CHAPTER 2

LITERATURE REVIEW

2.1. Introduction
In this section, we have discoursed various existing approaches from the field of resource allocation, load balancing, green computing and trust management. This section aims to identify the research gaps and focus on the current state of artwork in the field of resource allocation, load balancing, cost efficiency and green computing.

2.2. Resource Allocation
Many researchers have done research and introduce us some beneficial and optimal scheduling algorithm. [22] Proposed a modified Min-Min algorithm, this chooses the task with least completion time and schedule to serve accordingly. Author has proposed load balancing Min-Min algorithm which having basic properties of Min-Min algorithm and consider minimizing completion of all request. In this proposal three level of service models are used.

1. Request manager- To take request and forward to Service managers.
2. Service manager- various manger works or task and dispatch them to respective service node.
3. Service Node- Service node provide service to request which came to request mode

They have merged two approaches (OLB Opportunistic load balancing and load balance min-min) scheduling algorithms in this model. The main focus of combined approaches is to
distribute the request or dispatched task basis of their completion time to suitable service node via an agent. This approach not saying about main system, suppose if request are somehow moving or scheduled in the same server and due to lots of load sever need more power to complete these request and more physical heat will generate and to stop heating system need an external cooling system which also lead to extra power source and one more important thing is due to overheating system performance slow down. The same way [23] proposed and another algorithm for task scheduling, this paper proposed VM resource allocation basis on genetic algorithm to avoid dynamic VM migration to completion of request. They have proposed a strategy to share or allow resource equally to VM so it can work fast and minimize response time to subscribe. They also proposed hotspot memory (virtual memory) assignment and dispose that after completion of request via remapping of VM migration. Here VMware distribution tool is used to schedule computation work in a virtual environment. As genetic algorithm characteristics is to find best, fittest VM in terms of Cloud computation.

This paper checks fitness of each VM and schedule task accordingly. When creating a VM a process executes to create that and increase process work that also lead to more process and increase energy consumption. Hu, Jinhua et al. [24] Proposed another scheduling algorithm, this paper proposed an approach for collective collaborative computing on trust model. The trust value taking as a factor for task scheduling, trust value mutually took from consumers as well service provider, which make it fail free execution environment. Here they have proposed a mathematical equation to calculate the Reputation point which enhances the reputation of VM in terms of fast execution and type of task. If the reputation of VM is high them more task allocation will be happening to that VM. To calculate Reputation many factors have to consider which also reflect QoS of cloud computing. This paper also proposed a way to serve a request reliability, as well trust management with a reputation of VM factor which are lead to trustworthy. Trust has calculated by a mathematical equation and schedule accordingly.

Hu, Jinhua et al. [25] proposed a live VM migration algorithm, this paper proposed a method for VM live migration with various resource reservation system. VM migration is taking place on the basis of source machine load, if the load is high then it can wear, during execution of the request it migrates the VM to another server or data centers to complete the task without interruption for better performance. Resource reservation done both sides, i.e.,
Source machine and target machine as well will in such manner CAP (maximum availability of CPU) allocate them and adjust memory resource dynamically. At the end of target machine, they properties time bound program which will keep monitoring for cup resource utilization. Memory Reservation done by allocating crating certain number of VM and when the migration process comes into existence these VM got shut down to evacuate the space to migrate VM. Sometime it may be possible that target machine not having enough space to migrate in such condition that physical machine should remove from candidate machine for migration and which physical machine having the capability or enough space will lead to migrate VM. This paper implemented and simulated using Xen Virtualization.

Barroso et al.[26] This paper proposed an algorithm, dynamic and integrated resource scheduling algorithm for cloud data center which balance load between servers in overall run time of request, here they are migrating an application from one data center to another without interruption. Here they are introducing some measurement to ensure load balancing. They have given a mathematical reputation to calculate imbalance load to calculate average utilization to its threshold value to balance load. To implement DAIRS they have used physical server with physical cluster and Virtual servers with virtual cluster. Application migration saves time instead of migrating whole VM data. Zhanjie Wang [27] proposed an dynamic algorithm for resource allocation in cloud using fuzzy logic and pattern recognition based on power and storage parameters. The propose algorithm is derived from FastBid algorithm. The algorithm tries to improve the network traffic and communication load over the system. The algorithm shows better result than Min- Min algorithm in term of makespan and network load.

Parvathy S. Pillai [28] et al. proposed a novel resource allocation algorithm derived from game theory for resource allocation in cloud. In this work author has used uncertainty principle of game theory for allocation of virtual machines in cloud. This work improves the communication cost and resource wastage over the system. Abdullah Yousafzai [29] et al. surveyed and reviewed resource allocation algorithm in cloud. This work contributed an review and comparative study or current state of art cloud resource scheduling and allocation algorithms for cloud. Moreover this article proposes an taxonomy for resource allocation in cloud environment, which shows various ways to solve the issue of resource allocation and different aspects of resource allocation. Figure given below shows the taxonomy.
Many other resource allocation algorithm are been proposed [88 -97] using various dynamic techniques to improve the performance of the system are been studied.

![Cloud Resource Allocation Schemes](image)

Figure 2.1 Taxonomy for resource allocation in cloud [29]

### 2.3. Load Balancing Algorithms

Various load balancing tactics have been proposed till now which can also be classified into static and dynamic in nature [30]. Yamamoto et al. [31] proposed a disbursed strategy to balance the load using replication of data. Authors have proposed two replication approaches 1) in the route random replication procedure, replicas saved within the peers along the trail of soliciting for to look. 2) In the course adaptive replication procedure replicas saved most effective within the peers in step with their likelihood of replication. This paper does not use the knowledge about the ability of servers for selection of server.

In [32] Rao et al. have offered a framework for load balancing in distributed environment, named as HiGLOB. Right here, authors have used two principal add-ons 1) histogram supervisor - generates a histogram to preserve a global information regarding the distribution of the load within the system, and 2) load-balancing manager - reallocates the load at any time when the node becomes overloaded or under loaded. Nevertheless, there's overhead associated at the same time setting up and preserving the histograms.

Zeng et al. [33] have proposed a load re-balancing algorithm to work out the crisis of load balancing in distributed environment. They have additionally ensured the reliability of the process where one chunk of a file and two duplicate copies are allocated in three exceptional
different servers at a time. In this algorithm author uses the master server periodically for checking of chunk servers and to differentiate which chunk server is over-loaded and which is not. Nonetheless, this master server turns into a single-point failure for the load balancer. Fan et al. [34] have proposed an adaptive load optimization algorithm (AFLBA) for the Hadoop distributed file procedure which uses two modes: 1) disk utilization expense system and 2) carrier blocking off rate system. The proposed algorithm uses the storage average utilization of each data node and probability of blocking consumer request of each knowledge node. Since this algorithm isn't disbursed so it creates a performance bottleneck node within the HDFS.

Hasio et al. [35] and Chung et al [36] have proposed an improved load balancing algorithm for distributed file system to overcome the issue of bottleneck and improve the performance of system. They have proposed to use CHORD protocol for creation the node server.

Many other load balancing approaches are been proposed to avoid the condition of over loading [101, 102] using max-min, min-min and dynamic strategy.

2.4. Power Efficient Algorithms

Several researchers have introduced various models and methods to conserve energy. Some of them are discoursed below.

Louis Rilling et al. [37] proposed a virtual infrastructure optimization solution using the ant colony optimization algorithm for finding better paths through graphs. The most common approach while performing workload consolidation is that the workload is allotted to a physical machine (e.g. CPU) and those resources which require excessive provisioning are converted into a lower power state.

Osvaldo Adilson de Carvalho Junior et al. proposes the use of a function that can ensure the most appropriate behaviour to the principles of Green IT but not the quality of service. For this he proposes the use of GreenMACC (Meta-scheduling Green Architecture) and its module LRAM (Local Resource Allocation Manager) to automate the execution of all scheduling policies implemented in the Scheduling Policies Module so as to provide Quality of Service in Cloud Computing and determine its flexibility. [38] Task consolidation is an efficient method which is used to reduce power consumption by increasing the resource utilization but due to task consolidation resources may still draw power while being in the idle state. Young Choon Lee et al. has introduced two algorithms to maximize the utilization
of resources of the cloud. The two algorithms are ECTC and MaxUtil. ECTC works on the premise of calculating the energy which is being used by a particular task when there are simultaneous tasks running parallel with it, and then it is compared with the optimal energy which is required. MaxUtil focuses more on the mean usage of a particular task when it is being processed. [39]

Dzmitry Kliazovich et al. presented a simulation environment for data centers to improve their utilization of resources. Apart from working on the distribution of the tasks, it also focuses on the energy used by the data center components. The simulation outcomes are obtained for various architectures of data centers. In [40] Robert Basmadjian et al. proposed the use of proper optimization policies reducing the power usage and increasing the resource utilization without sacrificing the SLAs. He developed a model which worked on incrementing the capability of the processor to process tasks. [41] Zhou Zhou et al proposes a Three Threshold Energy Saving Algorithm [TESA] which has three thresholds to divide hosts between heavy load, light load & middling load. Then based on TESA 5 VM migration policies are suggested which significantly improves energy efficiency. [42].

Dung H Plan et al. proposed GreenMonster protocol which improves renewable energy consumption while maintaining performance by dynamically moving services across IDCs. GreenMonster uses Evolutionary Multi-objective Optimization Protocol [EMOA] to make service placement and migration decisions. [43]. Liang Liu et.al. proposed a new VM architecture which has capabilities of Live Virtual Machine Migration, VM placement optimization and online VM Monitoring. This architecture gives us a considerable energy saving. [44]. Aman Kansal et al. proposes a power metering solution for virtual machines. The proposed solution has a very small runtime overhead and provides accurate and practical information for power capping to improve the energy efficiency of the datacenters. [45].

Abbas Horri et al. [46] proposed a novel approach to improve the power efficiency of system for cloud infrastructure based on the resource utilization history of virtual machines in cloud. The first work in large-scale virtualized datacenters has been proposed by Nathuji and Schwan [47]. In their proposed method, the resource management is split into local and global managers. Local manager coordinates power management methods of VMs in each host because the authors assumed that VM guests have a power aware OS. Global manager monitors the performance of multiple hosts and selects the appropriate host for requested VM migration. However, in situation that the guest OS is non-power-aware, this power
management method may be inefficient [47]. Salimi and Sharifi in [48] proposed an approach to schedule a set of VMs on a shared PM. The goal of the scheduling algorithm was to minimize the execution times (Makspan) of batch applications running on VMs based on considering the interferences of concurrent VMs. To identify the interference, they first presented an interference model in terms of number of concurrent VMs, processing utilizations of VMs and also the network latency. Nasrin Akhter & Mohamed Othman [49] surveyed and reviewed energy aware resource allocation algorithm in cloud. This paper reviews lasted proposal made for improving the energy efficiency of system. Major contribution this work is the broad study and classification of various ways to improve power consumption in cloud environment. The figure below shows the taxonomy proposed

![Energy aware allocation taxonomy](image)

**Figure 2.2 Energy aware allocation taxonomy**

### 2.5. Cost Efficient Algorithms

Li Chunlin et al. [50] proposed and const and energy aware resource provisioning algorithm for cloud. This paper presents the cost and energy aware service provisioning scheme for mobile client in mobile cloud. Proposed work proves to be cost optimal and energy efficient
as compares to simply cost aware allocation algorithms. Ehsan Ahvar et al. [51] has proposed an network aware cost optimal algorithm. This algorithm takes into consideration network performance and cost for resource allocation and selection of best server, using artificial algorithm to perform better than typical greedy heuristics. Khaled Metwally et al. [52] proposed a Mathematical modeling based on Integer Linear Programming (ILP) technique to solve optimally the resource allocation problem. However, ILP technique is knows for solving well known problem of scheduling in operating system. Author has proposed a model to use linear programming for selection of appropriate resource. Balaji Palanisamy et al. [53] proposed a cost aware allocation algorithm for MapReduce in cloud. This article presents a new MapReduce service model for cloud named Cura. Cura is cost efficient MapReduce model and cloud service to select the resource at run time for distributed problem with least cost and most efficient resource. Cura is also responsible for creation and selecting of cluster for dealing with workload. It also includes, VM-aware scheduling and online virtual machine reconfiguration, for better management and reconfiguration resources.

2.6. Behavior based algorithm

Behavior are the algorithm which are inspired from behavior of nature and behavior of animals and other living organism around us and their hierarchical evolvement over the decades. These behaviors inspires us to make decisions based on previous behaviors or the environment for making better decision that may be prediction or forecasting. Some of these algorithms proposed in the field of cloud computing are discoursed below.

Bei Wang &Jun Li [54] proposed a genetic algorithm based load balancing algorithm. In order to boost the search efficiency, the min-min and max-min algorithm are used for the population initialization. But these may stuck in local minima and to find best solution genetic algorithm is proposed. Proposed algorithm proves to provide better solution but the scheduling delay to find best solution is much higher than min-min and max-min algorithms. Keke Gai [55] proposed and cost efficient data / storage allocation algorithm using genetic algorithm for video and metadata storage over cloud. This algorithm aims to provide heterogeneous memory storage space over cloud with least cost using genetic programming to select cheapest service provider. Output proves that the proposed algorithm proves to provide improved communication costs, data move operating costs and energy performance.
Lizhen Cui [56] proposed a genetic algorithm based replica management algorithm for cloud. Author has proposed a tripartite graph based model to formulate the data replica placement problem and propose a genetic algorithm based data replica placement strategy for scientific applications to reduce data transmissions. The proposal provides better performance that random selection policy in Hadoop Distributed File System. Jasraj Meena [57] proposed a cost efficient genetic algorithm to optimize the cost for workflow schedule rather than for single tasks. The proposed algorithm proves to execute the workflow with least cost. The algorithm is been tested over popular workflow like Montage, LIGO, CyberShake, and Epigenomics.

Anjuli Garg [58] proposed a honey bee life cycle based task scheduling strategy for cloud. Author has taken into care utilization and task size to schedule the task and select the server which can execute with least execution time. Anqi Xu [59] proposed an Particle Swarm Optimization for task scheduling for cloud infrastructure to improve the Quality of Service of system. The author has taken into consideration multi objective to improve Makespan and cost. The algorithm proves to perform better than ACO and min-min algorithm. Bohrer et al. [60] proposed a most known base scheduling algorithm ACO (ant colony optimization) they proposed ant colony optimization algorithm to load balance by distributing request in a cloud computing environment. This paper proposed LBACO with dynamic load balancing strategy to distribute load among the node. The problem with traditional ACO in cloud is that it's a schedule task to most frequent (high pheromone intensity) node, if what if node is bearing heavy load in such situation may create a problem of overhead. This paper proposed and LBACO algorithm to reduce such problem. In this algorithm decrease the time of computation and monitor load on each VM with tracking previous scheduling. Xiaobo et al. [61] proposed and Real-time VM provisioning model, which is based on energy models which follow a Min-Price RT-VM Provisioning to allocate VM. Suraj, S. Rin et al. [62] proposed a genetic algorithm for task allocation in cloud environment with least execution time and maximum resource utilization.

Many other proposal made [82-86] using ACO, genetic algorithm and other learning based algorithm are been studied. Jaradat [87] proposed a Big Bang-Big Crunch optimization algorithm to solve the problem of scheduling classed for a timetable. This algorithm has proved to perform better than existing GA based algorithm.
2.7. Trust Models

Numerous trust models have been proposed in cloud. MohdaIzua Mohd Saad proposed a novel data provenance trusted model to provide secured access to data provenance via a secured communication channel [63]. This model also proposes consolidation storage with logging for virtual storage at physical layer in cloud environment. As shown in figure 1.

![Figure 2.3. The trust cloud framework.](image)

WenAn Tan proposed a trust service-oriented workflow scheduling algorithm [64]. The scheduling algorithm uses a trust metric that is combination of direct trust and recommendation trust. Proposed model also provided balancing policy to balance user requests, based on time, cost, and trust. Rizwana A.R. Shaikh proposes a trust based solution in terms of a trust model that can be used to calculate the security strength of a particular cloud service [365]. Proposed algorithm uses trust value for selecting a trusted cloud service.

[66] Xiaodong Sun introduces a trust management model based on fuzzy set theory and named TMFC including direct trust measurement and computing, connecting, and trust chain incorporating where the issue of recommended trust has been addressed to find the miss behavior of intermediate middle nodes. And this proposed model is designed for the cloud users to make decision on whether to use the services of some cloud computing providers by using trust value sets about providers and then finding trust relationships among them.

QiangGuo introduced a definition of trust in cloud systems and the properties of trust are analyzed [67]. Based on the properties of trust of a server, a trust evaluation model called ETEC is proposed. Proposed trust model includes a time based comprehensive evaluation
method for calculation of direct trust and a space evaluation method for calculating recommendation trust of server. For computing the trust in cloud, an algorithm based on the trust model is given. Experimental analysis shows that the proposed model can calculate the trust value of server effectively and reasonably in cloud computing environments. Xiaoqiong Yang also proposed A Statistical User-Behavior Trust Evaluation Algorithm Based on Cloud Model for statistic behaviors. Proposed algorithm used threshold for each type of behaviors and each user’s performance and its membership status in cloud [68]. Then the membership degree and the behavior weight will be used to calculate the user’s trust using a simple normalization function. Proposed algorithm uses the evaluated domain trust and recommendation trust, behavior trust for users’ further dynamic authorization of access control and request load balancing. Junfeng Tian proposed a Trusted Control Model of Cloud Storage with access control (TCMCS) to handle all the interactions between a client and cloud storage to ensure the secure user access and data manipulation. The proposed trust model is responsible for managing different cloud storage and manages security and integrity of user data over the cloud. Since users only need to care about their own business logic and the development of application program is greatly simplified [69]. Proposed model can be specified as shown in figure 2.

![Figure 2.4 The logical structure diagram of TCMCS.](image)

In [70] gupta has proposed a QoS Based Trust Management Model for Cloud IaaS that is suitable for trust value management for the cloud IaaS parameters. Proposed a scheduling algorithm based on trust value is done for better resources allocation and enhance the QoS provided to the users. In this paper, an approach for managing trust in Cloud IaaS is proposed.
Various other trust models are also been proposed [103 -106] to improve the system performance and reliability of reliable computing.

2.8. Conclusion

From above discussion, this can be seen that various resource allocation and load balancing algorithms fall short with problem of request scheduling using dynamic techniques to improve the performance of system. These algorithms do not consider load over the system or faults that may occur periodically over system, and also lack into consideration of previous performance history of the machines/servers.

In section 2.3, various load balancing algorithms are discoursed but these proposals consider only current load over the servers and do not search for global solution that can be based on performance of the server. These algorithms have considered only the current performance and not the physical capability of servers. Moreover, existing load balancing algorithms have assumed system as non faulty that leads to a large number of faults over the system.

In section 2.3 and 2.4 discourses about energy and cost efficient allocation and load balancing algorithms for cloud but they are not suitable for real-time systems because they do not take into consideration physical aspects of servers, deadline of requests, and considers cloud as non faulty. So proves to provide poor performance for request with tight deadline and if the system is faulty in nature i.e. the server may have high capability in terms of number of cores and RAM but may be faulty in nature, in that case existing algorithm goes under worst case performance. These algorithms also do not provide support for the reliability of the distributed systems.

Above discoursed behavior based algorithms in section 2.5 are inspired from nature which are used for resource allocation and load balancing but they only take into one parameters at a time like utilization, cost or power efficiency so cannot improve the performance of the system in all. Moreover, these algorithms do not consider the faulty nature of cloud, deadline, physical aspects of servers and previous performance of servers i.e. request failure count that had occurred or load over the serves, so cannot guarantee a high reliability and fault tolerant of the system. This may lead to large request failure counts due to overloading, deadline failure, and faults over the system which results in poor performance, and low reliability of the system.
This chapter identifies various techniques and issues and provide future directions to propose new methodologies for efficiency improvement, and fault tolerance in the cloud computing.