vary with time [114]. This means that the firm belief is dynamic value which varies with time.

Trust can be divided into different categories [115]:

- **Blind trust**: A default trust before the execution of any event is initiated with unknown entities.

- **Conditional trust**: Conditional trust is subject to some conditions that may be used for evolutions based on the limitations and constraints of the users.

- **Unconditional trust**: This is probabilistic in nature and can be configured by administrators without bothering about the outcomes.

Many classification of trust models are proposed that are based on system parameters [116][117][118]. These trust values may vary as per the activities performed on the entities [119]. Reputation based trust model is also proposed for grid computing in which the past
experience has been taken into account for calculating the trust value. A decay function based on last experience is defined and is used for trust value calculations [25].

(b) Fault Aware Model

Fault tolerance is a major issue because failure in cloud model is inevitable. Various CSPs have history of major failures, downtimes, network breaks and many other failure issues. Hence reliability, outage and failure is one major issue. Various fault tolerant recovery and redundancy measures are taken into account to minimize the problem, e.g. Recovery-Oriented-Computing (ROC), Google File Systems, and Server replication measures are adopted in Big Data centres [120].

Several concerns have been highlighted regarding the adoption of cloud computing namely security and privacy, interoperability support, compliance to heterogeneous environments and availability. Certain examples of failures are cited from different service providers like Amazon’s S3, Google’s Application Engine services, Citrix’s services and Blackberry services, which suffered failures. Although 100% availability can’t be guaranteed by the service providers hence an alternate arrangement to maintain a backup or on-premise storage for mission-critical data is suggested. Resolution of problems must also be addressed by the vendors. These issues including compliance are not yet major concerns as cloud computing is in its initial development stage, but these issues may become a major concern tomorrow [1][12].

A similar algorithm based on cost was proposed [84] for cloud IaaS, which takes into consideration the cost of resources of the datacenter comprising cost of CPU, memory and number of cores. Based on these, cost was calculated according to user requests. Cost was calculated at different datacenters and the datacenter with optimal cost was allocated to the user. Similar improved cost based algorithm was proposed [85], which takes into consideration additional dependent parameters like MIPS and CPU of datacenters for cost.

An algorithm based on the history was proposed [85], which uses vector cloud computing i.e., it uses the previous history of resources allocated to schedule new requests.
All above proposals are based on cost and profit in many terms but the proposal given by [119] discussed about the cloud’s Software as a Service (SaaS) provided by Hadoop and proposed a clustered scheduling algorithm for SaaS. This algorithm used map reduction technique to reduce the task and find the best suited resource based on profit based on graph.

Genetic algorithm for scheduling of resources was proposed by [88] which described the way a genetic algorithm can be adopted for resource allocation in cloud. Priority based scheduling algorithm are proposed in grid computing environment which is similar to cloud environment. Weifeng, S., et. al., [44] proposed a priority based scheduling algorithm in grid computing using few parameters to evaluate the grid parameters. In this a directed acyclic graph is drawn based on the priority and then scheduling was done based on descending order of priority. The algorithm was compared with Min-max and Min-min algorithm. Similar to this algorithm Lee, Z., et. al., [121] proposed a priority based scheduling algorithm for cloud environment. In this priority of the request or task dynamically changes based on time spent in the queue.

Huang, Q., et al. [87] proposed a multiple QoS based priority scheduling algorithm for cloud. Multiple parameters related to a request are taken into consideration like cost, time delay, deadline time, number of processors requested and MIPS.

\[ C = \frac{pm}{MIPS} \]

where, C is the cost of resource.pm refers to the budget or the dead line for a task. Expected time to complete task is referred as:

\[ \text{Expected time} = \frac{\sum_k \text{length}(t) + \sum_q \text{length}(t)}{\text{resource.rate}} \]
These scheduling algorithms are implemented in real clouds. For example, in Eucalyptus, Greedy and round-robin algorithms are implemented. All algorithms mentioned above do not take into consideration fault rate at the datacenters. Since datacenters are the resource providers and if the resource provider is itself faulty, then the expectations of QoS by users cannot be fulfilled.

Agent based distributed scheduling algorithms are studied that can exploit the characteristics of distributed architecture which can be efficiently used in the clouds. Grid computing is one example that exploits the benefits of agent based algorithms. To overcome the drawbacks of basic algorithm over agent based algorithm, Ant Colony algorithm was proposed for job scheduling in grid environment by Ku, M., [122]. This algorithm has a drawback that the next request is allocated on the basis of allocation of previous request. Due to the property of ant, that it always follows the path which has the highest pheromone, this algorithm is best suited for similar type of requests. If the types of requests differ, then the profit function for each request was required to be evaluated, as in case of cloud IaaS [123].

![Ant Colony System Architecture](image)

**Figure 2.4:** Ant Colony System Architecture

These scheduling algorithms only considered either the power or simple resource allocation techniques. To overcome these problems, a honey bee based fault aware scheduling algorithm has been proposed. This algorithm was previously used in grid computing and cloud computing environment for load balancing [86]. In these algorithms also, the performance of datacenter and QoS provided by datacenters were still not taken into consideration. Honey Bee Algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal solution [124].

Currently network traffic is exploding with rapid development of internet. CDN is a popular solution to balance the heavy network traffic over a distributed system which acts as a single
system for users. Many proposals [125][126][127] are made to balance load based on response
time, cost and load on server. Energy consumption and rate of data transfer based models are
also proposed for CDN [128][129][130]. We have proposed a scalable and reliable
architecture for CDN along with fault aware load balancing algorithm. Although many
existing approaches address the issue of load balancing in CDN but they do not take into
consideration failures at servers which increase with increase in load. The proposed algorithm
takes into consideration both load and failure over a server and scalability of CDNs. To
summarize, the proposed algorithm tries to solve the problem of scalability and load balancing
in CDN and to overcome the drawbacks of existing techniques [131].

Michael D [132] proposed least-loaded (LL) load balancing algorithm and is best example of
dynamic load balancing. In this requests are distributed to a server which is least loaded, in
terms of queue length, until it is saturated. To overcome the drawback of LL, response time
based algorithm was proposed. Carter and Crovella [133] proposed a response time based load
balancing algorithm which diverts the load to fastest response time server available.

Manfredi, S., et. al. [134] proposed load balancing taking care of load over the system and the
capability of server to process the request. In this algorithm each server is assumed to have a
fixed queue size and if the queue length increases load balancing is initiated and a least loaded
server with empty queue is selected to balance the load.

Matthieu, G et al. [135] discussed load balancing which takes care of hardware fault based on
Byzantine fault, i.e. an error in the system may lead to subsequent failure in the system. On the
other hand, software failure, which covers request because of resource unavailability or high
queue length also leads to request failure. In distributed system, failure can be correlated with
a workload using spatial and temporal correlation between workload type and intensity of
failure at different servers in short interval of time. Spatial correlation refers to multiple
failures occurring on different servers in short interval of time. Temporal correlation means
skewness of the failure spread over time, where correlation between failure is the time
between two consecutive failure. Let Ts(Fi), Ts(Fj) be the start time of failures i, j
respectively. Temporal correlation can be calculated as:
\[ L_{ij} = \| F_i + F_j \| = \left| T_s(F_i) - T_s(F_j) \right| \]  

\[ C_t(L) = 1 - \alpha \frac{L}{\Theta} + \beta \left( \frac{L}{\Theta} \right)^3 \]

where \( \Theta \) is an adjustable time scale parameter for determining the temporal correlation between two failure events, and \( \alpha \) and \( \beta \) are positive constants where

\[ \alpha = \beta + 1 \]

A hybrid approach based on random and LL was proposed by Mitzenmacher M. [136] two random choice algorithm (2RC). In this 2 servers are randomly chosen and least loaded among them is selected. This approach is beneficial as there are large number of servers and random choice algorithm helps to provide an equal probability of a server being selected.

Naoya, M et al. [127] proposed a mechanism that automatically optimizes distribution period traffic localization. A pictorial view of this model is shown in figure 2.5.

![Figure 2.5: Cost based Distribution](image)

Mahajan et al. [105] proposed an affinity based round robin load balancing algorithm for cloud to balance requests in cloud infrastructure. This algorithm takes into consideration the load over the data center and then if the datacenter is over loaded initiates load balancing in round robin fashion. However all existing techniques [137][138][139] consider the system non faulty and do not take into consideration reliability due to request failure and failure in
system. So to overcome all these issues a fault and reliability aware distributed load balancing algorithm is proposed in this work.

(c) Energy Aware model

Energy Aware computing and resource allocation strategies that may contribute to green computing are inherently needed in cloud computing model as huge amount of power is consumed in data centers to perform the computations [140].

For scheduling in hybrid clouds, the data transfer speed has to be taken into consideration whereas for mobile environments scheduling becomes dependent on context change. Moreover scheduling can be improvised on many fronts such as energy efficiency, cost minimization, Maximization of resource utilization, etc. Scheduling is a major factor needing attention in the field of cloud computing. Amount of energy consumed, cost incurred to provide services over the cloud, amount of execution time, are major causes of concern and improvising the scheduling of tasks helps in minimizing these. Techniques such as VM resizing introduced by the EnaCloud algorithm and the neural predictor put forward by the green predict scheduling algorithm help in minimizing the energy consumption [141]. Users also expect the makespan of tasks to be as small as possible. Algorithms such as Hyper-heuristic scheduling algorithm aim at reducing the execution time of tasks.

With increasing use of Cloud environment and data centers, Energy consumption by data centers is a major cause of concern. Energy consumption by data centers is growing at a rate of more than 10% every year, as per the report published by Department of Energy, US [142]. This substantial increase in energy consumption is a major challenge for the computing experts and for the data center service providers, viz. Google, Microsoft, HP, Dell, IBM and other such Cloud Infrastructure Service Providers.

Efficient use of energy can benefit in many ways such as cost saving, efficient utilization of resources [143] as also environment protection. So energy consumption, cost and time are important decision making factors for both users as well as cloud service providers. QoS conscious scheduling of jobs along with energy awareness is very important, especially in cloud environment, where large datacenters are to be maintained involving huge computations. Hence, energy aware computations and scheduling is a big future concern that
may heavily contribute to maintenance of the nature’s environmental systems, ecological balances and may avoid direct and indirect health hazards. Omni-directional benefits are the outcome of using energy aware scheduling techniques [144] for Cloud environments and that too without compromising the Quality of Service. An energy aware resource allocation algorithm which takes into consideration energy consumption of datacenter to prioritize them based on energy efficiency. This paper also discussed about different energy aware load balancing and resource allocation algorithm.

Various methods are especially discussed and designed for emerging cloud environments including the server connections, client connections and the session management between the clients and the servers [145][146]. Various systems and techniques have been developed and abstracted in different layers to practically facilitate the cloud services using the resources placed onto the cloud. These abstract layers have been patented by different organizations [147]. Various techniques are developed to predict the future utilization of cloud resources that may be requested by the users. Based on such predictions, more accurate allocations can be estimated to reduce the processing time [148]. As the cloud resources are requested and utilized on pay-as-per-use basis, so, some well-defined billing technologies were developed to measure the usage in an accurate and precise way [149]. Managing the resources on the cloud is a major concern. As the servers are deployed in a distributed fashion at various locations. Therefore, the complexity of coordination to access the resources requested by the users. Various resource deployment strategies are developed and adopted to manage resources in cloud environment [149].

Most Service providers do not take into consideration multidimensional resource usages like, CPU, memory and network bandwidth, while placing the Virtual Machines (VMs) onto the physical machines. The major reason for this is that a lot of pre-computations are required to find the resource usages in varying and heterogeneous nature of workloads.

Different algorithms focus on specific resource parameters, accordingly, various techniques are used to reduce the power consumption, but none of them addresses the adverse effects of power reduction. In many cases, the Quality of Service (QoS) may be affected, for example, reducing the power consumption may slowdown the overall system speed. In such cases, actual energy consumed for completing a particular task may even be more than what it would
have consumed without using the energy saving scheduling techniques. This may also increase the make-span time and consequently increasing the processing cost for a particular task, which is undesirable.

Hence, QoS based Energy Aware Scheduling techniques are the need of the hour in Cloud Computing Environments.

Luo, L., et al. [89] put forward a strategy to save energy in case of tasks which don't use the all the hardware allocated to them. Allocation of unnecessary components is avoided by this strategy. The components can be categorized as CPU, storage, memory and network. The various tasks come under IO intensive, CPU intensive and network intensive tasks. Tests showed that IO intensive tasks consume very less of CPU and majorly work using memory and storage whereas IO tasks over a network use a small percentage of CPU and network apart from memory and storage. Thus according to the specific needs of the task types component allocation policies are generated. Allocation of components is done according to the task type using predefined policy.

Bo, L., et al. [141] proposed a strategy to save energy wasted due to servers sitting idle during the placement of applications with continuously varying requirements. The key to the algorithm is that a task is enveloped by a VM to enable transferring of the task to another resource while the task is in progress. This helps in reducing the number of active servers. During insertion, task at hand replaces a smaller task if it can and the smaller one is now inserted in the same manner. The principle behind this strategy is that smaller tasks can be easily accommodated into already active servers as compared to bigger tasks.

Vinh, T., et. al. [151] presented a Green scheduling algorithm based on the decision made by a neural network predictor. It was observed that powering off the servers while they were not in use saved a lot more energy as compared to lowering the voltage. The problem encountered was to assess the future demand and power off the extra servers as per the assessment because in case of a wrong prediction the drop rate of user requests increases and service level agreement is not assured. The neural network predictor put forward by this paper gives a solution to this problem. It assesses the need of number of servers in the future based on the previous records. The predictor works in four modes namely optimal mode, conventional mode,
predictor mode and predictor mode with additional 20% servers. Tests proved that predictor mode with additional 20% servers proved to be the best as it helped to save the maximum amount of energy at minimum drop rate.

Energy Aware Computation in Cloud environment has become one of the major causes of concern for researchers and Cloud Service Providers. Lots of works are going on in this direction to address the issue at software and hardware levels. Some heuristics based algorithms [38] were suggested to manage the allocations of resources. An efficient architectural framework is proposed where numerous Green Compute and system resource utilization under type of loads and amount of loads on the data centers are considered. Task Consolidation based [33] on energy consumed by each of the tasks were also suggested to reduce the energy usage where scheduling of the VMs were based on current CPU usage by any task in hand. Static and Dynamic VM migration techniques [100] in which some strategically selected VMs that use more energy onto the currently allocated machines were forced to migrate onto another physical machine. These migrations were based on loads on current machine, computational load of the individual VMs and system capacity. Dynamic scheduling algorithms [104] for reducing the energy consumption were also suggested for the data centers. These algorithms were based on the overall runtime and levels of imbalances between various underlying system resources, viz, CPU, memory and network bandwidth consumed by the physical machines as well as the Virtual Machines running on those machines. Some Hotspot and Coldspot based migration techniques [35] techniques were also suggested to reduce the energy consumption. These techniques were based on the upper and lower temperature thresholds of the CPU. CPU temperature proportionally increases when the computational load on it increases. This temperature further increases if it is highly used and reaches a threshold known as Hotspot, beyond which the system performance degrades inspite of more CPU usage. On the other hand, if the CPU temperature of a machine falls below a certain threshold, all VMs of such machines are again migrated to another normal machine. This saves the energy by switching off such under-loaded machines. Consolidating the Virtual Machines onto the optimally running physical machines and switching off the remaining physical machines based on work-load awareness were also suggested to save energy [36][37]. Scheduling techniques based on either alone or a combination of workflows based on cost
constraints, deadline aware and other energy aware parameters for homogeneous and heterogeneous processors were also implemented [38].

All the above mentioned algorithms and techniques significantly contributed to the energy savings but the Quality of Service parameters were neglected. For example, the lower the energy consumption, lower is the frequency usage, which further lowers the voltage of the machine. This reduction in voltage may slow down the computational speed and consequently the Quality of Service may get affected.

![Resource Allocation Strategies for Cloud IaaS Environment using Virtualization](image)

**Figure. 2.6:** Categorization of Virtualization based Resource allocation strategies for IaaS Environment

---

52
2.7 IDENTIFIED ISSUES

In cloud computing environment, large amount of data movement takes place in between users and data centers. Cloud model caters on-demand requests. Its design should be highly scalable. This implies that the demands for underlying system resources may vary in an unpredictable manner. Hence computationally efficient, on-time, trusted, reliable, fault tolerant and energy efficient resource allocation is needed in cloud environment.

Resource allocation issues in cloud computing environment using virtualization can be addressed using two broad approaches:

2.7.1. ISSUE 1: System Parameter based Approaches: System parameters like, CPU usage and energy consumption may be desired system design issues. We may wish to use the CPU cycles of a system to its maximum capability. Use of CPU beyond certain limits may increase the system temperature to levels where performance may degrade due to overload which also results in high energy consumption by CPUs and cooling systems etc. This requires to migrate the running VMs to another physical machines. This conditional load balancing can be achieved by generating a trigger based on hotspot defined as “A hot spot/cold spot is an undesirable tightly-focused local temperature variation which often occurs when data center is highly loaded or the equipment is improperly cooled. When servers are installed into a tight, highly integrated chassis, the cumulative heat may not be dissipated adequately by the available cooling resources. In extreme cases, the excess heating may lead processors, memory modules or other server components to fail”. In generation of trigger, an identified VM is live migrated to another most suitable physical machine based on its temperature. Also a dynamic NFS based load balancing technique may be advanced for efficient resource utilization. User requests may be allocated to VMs so as to optimally balance the load on VMs.

2.7.2 ISSUE 2: Quality of Service based Approaches: To improve QoS, user requests may be allocated based on:

- **Trust:** A trust based framework and an algorithm may be used to forward the requests to trusted data centers and avoids allocation of user requests to untrusted data centers.
- **Fault:** Fault Aware algorithm may be proposed to reduce request failures at the data centers.

- **Energy:** Combination of energy aware, makesapn time and processing cost based algorithms can be used for optimal resource, time and cost utilization.

Three broad approaches may be used for improving the Quality of Service. These are Trust Aware, Fault Aware and Energy Aware.

- **Trust Aware**
  
  An open source trust based framework can be proposed. This framework may use both novel trust based scheduling and load balancing algorithms. The trusted Cloud model can provide IaaS. It may ensure high QoS to users by running their VMs on trusted data centers. A scheduling algorithm may be proposed that may consider the fault rate of data centers in addition to normal QoS parameters that may significantly improve the performance in cloud IaaS.

- **Fault Aware**
  
  Since Content Delivery Networks are used for data movement in cloud environment that supports high speed communication. A fault aware load balancing algorithm for CDNs may be suggested that may further improve both QoS and reliability. Performance of existing load balancing algorithm shall be investigated and compared in the presence of faults. The performance of the suggested algorithm shall be compared with other existing algorithms used in CDNs to observe the experimental results for better robustness and resilience to fault without affecting QoS.

- **Energy Aware**
  
  Energy aware scheduling techniques for Cloud environment may also be proposed. Software based scheduling and testing can be implemented and tested with DVFS. A comparative study can be made of these algorithms to observe the improvements.

### 2.8 THESIS OBJECTIVE
The main objective is to explore Resource Allocation Strategies in Cloud Computing environment for optimal utilization of resource so as to minimize cost, time and energy consumption and maximize trust, fault tolerance and reliability.

2.8.1 OBJECTIVE 1
To develop a Trigger based VM Migration technique that gets activated when CPU temperature increases beyond an upper threshold value, called Hotspot. Based on the Hotspot threshold, a VM be live migrated to another best threshold based identified available physical machine.

To develop a Network File System (NFS) based dynamic load balancing strategy for better system resource utilization. This can be done by selecting the most suitable VM for load allocation. A dynamic load balancing algorithm may be developed and implemented on Xen VMM for VM load balancing based on NFS that require less computational overhead as compared to normally used Migration strategies and other traditionally adopted load balancing techniques.

2.8.2 OBJECTIVE 2
- To develop a Trust Aware Cloud framework that is based on the trust value of data centers. Further, A QoS aware scheduling algorithm be proposed for cloud IaaS incorporating fault rate of data centers in addition to existing parameters, like, VM initiation time, price per hour usage, VM processing speed and bandwidth to achieve significant improvement in requests being allocated to trusted data centers.

- To develop a fault aware load balancing algorithm for CDNs that may improve the QoS and reliability of the systems. The effect of network failure on QoS and reliability of the system may be implemented in the presence of high request rate and network traffic. Performance of existing load balancing algorithm may be investigated and compared in faulty environment.

- To develop Software based scheduling and testing using DVFS based experiments for minimizing the processing cost, makespan time in Energy Aware environment so as to minimize energy consumption without compromising QoS.
2.9 CONCLUSION

Various literatures are surveyed related to resource allocations and related issues that uses virtualization technology. It is concluded that research related to Virtualization is now at its slope of enlightenment in cloud computing and other related scenarios. It has given birth to creation of a new way to the IT service class. Various VMs were studied and explored. It is concluded that this new software environment acts as a real physical machine by leveraging their complete functionalities under the supervisor and control of the VMM by virtualizing and multiplexing the underlying computer system resources.

Literatures related to live migration of the VMs were studied which consists of Guest Operating System and applications running on it. VM migration optimizes the system performance by dynamically balancing the load. It was concluded that based on the Hotspot threshold, a VM can be migrated to another best threshold based identified physical machine available. Some load balancing strategies were explored that gives better system resource utilization. These can be achieved by algorithmically selecting the most suitable VMs.

Some trust based scheduling and load balancing strategies were explored that may leverage better allocation of resources to further enhances the Quality of Service (QoS). Various Fault aware load balancing algorithms were studied that may effectively manage the load over cloud servers and to maintain the overall system performance with better QoS. Since CDNs are used in Cloud to maintain better QoS, so CDNs and their applications were explored for Cloud environments as it is highly used in Cloud environments to achieve high speed. Study suggested that CDNs offer services that improve network performance in terms of better bandwidth utilization, improved accessibility and reduced load on the servers. Some energy saving related literatures were also explored as it is an important resource and it substantially contributes to green computing. Literatures suggested that its efficient use can benefit in many ways such as cost saving, efficient utilization of resources and also saving the energy consumption, processing cost and makespan time. It was observed that QoS conscious scheduling of jobs along with energy awareness is very important, especially in cloud environment, where large datacenters are to be maintained with involvement of extremely large computations.