CHAPTER 2
REVIEW OF LITERATURE

2.1 OVERVIEW

Cloud Computing is an upgraded paradigm of distributed computing. As cloud technology is expanding day by day and facing numerous challenges, one of them being discovered is scheduling. Algorithms are crucial to schedule the jobs for processing. Job scheduling algorithms is one of the most arduous hypothetical problems in the domain of cloud paradigm. This Chapter focused on the research contributions of various authors which directly or indirectly results in performance improvement in cloud environment.

2.2 CLOUD COMPUTING AND ITS CHALLENGES

Cloud computing is setup in a very dynamic environment where users from all around the world can just plug-in and use computing resources. This dynamic nature of cloud computing raised several challenges in cloud environment. The wide spread area of cloud computing constitutes various open challenges or research concerns like Job Scheduling, Energy Efficiency, Resource Allocation , QoS, Security, VM Migration, Resource Scheduling etc. The performance of cloud datacenters is directly affected by these mentioned challenges and research concerns. Following sections represents a detailed review of research contributions of various authors in the major area of cloud computing.

2.2.1 Literature Review on Cloud Computing

Peter Mell et al. (2011) [48] had found cloud computing to be task and user centric. Authors also addressed that distributed computing can be proved to be more effective for collaborations and sharing of resources in a group. The National Institute of Standards and Technology (NIST) report further puts forward five essential characteristics of the cloud computing - broad network access, rapid elasticity, resource polling, on-demand self-services and measured service. Further the four deployment models - private, community, public and hybrid cloud are stated in their report. NIST also presented another scope of the basic services offered by the cloud computing that involve platform, software and infrastructure as a service.

David S. Linthicum (2009) [49] in his book summarized total seven service components of the cloud computing. These include integration as a service, database as a service, process as a service, storage as a service, information as service, platform as a service and application as a service. The cloud computing components constitute application semantics, information, information model, data, metadata, data dictionary, data catalog, and schema. To be able to setup the data in a cloud, organizations need to deal with three concerns beforehand. First, a well-performing architectural foundation is via cloud computing. Enterprises also need to find out the validity of the moving to cloud platform and must learn about the underlying information. Secondly, the process of moving that data had to obey the essence principle that states that data moving to services must start transferring in small and simple steps. It is the only logical suggestion for most cloud users as in most of the situations; it ensures the success of the process. Lastly, selecting a platform is very crucial for an organization to be able to control and organize their data. Author concluded that, above discussed are the reasons why an enterprise needs to fully understand the information issues beforehand so that they can make an informed decision about the platform they want. Cloud computing, as a platform, is one of the best choices available for enterprises.

Sumit Goyal (2014) [50] presented a review to help the business organization to get over their confusion and take decisions to adapt a good cloud model for their enterprises. Author summarized what model will be the best for those business enterprises and also emphasized that cloud computing is the emerging technology that every business would want to adopt for more scalability and profits. This review defined cloud computing and highlighted all the service models of cloud computing and discussed features like hybrid, private, public and community cloud computing. Technology wise, there is not much difference in the four models as they all run on same technology but one of the cons of public cloud computing is the lack of security and privacy. Whereas private cloud is more secure but costly at the same time, making it impossible for many small sized organizations to afford it. Hybrid cloud is a mix of public and private cloud and allows the organizations with an option to keep their regular data in the public cloud and sensitive data in the private cloud. A community cloud combines public and private cloud wherein some organizations get combine their resources and form their own private cloud called as community cloud'. At last cloud security issues were also discussed by the author.

Maricela et al. (2014) [26] analyzed the factors that an enterprise needs to consider while deciding about the adoption of cloud computing. The analysis was made from the companies' point of view as several companies are shifting to cloud computing just because it is the latest IT trend. But there are many companies that don't even take this into consideration as the idea of having sensitive business information away from their company premises is a big no for them. Both of these cases are about companies that are not well informed about the power of cloud computing. Cloud computing may not be the perfect solution for everybody but adopting it or not must be based on some real information and analysis. There is a need to analyze the good and bad aspects of the following factors - existing software available, setup and running cost, return on investment, existing IT infrastructure, performance and security. Also, there is a need to correlate all these factors with business area and the company size so that clients can identify whether a cloud computing solution would be suitable for their needs or not.

Amol C. Adamuthe et al. (2015) [6] had tried to forecast the use of cloud computing technologies in India using the opportunities, strengths and weaknesses as a tool. Cloud computing offers good solution that can increase productivity and cost effectiveness in the organizations to reduce time and effort. But the major limitations of cloud computing that are reported by several researchers are the interpretability, privacy, security and compatibility. Authors concluded that as many researchers are focusing on the overcoming the weaknesses and threats that arise but more efforts are required to improve the business related issues and several aspects like licensing issues, billing issues, pricing issues, service level agreement, adoption framework etc.

Rajkumar Buyya et al. (2009) [1] in their work attempted to represent certain of vision for the creation of global cloud exchange for the trading services. Authors had discussed some representative platforms that cover state-of-the-art. Such cloud technologies offers limited support for the market oriented resource management and they have to be extended to support mechanisms as well as algorithms for allocation of the VM resources to meet the service level agreements, negotiate QoS between the providers and users as well as establish SLAs and manage risks associated with the violations of the SLAs. Moreover, the interaction protocols have to be extended so that they can support interoperability between various cloud providers. Also, authors emphasized that users need programming environments and tools that can allow rapid development of cloud apps.

2.3 JOB SCHEDULING

Scheduling is the allocation of the system resources to various tasks. It is used in cloud computing to achieve high performance as well as high system throughput. The optimized utilization of resources, fast speed and high efficiency is highly influenced by the type of scheduling used for the cloud computing environment. Various criteria popularly known for scheduling are: Minimum response time, minimum turnaround time, minimum waiting time, maximum CPU utilization and maximum throughput. Response time is the time between the submission of a request and the first response to that request. Turnaround time is the amount of time taken to execute a particular request. Waiting time is the sum of time periods spent in waiting in the queue. Throughput is the number of processes that finish their execution per unit of time [32]. Cloud computing environment can incorporate large number of resources on which submitted users jobs get executed so to achieve high performance in such scenario, an excellent job scheduling algorithm is a must. It must also perform in a way to optimize the use of resources, generate profit for the cloud service provider and offer flexibility to the users.

2.3.1 Job Scheduling in Cloud Computing

Job scheduling in cloud is a nondeterministic polynomial (NP) Complete problem. In the task scheduling process, the users submit the jobs that need to be done to the cloud scheduler. The latter inquires the information service for availability of the resources and the properties. It, later, allocates the tasks to various resources depending on the task requirements. Cloud scheduler often assigns multiple tasks to multiple VMs. Good scheduling means that this task assignment to the VMs is done in a very optimum manner. Such an algorithm will also improve the turnaround time, throughput and CPU utilization. Task scheduling can happen in different ways depending on different parameters. The tasks can either be statically allocated during the compile time or dynamically allocated during runtime [33].

Several goals are important for a scheduler - minimizing energy consumption, maximizing fairness, minimizing response time and maximizing throughput. All of these need to be prioritized equally, in theory at least. But in practice, all of these cannot be prioritized. These goals often conflict one another and lead to a compromised output. Only one of these four goals can be actually prioritized; depending on user requirement and goal. The Figure 2.1 describes scheduling process in cloud computing

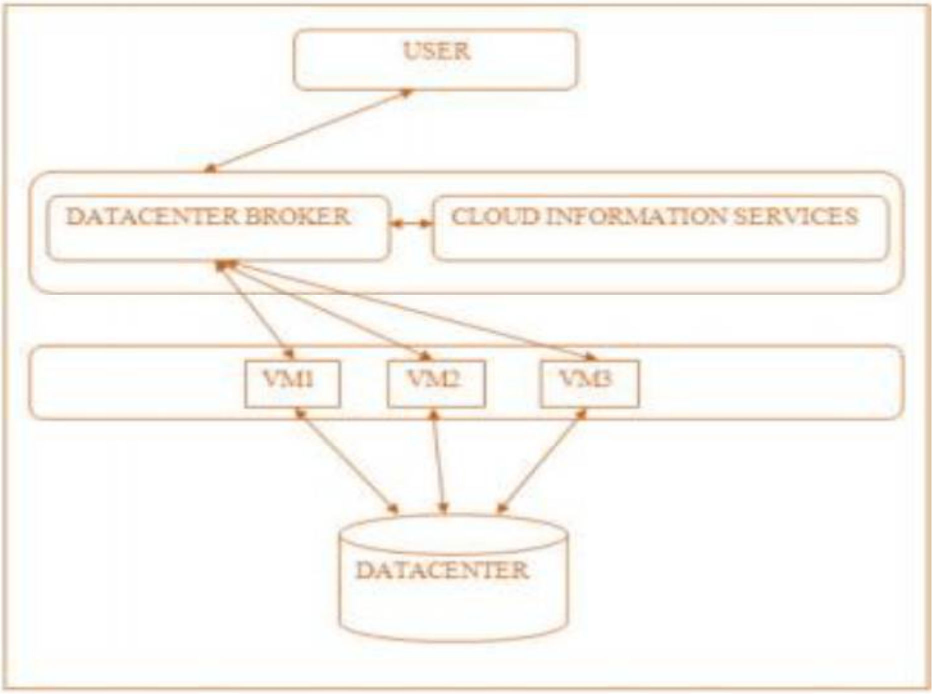

Figure 2.1: Scheduling Process in Cloud [56]

There are two main entities [54] involved in a cloud are: Cloud Service Provider and Cloud Consumer. Both have their specific motivations when they enter the cloud environment. While consumers care about performance, providers aim for effective utilization of the resources. Thus, the optimization metrics are classified into two types as Consumer Desired and Provider Desired.

(a) Consumer-Desired

Make span: Minimizing make span is the most popular and important criteria for task scheduling. Make span refers to the finish time of the last task. It is important as users always look for fastest execution for their task.

Fairness: Fairness is ideally required for every task deserves same amount of CPU time, thereby preventing any task from undergoing starvation of resources.

Waiting Time: Waiting time is the time between when the task is submitted and when it gets executed.

Turnaround Time: Turnaround time is the actual amount of time taken to finish a task after its submission.

Flow time: Flow Time is the time at which all tasks are completed. It is crucial to process all the tasks in an ascending order so as to minimize the flowchart. The reduction of the average response time of the schedule is also called flow time minimization [54].

(b) Provider-Desired

Throughput: Throughput referred to as the total number of jobs that complete their execution per unit of time.

Resource Utilization: The provider wants to keep the resources as busy as possible. Optimization has an objective and a constraint. The objective defines the best possible and available option whereas constraint defines the limitations. Hence, both are interrelated. Here are some constraints that are considered during scheduling:

Deadline: Deadline represents the time till which the task has to be finished.

Priority: Priority represents how important the task is and the urgent tasks must be dealt with first. There are various ways to determine priorities for example, on the basis of the deadline, arrival time or pre-reservation [54].

2.3.2 Literature Review on Job Scheduling In Cloud Computing

Bradley R.Swim et al. (1997) [51] in their work emphasize that scheduling plays an important role in the real-time OS that always fare time constraints. Real time systems have mission-critical applications where tasks have to be finished before the deadline. Most of the real-time systems tend to control the unpredictable environments and need OS that can handle unseen and unknown tasks. So, in such cases, not only dynamic task scheduling is needed but also adaptive software and hardware is needed to cope up with unforeseen configurations. So keeping in mind this issue of real time distributed operating system, the authors proposed job scheduler architecture which effective supports dynamic task scheduling in the real time distributed operating system (RTDOS).

Abishi Chowdhury et al. (2014) [52] described that in dynamic computing environments such as that of cloud computing, it is tough to analyze and maintain the reliability of various resources. Authors, in their work gave an attempt to propose a computing technique that can calculate the reliability of any cloud data center. It was a reliability computing technique. Further, a mechanism to continuously update the reliability of cloud resources and provide reliable scheduling of resources to the users had been proposed by the authors. A simulated environment was used to generate the allocation matrix.

Baomin Xu et al. (2011) [53] proposed a job scheduling algorithm based on the berger model. The algorithm established double fairness constraint in the job scheduling process. The first constraint helps to classify the user tasks according the quality of service (QOS) preferences and establishes a general expectation function according to the classification of the tasks so as to restrain the fairness of resources in the selection process. The second constraint helps to define the resource fairness justice function that can judge the fairness of the resource allocation. Authors had used Cloud Sim as simulation platform for the implementation of the proposed algorithm. The experimental results concur that the algorithm can effectively execute the user tasks and establish more fairness.

Saurabh Bilgaiyan et al. (2014) [54] explained that by scheduling one can manage the resources efficiently, minimize the idle time and also it increase the performance of the systems. In the modern computing environment, the amount of data that has to be processed is increasing rapidly, and the costs involved in the execution and transmission of such data is also increasing significantly. So the authors had analyzed various swarm based and evolutionary algorithms which are effectively used for scheduling of tasks on different resources in the cloud environment. Authors also highlighted that appropriate scheduling of tasks is required as it will help to manage the increasing costs of the data intensive applications. The benefit of using such methods is that large and complex problems spaces can be searched and an optimal solution can be established in lesser time.

Anjuli Garg et al. (2014) [55] described that the task scheduling is actually done to allocate the tasks to various resources in an effective and efficient manner. A good scheduling strategy can easily adapt to modifying environment. The authors designed a meta-heuristic scheduling algorithm for public cloud and named it as improved honey bees scheduling algorithm. The proposed algorithm scheduled any type of tasks and maps them to the resources in a cost effective manner. The performance metrics of proposed scheduling algorithm is measured and after comparison with other algorithms the simulation results yield that the proposed algorithm gave better performance in terms of cost.

2.3.3 Job Scheduling Approaches

Pinal Salot et al. (2013) [56] done an insight study about various existing scheduling algorithms like FCFS, Round Robin, Priority and Shortest Job First and also detailed the working , merits and demerits of each algorithm. In FCFS the jobs are processed in the first come first serve manner i.e. according to their arrival time. As there is no prioritization, it causes long waiting periods even for urgent and high-priority tasks and leads to missing deadlines. The round robin scheduling algorithm greatly improves the average response time. But, it fails when the number increases as the waiting time is highly dependent on the number of processes. But in priority scheduling, waiting and response time are different for each process. High priority processes have small waiting times whereas low priority processes have to wait for a long time in the queues and it certainly affects the performance whereas in the case of SJFS scheduling, longer processes get affected. It is in fact a special case of priority scheduling that takes waiting and processing time into account.

Sukhija et al. (2014) [57] proposed a new CPU scheduling algorithm which is closely related to MIN-MAX as it behaves both as the pre-emptive and non-pre-emptive algorithm, depending on the burst time so as to improve the CPU efficiency in a multiprogramming operating system. The proposed scheduling algorithm reduces the starvation , context switch and the waiting time issue among various processes. The proposed algorithm compared with other scheduling algorithms like FCFS, SJF, RR and in terms of throughput, response time, overhead, effect on processes, starvation and decision mode. The evaluation results show the dominance of the proposed algorithm over the other algorithms.

Jing Xiao et al. (2012) [58] proposed a scheduling algorithm based on priority. In the cloud computing, the arrival of the requests from the users is dynamic whereas the number of machines request is very random. Hence, a dynamic scheduling algorithm is needed which effectively deals with this scenario. In the proposed strategy, the authors ranked the requests according to the profits these request can bring upon. The proposed strategy when tested, the result proved that sharp rise in the benefits as proposed strategy had higher average resource performance when compared with first come first serve strategy.

Yuli Yang et al. (2013) [59] proposed a algorithm that is based on the trust. The algorithm deals with decreased the probability of failure of task assignments and further promote the trustworthiness of execution environment. The algorithm maps the tasks to the resources by judging the possibility of the resource failure. The possibility was calculated or estimated by the reliability and security constraints. Simulation results depicted that the proposed scheduling algorithm is more valuable than the traditional algorithms because of the reason, it can find the most trusted execution flow in a coherent manner and useful for scheduling application workflows.

2.3.4 Heuristic Based Job Scheduling Approaches

LipsaTripathy et al. (2014) [60] has designed a protocol that can minimize the switching time and also improve the resource utilization. It also boots up the throughput and the server performance. In the proposed approach design the authors assigned a priority to each user jobs which help in minimizing the make span by increasing the resource utilization. It can be used to minimize the overall switching time and improves the utilization which, in turn, improves the cloud computing cluster.

Medhat Tawfeek et al. (2015) [61] had defined that one of the core issues in the cloud environment is related to the task scheduling. Task scheduling in cloud computing is actually an NP-hard optimization problem. Hence, several meta-heuristic algorithms have been suggested to solve this problem. An efficient task scheduler adapts the scheduling strategy according the change in environment and the type of tasks. Authors in their work compared ant colony optimization (ACO) with the first come first serve (FCFS) and Round Robin (RR) scheduling algorithms. The results signify that ACO outperforms the latter two.

Yuan Shi et al. (2009) [62] had put forward a solution to the time-varying scheduling problems. With reference to ACO approach, authors had aimed to reduce the total cost in particular period while still meeting the deadline constraints. To make it possible, an integrated heuristic had been designed depending on the average value of the cost and deadline heuristics. The ultimate goal of the proposed algorithm is to maximize the gain of the cloud service providers in a situation, when the current resources are not sufficient to process all the requests in time and also to produce an optimal schedule which decreases the total cost of the workflow in the specific period. A time varying grid workflow which contains four topologies was built to test the algorithm, performance in various topologies within a time period is considered to calculate the fitness value. The algorithm proved to be a robust and powerful approach.

Dinesh Komarasamy et al. (2015) [63] proposed novel approach and named it as Adaptive Deadline Based Dependent Job Scheduling (A2DJS) algorithm for cloud environment which constitutes of three components job manager, data center and virtual machine (VM) creation. The mail aim of the proposed approach is to enhance the utilization of the processing speed of the VM and to minimize the make span of submitted user jobs. The proposed approach achieved specified aim in successful manner when compared with other existing algorithms.

Bagherzadeh et al. (2009) [64] had tried to introduce the concept known as biased initial ants to bring improvement to the ACO algorithm. Authors proposed approach takes up the outcomes from the deterministic algorithms for the concept of biased initial ants. The authors took the standard deviation of the jobs as well as the heuristic information, pheromone and expected execution time of a job on any given machine into consideration. The experimental results demonstrate a make span reduction of 33 and 20 percent in comparison with the Max Min and Min Min respectively. The environment for the same is consistent, and with low task and machine heterogeneity.

A. V. Karthick et al. (2014) [65] proposed a multi-queuing model that can reduce the problems caused by the existing scheduling method. The authors devised a algorithm called Efficient Multi-Queue Scheduling algorithm that works by dividing the jobs into three queues - small, medium and long. Scheduler then makes dynamic selection of process from these three queues and allocates them to resources for processing. The algorithm is able to reduce starvation by clustering various jobs on the basis of burst time. The proposed algorithm showed better performance when compared with the other algorithms in cloud environment.

Iqra Sattar et al. (2014) [66] proposed a Multi-level Queue with Priority & Time Sharing scheduling algorithm. In the proposed algorithm, a priority level for the user task is set by considering the characteristics of the process and then the task is assigned to the queue. Further, the queue got executed for a specific amount of time and then a new queue is formed for the next round. The experimental results expressed that method improved starvation but it causes CPU utilization issues.

Luqun Li et al. (2009) [67] had worked on building a non-pre-emptive M/G/1 queuing model. The model enabled task scheduling that meet the QoS requirement for the cloud users. During testing, the technique yield maximum profit for the cloud service providers.

Ali Rezaee et al. (2011) [68] proposed a multi-level queue management technique. weighted fair queue (WFQ) is combined with the fuzzy inference system to form a proposed FDWFQ i.e. Fuzzy Dynamic Weighted Fair Queue (FDWFQ). The proposed technique made use of agile knowledge base to organize the queue weights dynamically. Authors build a fuzzy inference system which accepts the feedback from the QoS measurement component and tries to set weight of each queue independently and dynamically regarding to feedbacks and QoS class-expected metrics. The proposed technique can be commonly used in QoS-aware systems such as Grid services.

Rakesh Kumar Yadav et al. (2012) [69] had put forward a multi-level feedback queue scheduling i.e. Multi-Level Feedback Queue Scheduling (MLFQ) scheduling algorithm that divides the queue into three parts and each of the user processes get executed in all the three queues for a specific amount of time. Experiment results concur that the algorithm minimized the turnaround and waiting time but it has CPU utilization related concerns as it has to wait until a queue of processes is build.

Wanqing You et al. (2014) [70] studied existing scheduling algorithms that work well in presumptive cases in a single machine as they were unable to make best decision for the future because in real world, several virtual machines and tasks would work in parallel. The authors had presented a new scheduling algorithm based on the multi-queue concept which can schedule tasks by considering parameters like capacity of the machine, task priority and history log. The algorithm, considered the history of new incoming jobs, tried to dispatch unscheduled jobs in global queue immediately to escape from a long idle time on virtual machines so as to optimize the usage of all resources and accomplish load balancing at the same time.

Raheja et al. (2014) [71] proposed a two-layered architecture of a multilevel queue scheduled based on a vague set theory that schedulers can handle the imprecise data and improve the starvation of low-priority tasks and also optimize the performance metrics and response time through MATLAB simulation. The algorithm also effectively does the dynamical allocation of CPU time between multiple queues because of its ability to approximate the user priority , input parameters and number of tasks.

2.4 ENERGY SAVING IN CLOUD COMPUTING

Seyedmehdi et al. (2014) [72] described a means for the task execution using an efficient virtual machine scheduling strategy. The lesser requirement for the migration of the VMs is also portrayed by the authors. Operational costs get easily affected by the energy consumption; hence energy efficiency becomes really important. The authors proposed a VM scheduling algorithm which was based on the unsurpassed utilization level so as to provide reduction in total energy consumption as well as meeting QOS levels. The proposed algorithm aims to balance execution speeds of virtual machines (VMs) on the host with a result that the host worked at maintaining optimal energy level also helps to meet the task deadlines, thereby validating the SLA.

Farahnakian et al. (2014) [73] also detailed a scheduling mechanism that make use of reinforcement learning technique i.e. Reinforcement Learning-based Dynamic Consolidation (RL-DC). The technique used agent to determine the most optimum power mode policy thus minimizing energy consumption. Based upon the past knowledge the agent learns to decide when a host should be switched to either to the sleep or to the active mode and further improves itself as the workload changes. The energy consumption reduces as the unused resources are turned off. So the proposed technique, RL-DC does not depend upon any prior knowledge about the workload because it dynamically adapts to the environment to achieve better energy consumption. The simulation results yields that proposed technique reduces energy consumption and maintained desired performance level.

Anton Beloglazov et al. (2010) [74] also suggest an energy efficient resource managed system. The system was for the virtualized cloud data centers that reduce the operational cost as well as ensure required QoS. Energy can be saved by continuous consolidation of the virtual machine according to the utilization of the resources, thermal state of the computing models and the virtual network topologies implemented between various VMs.

R. Buyya et al. (2012) [75] proposed an architectural framework as well as principles for an energy-efficient cloud computing environment. Based on proposed architectural framework, the author developed allocation algorithms and resource provisioning that could assign data centre resources to the client apps so as to improve the efficiency of the data centre and also delivers the required QoS.

M. Guazzone et al. (2011) [76] proposed a framework so as to manage the resources in the cloud environment in an automatic manner while reducing the energy used and minimizing the SLA violations. After the simulation authors was able to demonstrate that the proposed approach was able to dynamically adapt to workloads that vary over time and also improve the system performance as compared to traditional approaches.

Anton Beloglazov et al. (2012) [99] concluded that, application of the energy-efficient management strategies is crucial because it could improve and maximize return on investment. Such strategies involve power saving modes for idle servers and dynamic consolidation of the virtual machines. But, such consolidations can be tricky as they can lead to violation of the service level agreements (SLAs). Authors had proposed a novel adaptive heuristics that are worked on the principle of analyzing the historical data on the resource usage to output the energy and performance efficient dynamic consolidation of VMs. The authors evaluated the proposed algorithm using workload traces through extensive simulations on a big scale experiment setup. The results showed that the proposed algorithm significantly surpass other dynamic VM consolidation algorithms n terms of reducing the energy consumption with maximum adherence to the SLAs.

Ning Liu et al. (2013) [100] proposed an optimization model for the task scheduling that aimed to minimize the energy consumption in the data centers. The proposed model worked as an integer programming problem that minimized the energy consumption through scheduling of tasks to a minimum number of servers while controlling the response time constraints. The number of active servers required and average task response time to cope up with such time constraints are bounded through the use of a greedy task scheduling scheme. The proposed model focused on data centers that had heterogeneous user tasks to process. The simulation results proved that the proposed model lead to less energy consumption as compared to any non-optimized scheduled scheme.

Young Choon Lee et al. (2010) [101] presented two energy-responsive task consolidation heuristics that aimed to maximize the resource utilization at the data centers while taking in account both idle as well as active energy consumption, explicitly. The heuristics worked in this way that it assigned the task to a resource that would have the minimum energy consumption without degrading the performance. Based on the results, the heuristics showed a very promising energy-saving ability.

Gong Chen et al. (2008) [102] believed that energy consumption is a pressing issue that hinders the scaling up of the hosting internet services. Through the use of dynamic server provisioning, it was possible to turn off the unnecessary servers that were not in use, to be able to save energy. So the authors worked on designing server provisioning as well as load balancing algorithms. Authors studied the interactions between these and the results show that algorithm was able to save significant energy without compromising the user experience.

Shekhar Srikantaiah et al. (2008) [103] studied the issue of request scheduling for the multi-tier web apps in a virtualized heterogeneous system and analyzed how energy usage, performance and resource operation changes as multiple workloads with changing resource usages are clubbed on the common servers. The aim was to minimize the energy consumption without compromising with the performance requirements. An effective heuristic for the multidimensional packing problem was proposed by the authors to perform consolidation in a simplified scenario. The authors address the complexities involved in performing consolidation for power optimization. There are many issues that affect consolidation, including server and workload behavior, security restrictions requiring co-location of certain application components, and power line redundancy restrictions. As the proposed approach is independent of the workload type so it effectively deals with these issues and proved to be energy efficient consolidation approach.

Jinhai Wang et al. (2013) [104] came to a conclusion that allocation of CPU as well as memory plays a significant role in determining the energy efficiency of a system. Thus, CPU and memory are quite dominant factors when the performance of the system and energy consumption is concerned. The authors focused on the VM placement, thereby suggesting a heuristic greedy algorithm that worked on VM deployment as well as live migration, so as to minimize energy consumption and maximize resource utilization. The algorithm is based on the quadratic exponential smoothing method for predicting the workload. The algorithm, however, ensures that CPU intensive as well as memory intensive services that are mapped to the same physical server become more complementary. Experiment results proven that a significant improvement in energy-saving, scalability, and workload balancing is achieved by the algorithm when compared the approaches that are based on CPU utilization aiming at a single objective.

Huaimin Wang et al. (2010) [105] belived that the energy consumption in the large scale data centers is the major problem in achieving green computing and decreasing the cost. So authors proposed a self-reconfiguration approach to allocate the virtual machines in large scale data centers and named it as GABA. The proposed GABA algorithm worked in an adaptive manner to reconfigure the virtual machines in such data centers that contain heterogeneous nodes and are virtualized. GABA algorithm which is based on the online-reconfiguration approach on the virtual machines that has a web server running on them. The algorithm can efficiently predict the number of VMs required for each application. It can also decide the locations of the VMs by considering the environmental conditions as well as time variation requirements. The biggest benefit was evident, that the algorithm allows self-configuration of the data centers without requiring any explicit specifications. Therefore, it is possible to search the optimal solutions online in a very efficient manner through the use of the complex configuration spaces.

Dzmitry Kliazovich et al. (2013) [106] proposed scheduling algorithm that worked by delaying the congestion related packet loss and named it as e-STAB. The algorithm aims is to optimize the energy consumption of the equipment for the data centre as well as enabled load efficient allocation of the network traffic. e-STAB upgrade quality of service of running cloud applications by minimizing the delays relevant to communication and congestion associated packet losses. The authors used a green cloud simulator to test the results of the algorithm; the algorithm proved that the energy consumption does no t increase.

Vijindra R. et al. (2012) [107] proposed an energy efficient scheduling framework for the problem of job scheduling for cloud environment. The framework is built on three objectives firstly minimizing the completion time of each job in the system secondly minimizing the energy consumption in the data centers and lastly balancing the incoming load. The authors compared the results of this new scheduling framework which is supposed to be energy efficient with the previously implemented job scheduling algorithms in the cloud computing. The simulation results was the proof that showed the proposed framework is able to achieve much better results than the existing algorithm in terms of energy efficiency.

Jiandun Li et al. (2011) [108] proposed a hybrid scheduling approach for the private clouds that was energy efficient and also explored various characteristics about the VM workflow scheduling. The approach is based on the least-load-first algorithm and pre-power technique. The approach saved a lot of time of the users, improve energy consumption and achieve better load balancing. The drawback associated to the approach is that it is mainly applicable on the private clouds not on public clouds.

Fei Cao et al. (2013) [109] describe that the cloud infrastructure has enabled efficient execution of several scientific applications through proper resource sharing as well as customized configuration with proper flexibility. Such applications are modeled as DAG (Directed Acyclic Graph) structured workflows however the energy cost is quite high. To make cloud computing technology more sustainable even with the massive increase in the size of big data as well as complexity of operations, the authors had proposed a energy-efficient scheduling algorithm that minimizes the energy consumption. The proposed energy-efficient algorithm uses DVFS (Dynamic Voltage and Frequency Scheduling) technology to minimize the energy consumption while maintaining the performance. The algorithm also minimized the VM overhead which further reduces the energy consumption and also improves resource utilization rate.

Inigo Goiri et al. (2010) [110] to enable power aware allocation of the resources, the authors had presented a dynamic job scheduling policy. The policy tried to combine the workloads from several separate machines into a small number of nodes, while ensuring to fulfill the number of hardware resources required to maintain the quality of service which makes it possible to turn off the servers that are not in use at the moment resulting in reduction of energy consumption. The policy also considered all the virtualization overheads while making decisions and many other important parameters such as the reliability of the system and dynamic enforcement of service level agreements while reducing the power consumption. The evaluation of the proposed policy had been done by comparing it to other existing coming policies in a simulated environment. As the experiment was performed using real workloads, it proved that the proposed policy enabled substantial improvement in the performance and energy efficiency in the given scenarios.

Young Choon Lee et al. (2009) [111] address the issue of task scheduling on an Heterogeneous computing system (HCS). Authors proposed a ECS (energy-conscious scheduling) heuristic that considers both make span and energy consumption. The scheduling heuristic incorporates dynamic voltage scaling (DVS) that helps in reducing the energy consumption. An objective function is used in the scheduling phase of this algorithm that ensures optimum trade-off balancing of the two performance considerations. Also, the energy reduction phase uses the MCER (make span conservative energy reduction) technique. Authors performed a comparative evaluation that clearly showed that ECS outperforms the previously common scheduling algorithms with a noticeable margin for energy consumption.

Anirban Basu et al. (2016) [112] elaborated that network bandwidth is utilized to connect users in the cloud environment, the network transmission and switching components present in the cloud also consumes good percentage of overall energy.

Keeping in mind this issue an optimized energy efficient task provisioning which considered memory as well as CPU resources for cloud data centers using both meta heuristic PSO (Particle Swarm Optimization) and Bat algorithms was proposed by the authors. The approach considered the advantages of both of the previously mentioned algorithms and then formed a scheduling strategy to optimized energy efficient migration that was able to determine the total number of tasks that can be assigned to each virtual machine with least amount of energy consumed. The migration was done in an energy-efficient manner by utilization of the bandwidth among the virtual machines to minimized energy consumption.

Qing Zhao et al. (2016) [113] proposed a deadline and energy aware task scheduling algorithm for the data-intensive applications. The objective was to develop a scheduling framework with maximum energy efficiency and minimum service level agreement violations; satisfying the technological requirements of the cloud computing. Both, datasets as well as the cloud system, was built in a tree-structure model by the algorithm through correlation of data based on clustering. Therefore, the amount of movement of global data can be reduced to a great extent, which further decreased the service level agreement violations and improved the energy efficiency of the network devices and servers in the cloud.

Youwei Ding et al. (2015) [114] emphasized that the current energy efficient scheduling algorithms working on virtual machines in the cloud are not effective if the physical machines are heterogeneous, and if their sum of power is considered. Authors in their work puts forward a scheduling algorithm with great energy efficiency and called it as EEVS. The process of this algorithm was divided into schedule periods. In each schedule period that was equivalent in nature; virtual machines are allocated to physical machines. Each of the active cores operates on an optimal frequency. The cloud was reconfigured after each period to combine the computing resources that further helps in reducing the energy consumption. The deadline constraint was considered during the scheduling and was satisfied. The simulation results further proved the effectiveness of this proposed scheduling algorithm in reducing energy consumption.

Knauth et al. (2012) [115] presented an energy aware virtual machine scheduler for the cloud infrastructure, as the energy efficient computing and cloud computing are two fastest growing IT trends at the moment. The authors named it as Opt Sched and addressed that the effectiveness of scheduling algorithm in reducing the energy consumption is dependent on how many minimum number of servers it utilize to service a given workload. The proposed Opt Sched makes use of the reservation length of the timed instances so as to optimize the virtual to physical machine mapping. The objective function considered by the authors was cumulative machine uptime (CMU). CMU holds the total up time of all machines present in the data center. The target was to reduce the cumulative machine uptime because by reducing CMU it directly reduce the data center energy consumption. Experiment result concluded that Opt Sched reduced the machine uptime by up to 60.1% in comparison with the round robin scheduling algorithms; 16.7% in comparison to the first fit scheduling algorithm.

Yu Li et al. (2009) [116] developed a novel energy aware algorithm so as to schedule the tasks on heterogeneous clusters and named it as EAMM. The approach used by the authors laid its foundation on the general adaptive scheduling heuristics Min-Min algorithm. The capability of proposed algorithm EAMM in terms of energy saving was examined by simulation experiments and contrasted with traditional Min-Min algorithm. The results concluded that proposed EAMM significantly reduce energy consumption of large heterogeneous cluster systems with only a marginal breakdown in the make span performance. EAMM showed remarkable scalability under various scales of the cluster systems. The said algorithm surpasses the traditional used Min-Min algorithm in terms of energy efficiency.

Gregor Von Laszewski et al. (2009) [117] developed a scheduling algorithm for energy efficient scheduling in the virtual machines in the hosts that are a part of virtualized clusters. The proposed algorithm not only prevented degrading of the virtual machine performance beyond the unacceptable levels but also dynamically scaled the operating frequencies and voltages of the compute nodes presented in the cluster. When implemented the results of proposed algorithm showed better energy efficiency.

Rodrigo N. Calheiros et al. (2014) [118] targeted the issue of energy efficient execution of the CPU intensive Bag-of-Tasks (BoT) and urgent tasks in cloud environment. This is applicable in domains like healthcare and disaster management.

Authors had proposed a cloud aware scheduling algorithm that uses DVFS so as to ensure meeting the deadlines for the urgent CPU intensive Bag-of-Tasks jobs with minimized energy expenditure. To analyze the energy efficient scheduling of BoT applications in clouds, the BoT workload was submitted to the cloud to measure the effectiveness of proposed algorithm. As the user jobs were divided into two urgency classes recognized as low urgency and high urgency jobs so jobs was assigned to each class uniformly according to a defined share and a deadline is associated with each submitted jobs. While proposed approach focuses on execution deadlines, other approaches focus on the other programming models. The proposed approach significantly reduced the energy consumption while not compromising with Quality of Service that is offered to users.

2.5 SUMMARY

The literature review and contribution of work focusing the field of job scheduling in cloud environment presented in this chapter. The Chapter discusses the various analysis and research made in the field of energy efficient heuristic job scheduling approaches in cloud computing. The Chapter discusses the work done by several researchers in the relevant chosen field, which gives direction in the proposing the optimize energy efficient heuristic job scheduling algorithm in the cloud environment. Chapter 3 presents in detail the various concepts related to job scheduling and its algorithms.