CHAPTER 3

THEORITICAL ANALYSIS OF CLOUD OPTIMIZATION

3.1 INTRODUCTION

Scheduling process, task and data processing workflows on the cloud is a very complex and more challenging task. Generally, scheduling is meant to keep the resources busy; an idle resource is considered to be wasted while concerning about the time and other performance related criteria. Scheduling is the act of selecting the next process for resource to “service” once the current process leaves the resources idle.

3.2 GRID BASED SCHEDULING ALGORITHM

3.2.1 Evaluating Scheduling Algorithms on Distributed Computational Grids

Computational grids are becoming more prevalent as the cost of bringing together disparate computing resources declines. However, a number of challenges remain before these grids can be utilized efficiently. Several well-known scheduling algorithms are explored to schedule work on a grid under probabilistic work, arrival rates and varying task completion times. A method using Estimated Time to Compute (ETC) matrices was presented to model heterogeneous systems; it is found that the Minimize Completion Time (MCT) scheduling algorithm (Ryan J. Winsnesly 2006) attempts to minimize the total computational time, required for any job performed for the best out of a set of well-known scheduling algorithms.
To analyze MCT further, the concept of ETC perturbation was presented, and found that the MCT algorithm was quite stable against variations in job completion times. However, the most interesting result of the simulations was not the data itself, but the intriguing possibility that ETC values might be useful as inputs to robust schedulers. That is, it could be the case that ETC matrices, even when they are only rough estimates of underlying job completion times, and could be treated as true estimates without affecting robustness.

Cloud computing is emerging as the new paradigm for the next-generation distributed computing. It has attracted a lot of attention in both academia and industry. The rapid growth of visualization technologies and cloud computing has opened a new paradigm for utilizing the existing resource pools for on-demand and scalable computing. Cloud computing offers a new computing model where resources such as networking infrastructures, computing capability, storage and software’s can be shared as ‘services’ over the internet in the form of virtualized resources, which reduces the costs while increasing processing throughput and decreasing processing time for the business services. In future, the number of cloud provisioning offers is currently growing and so is the number of possibilities for running an application in the clouds, there will be many services available from the service clouds.

3.2.2 Self-adaptive Task Scheduling System for Non-dedicated Heterogeneous Computing

In order to provide high performance computation power to serve the increasing need of large applications, people strive to improve a single machine’s capacity or construct a distributed system composed of a scalable set of machines. Compared to the former, where the improvement is mainly up to the hardware technology development, the construction of distributed systems for resource collaboration is more complex. Some of the well-known
existing distributed systems composed of heterogeneous resources are Condor, NetSolve, Nimrod, Globus, and the grid computation environment. These systems, especially the Grid, have unprecedented computing power. However, delivering this unprecedented computing power to the users is still an elusive problem. One of the major issues is, scheduling large applications in these non-dedicated distributed systems. Many heuristic scheduling algorithms and systems are proposed by Ming Wu et al (2003), to address this problem. The efforts to construct a national scale grid computing environment have brought unprecedented computing capacity and complicacy. Exploiting this complex infrastructure requires efficient middleware to support the execution of distributed applications, which presents the challenge on how to schedule tasks in shared heterogeneous systems.

![Figure 3.1 Framework for GHS Scheduling](image)

Figure 3.1 Framework for GHS Scheduling

Most existing scheduling systems are based on pre-determined estimation of task completion time and resources availability. They may not provide appropriate scheduling if the underlying computing resources present an abnormal usage pattern during an application execution. For solving long-
running applications in a large-scale grid environment, abnormal usage of some resources may not be uncommon.

In this approach, a novel dynamic self-adaptive scheduling algorithm and its implementation under Grid Harvest Service (GHS) is proposed. Scheduling and re-scheduling algorithms and mechanisms are carefully investigated. To reduce performance loss caused by possible machine "mutation", a self-adaptive task scheduling algorithm is developed and three performance metrics are proposed. The results show that the proposed scheduling system provides an appropriate general-purpose scheduling mechanism for long term applications. The dynamic and self-adaptive scheduling algorithm adequately captures the dynamic nature of distributed computing and therefore provides a robust scheduling by reallocating tasks in the presence of resource abnormalities.

3.3 TASK SCHEDULING ALGORITHMS

Cloud computing is a new promising paradigm in distributed and parallel computing. It can offer utility-oriented IT services to users based on the pay-as-you-go model. It can also make good use of economies for scale and dynamically deliver/configure almost any IT related services on demand. Moreover, it can conserve more energy, which is the desirable computing platform for the coming low-carbon economy. Energy conservation in data centers can be done on both processor level and operation system level. This problem is addressed by scheduling and makes good use of energy conservation techniques on OS level. The trade off problem is also been addressed between energy and load balancing, which is also a power-performance trade off. Hybrid energy-efficient scheduling algorithm using dynamic migration is used, which not only reduce the response time, but also conserve more energy and achieve higher level of load balancing.
3.3.1 Task Scheduling Optimization for Cloud Computing Systems

The advancement of cloud technologies, there is a new need for tools to study and analyse the benefits of technology and how best to apply the technology to large-scaled applications. Task management (Sandeep Tayal 2011) is the key role in cloud computing systems. Task scheduling problems are premier which relate to the efficiency of the whole cloud computing facilities. Task scheduling algorithm is an NP (Non-Polynomial) - completeness problem which play a key role in cloud computing. In Hadoop, the open-source implementations of map reduce many scheduling policies such as FIFO scheduling which is used by the master node to distribute waiting tasks to computing nodes. So, it can still achieve more improvement in the task scheduling process. Hence, an optimized algorithm based on the Fuzzy-Genetic Algorithm optimization is used, which makes the scheduling decision by evaluating the entire group of task in job queue.

![Figure 3.2 Task Scheduling Model](image-url)
Efficient task scheduling mechanism can meet user’s requirements, and improve the resource utilization, thereby enhancing the overall performance of the cloud computing environment. But the task scheduling in grid computing is often about the static task requirements, and the resources utilization rate is also low. According to the new features of cloud computing, such as flexibility, virtualization etc, two level task scheduling mechanism based on load balancing in cloud computing is used. This task scheduling mechanism is not only able to meet user's requirements, but also can achieve high resource utilization.

3.3.2 Dynamic Task Scheduling Using Fuzzy Logic in Distributed Memory Systems

The scheduling and mapping of the precedence constrained task graphs of parallel programs to processors is considered as one of the most crucial NP-complete problems in parallel and distributed computing systems. A new fuzzy logic model for solving the task scheduling problem in distributed memory systems in a dynamic environment is introduced by Rasha M. Zohier et al (2010). The main objective of this technique is to improve the fuzzy decision which is used in task scheduling on a network of processing elements by introducing new input parameters to an existing fuzzy model and at the same time to improve the load balance on the network in a dynamic environment. This modified fuzzy logic model leads to more precise fuzzy decisions, even while dealing with larger number of processors and/or larger number of tasks while increasing the number of involved parameters in the fuzzy model, related to an existing one.

The usefulness of adding extra input parameters in the fuzzy system may be investigated, even more the relationship between the different input parameters and the effect of each parameter in the fuzzy system may be investigated as well. The proposed fuzzy model is capable of processing
inputs from the fly data that arises from the current state of the processors. According to the proposed model, tasks are generated randomly and are served based on the First-Come-First-Serve rule. When the task is ready to be assigned, its information is passed to the processors for bidding. Each processor has a local scheduler for managing its own activities, which supplies information on its current state and follows whatever decision is given where the fuzzy logic mechanism which is used in making decision on the task assignment. The comparative results show that the modified fuzzy model outperforms the existed one.

### 3.3.3 Many-Tasks Computing for Cloud Services

Many-Task Computing (MTC) paradigm embraces different types of high-performance applications involving many different tasks, and requiring large number of computational resources over short period of time. These tasks can be of very different nature, with sizes from small to large, loosely coupled or tightly coupled, or compute-intensive or data-intensive. To analyze the viability, from the point of view of scalability, performance, and cost of deploying large virtual cluster infrastructures over distributed different cloud providers for solving loosely coupled MTC applications.

MTC users, employ loosely coupled applications comprising many tasks to achieve their scientific goals. MTC applies to not only traditional HTC Environments such as clusters; MTC denotes high-performance computations comprising multiple distinct activities, coupled via file system operations. Tasks may be small or large, uniprocessor or multiprocessor, compute intensive or data-intensive. The set of tasks may be static or dynamic, homogeneous or heterogeneous, loosely coupled or tightly coupled. The aggregate number of tasks, quantity of computing, and volumes of data may be extremely large.
Figure 3.3  Architecture diagram of MTC Application

Figure 3.3 explains the architecture diagram of MTC Model and the new term MTC draws attention to the many computations that are heterogeneous but not “happily” parallel and Grids, assuming appropriate support in the middleware, but also supercomputers. Emerging pet scale computing systems, such as IBM’s Blue Gene/P, incorporate high-speed, low-latency interconnects and other features designed to support tightly coupled parallel computations. Most of the applications run on these computers have a SMPD structure, and are commonly implemented by using MPI to achieve the needed inter process communication (Ioan Raichu et al 2008) MTC is an amiable paradigm for supercomputers.

As the computing and storage scale increases, the set of problems that must be overwhelmed to make MTC practically (ensuring good efficiency and utilization at large-scale) exacerbate. The challenges include local resource manager scalability and granularity, efficient utilization of the raw hardware, shared file system contention and scalability, reliability at
scale, application scalability, and understanding the limitations of the HPC systems, in order to identify promising and scientifically valuable MTC applications.

### 3.3.4 Artificial Neural Networks Based Task Scheduling for Cloud Computing

In the industry, task scheduling is considered as one of the most famous combinational optimization problems. The main goal is to determine a proper sequence where tasks are executed while obeying to some (transaction logic) constraints. Implementations are labelled either as centralized or decentralized, static or dynamic, or a hybrid, all with their own strengths and limitations. In general, dynamic load-balancing mechanisms provide better performance than static, but do encounter higher overhead due to the systems information that have to be updated on-the-fly (redistribution of tasks, due to overloaded CPU’s may occur as well). With a static setup, task scheduling is not affected by the state of the system; it is based on a-prior knowledge.

With the proliferation of cloud computing, new and advanced methodologies are required to assess the performance, allocation, as well as capacity behaviour in the cloud with high fidelity. Hence ANN based task scheduling architecture is proposed to improve the scheduling behaviour, better utilize (balance) the available resources, lower aggregate task execution time and hence, minimize cost. The Cloud Demand Model acts as a control entity in the scheduler architecture, a fuzzy logic based implementation of that module may provide a finer granularity on the resource availability that could be used to further improve/optimize the resource (task) allocation/balancing behaviour.

The rapid growth of data and computational needs in distributed systems and cloud computing are gaining more and more attention. Clouds
are playing an important and growing role in today’s networks. The huge amount of computations in a Cloud can fulfil in a specific time but it can’t be done by the best super computers. However, Cloud performance can still be improved by making sure all the VM available in the Cloud are utilized by a good load balancing algorithm. The purposes of such algorithms are to make sure that, all nodes are equally involved in computations.

Figure 3.4 Cloud Task Scheduling Architecture
Swarm intelligence is proposed, which uses load balancing algorithm and based on Particle Swarm Optimization (PSO). This algorithm lead the cross operation, variation operation and select operation of the GA to the PSO algorithm, it effectively overcomes the inherent flaw of getting local optimal value by PSO algorithm and finds the global optimum value in the search space again. The performance of the algorithms will be evaluated using several performance criteria (e.g. make span and load balancing level). The use of PSO in cloud can yield better performance results in many scenarios than genetic approach.

3.4 JOB SCHEDULING

3.4.1 Job Scheduling Algorithm in Cloud Environment

Cloud computing can be also used for dispatching user tasks or jobs to the available system resource like storage and software. In cloud computing, scheduling plays major role for dispatching user tasks and hence it reflects as a new pattern of business computing. The basic mechanism of Berger model in cloud computing is to dispatch the computing tasks to resource pooling which is constituted by massive computers. It enables a variety of applications to gain computing power, storage and a variety of software services according to their needs.

The predecessors has implemented the algorithms of job scheduling based on Berger Model in cloud computing in order to be able to map the theory of distributive justice in Berger Model (Baomin Xu et al 2011) to resource allocation model in cloud computing. It is needed to carry on the task classification, fairness function definition of user tasks, the task and resource parameterization, the task, resource mapping, etc. Based on the idea of Berger model, two-fairness constraints of job scheduling are established in cloud computing. In this, the user tasks are classified based on Quality of Service
parameters like bandwidth, memory, CPU utilization and size. The classified tasks are given to fuzzifier, neural network and finally de-fuzzifier. The exemplar input is matched with the exemplar output label by adjusting weights in neural network. The result obtained reduces the total turnaround time and also increases the performance.

3.4.2 Job Execution Methods and Dynamic Makespan Optimized Scheduling

The research on Quality of Service (QoS) in grid computing has been strengthened with the development of virtualization technologies, which bring several advantages into the area of grid computing, especially to grid QoS provision. Many studies are focused on merging virtualization technology into the grid infrastructures. The virtualization technology is helpful to support grid QoS. A manual configuration method and an automatic configuration method are designed to build the virtual workspace environment based on virtualization technology.

Both the Meta Services and the automatic configuration software are employed to establish the virtual workspace environment. For dynamic jobs, the corresponding Dynamic Makespan Optimized (DMO) scheduling algorithms, (Zhili Cheng et al 2008) are developed. Results of the simulation experiments demonstrate that, compared to models without virtualization support, the job execution methods and DMO scheduling algorithms can achieve higher throughput, optimize job makespan, and improve the QoS of grid system.

3.4.3 Job Scheduling with Deadlines

The cloud computing environment is highly dynamic; the system load and the computing resource utilization exhibit a rapidly changing characteristic over time. Therefore, the cloud service provider normally over-
provision the computing resources to accommodate the peak load and the computing resources are typically left underutilized at nonpeak times. Often significant underutilized cloud resources leads to the potential of exploiting surplus resources for secondary jobs in cloud systems. The highly dynamic nature of the cloud environment leads to a time-varying resource utilization and the cloud provider can potentially accommodate secondary jobs with the remaining resource.

For better implementation of the idea of resource reutilization in the cloud environment, the problem of secondary job scheduling with deadlines under time-varying resource capacity is considered. A transformation is proposed to reduce the offline problem with time-varying processor capacity to that with constant capacity. For online scheduling of under loaded system, it is shown that the Earliest Deadline First (EDF) scheduling algorithm (Shiyao Chen et al 2011) achieves competitive ratio 1. For the overloaded system, an online scheduling algorithm V-Dover is proposed with asymptotically optimized competitive ratio when a certain admissibility condition holds. It is further shown that, in the absence of the admissibility condition, no online scheduling algorithm exists, with a positive competitive ratio.

3.5 RESOURCESCHEDULING

3.5.1 Resource Scheduling Algorithm based on Multi-Dimension QoS

Due to the shortcomings of cloud simulation task scheduling algorithm, cloud simulation scheduling algorithm based on Multi-Dimension QoS (Wuqi Gao et al 2012) was put forward. In this, analytic hierarchy process in economic field was introduced into the resource scheduling algorithm to compute every dimensional parameters weight, then the tasks was allocated to appropriate resource, according to customer satisfaction, QoS
distance and loading equilibrium. This cloud simulation scheduling algorithm not only meets customer needs for multi-dimension QoS, but shortens the simulation finishing time and greatly improves simulation resource utility rate.

3.5.2 Cloud Resource Scheduling Based on Parallel Genetic Algorithm

The main characteristics of virtualization technologies employed in Cloud atmosphere is the consolidation and proficient management of the resources. Managing and scheduling the resources is the most important

![Figure 3.5 Typical IaaS cloud component with an external cloud](image)
practice for Clouds such as Infrastructure as a Service Cloud. An optimized scheduling algorithm is proposed that concentrates on the efficient utilization of the resources for the Cloud scheduling problems. A Parallel Genetic Algorithm is used for scheduling the resources dynamically. The proposed work focuses on the analysis of the performance of the cloud resource scheduling in an efficient approach. The investigation shows the scheduling procedure that improves the utilization rate of the system resources and also the pace of allotment of the resource.

3.6 SCHEDULING OPTIMIZATION

3.6.1 Cloud Service Workflow Scheduling Optimization Schema Using Heuristic Generic Algorithm

Cloud computing has opened a new paradigm for utilizing the existing resource pools for on-demand and scalable computing, which enables the workflow management system to meet Quality-of-Service (QoS) requirements of the applications (Zhang Jun Wu et al 2010). It becomes crucial for cloud customers to choose the best cloud services in order to minimize the running costs, and how to match and select the optimum cloud service will be a challenger issue. An efficient cloud services workflow scheduling and optimization schema using heuristic generic algorithm was presented, which focus on the hierarchical cloud service workflow scheduling, cloud workflow tasks parallel split, syntax and semantic based cloud workflow tasks matching algorithm, and multiple QoS constraints based cloud workflow scheduling and optimization.

The resource service scheduling technology of the cloud simulation platform is different from other common computing and it possesses the following characteristics:
- Multi-machine collaboration which means the cloud simulation virtual machines is distributed in geological places, but they belong to different owners logistically.

- Dynamic isomerism which means that net virtual nodes can be added dynamically into and exited from cloud simulation system.

- Different QoS goals.

The function of a cloud workflow system and its role in a cloud computing environment is to facilitate the automation of user submitted workflow applications where the tasks have precedence relationships defined by graph-based modelling tools such as DAG (Directed Acyclic Graph) and Petri Nets, or language-based modelling tools such as XPDL (XML Process Definition Language). A cloud workflow system is a type of platform service which facilitates the automation of distributed applications based on the novel cloud infrastructure. Compared with grid environment, data transfer is a big overhead for cloud workflows due to the market-oriented business model in the cloud environments. A Revised Discrete Particle Swarm Optimization (RDPSO) is proposed to schedule applications among cloud services that takes both data transmission cost and computation cost into account.

A RDPSO algorithm optimizes the schedules of workflow application in cloud computing environment. In this algorithm, the candidate solution is presented by the set of task-service pairs, on which each particle not only learns from different exemplars, but also learns the other feasible pairs for different dimensions. The constructive position building procedure guarantees, that each position is feasible. This scheme greatly reduces the search space and enhances the algorithm performance. Based on the
simulation results, the new algorithm yields outstanding performance on scheduling workflow applications in cloud environment.

Experiments are conducted with a set of workflow applications by varying their data communication costs and computational costs according to a cloud price model. Comparison is made for make span and cost optimization ratio and the cost savings with RDPSO, the standard PSO and BRS (Best Resource Selection) algorithm. Experimental results show that the proposed RDPSO algorithm, can achieve much more cost savings and better performance on make span and cost optimization.

3.6.2 Data Processing Flow Based Scheduling Optimization

Scheduling data processing workflows (dataflows) on the cloud is a very complex and challenging task. It is essentially an optimization problem, very similar to query optimization that is characteristically different from traditional problems in two aspects: its space of alternative schedules is very rich, due to various optimization opportunities that cloud computing offers; its optimization criterion is at least two-dimensional, with monetary cost of using the cloud being at least as important as query completion time.

The scheduling of data flow that involves arbitrary data processing consists of three different problems and are is given below,

- minimize completion time in a given fixed budget,
- minimize monetary cost for a given deadline, and
- find trade-offs between completion time and monetary cost without any apriori constraints.

These problems are formulated and an approximate optimization framework is presented to address them, which uses resource elasticity in the
cloud. To investigate the effectiveness, a devised framework was incorporated into a prototype system, for dataflow evaluation and instantiate it with several greedy, probabilistic, and exhaustive search algorithms. Several experiments were conducted with the prototype elastic optimizer on numerous scientific and synthetic dataflow, and identified several interesting general characteristics of space for alternative schedules as well as the advantages and disadvantages of various search algorithms. The overall result is quite promising and in term indicates the effectiveness of this approach.

3.6.3 A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications

Cloud Computing model compute nodes and storage reside at service provider’s side. Physical machines hosting virtual nodes at data centers can be dynamically provisioned as and when required. It is scalable to large number of nodes as user requires. Minimizing the total cost using optimized heuristics can be done by scheduling tasks onto a efficiently selected resources, transfer of data from or to resources according to price and mapping the heuristics to execute the application workflow.

A heuristic based scheduling algorithm on Particle Swarm Optimization (PSO) was presented (Suraj Ponday et al 2010). This heuristic approach is used to minimize the total cost of execution on cloud computing environment which examines the impact of change in cost of resources with varied total data size. When the result is compared with quickest resource selection algorithm, the cost of resources was saved three times in the form of load balance, quicker solution and generic solution.
3.7 ONLINE SCHEDULING

3.7.1 On-line Scheduling of Real-time Services for Cloud Computing

Cloud computing has the potential to dramatically change the landscape of the current IT industry. While there exists different interpretations and views on cloud computing, it is less disputable that being able to effectively exploit the computing resources in clouds to provide computing service at different quality levels is essential for the success of cloud computing. For real-time applications and services, the timeliness is a major criterion in judging the quality of service. Due to the nature of real-time applications over the Internet, the timeliness here refers to more than the deadline guarantee as that for the hard real-time systems.

In this regard, an important performance metric for cloud computing can thus be the sum of certain value or utility that is accrued by processing all real-time service requests. To improve the performance of cloud computing, one approach is to employ the traditional Utility Accrual (UA) approach (Shuo Liu et al 2011) for real-time cloud computing services. The real-time tasks are scheduled non-pre-emptively with the objective to maximize the total utility.

The most unique characteristic of our approach is that, compared to the traditional utility accrual approach that works under one single time utility function (TUF), we have two different TUFs—a profit TUF and a penalty TUF—associated with each task at the same time, to model the real-time applications for cloud computing that need not only to reward the early completions but also to penalize the abortions or deadline misses of real-time tasks. The experimental results show that the proposed algorithm can significantly outperform the traditional scheduling algorithms such as the
Earliest Deadline First (EDF), the traditional utility accrual scheduling algorithm and an early scheduling approach based on the similar model.

Cloud computing is internet based computing, whereby shared resources, software and information are provided to computers and other devices on demand, like a public utility. Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. This technology allows consumers and businesses to use application without installation and access their personal files at any computer with internet access. Various problems, issues and types of scheduling algorithms for cloud workflows as well as on designing new workflow algorithms for cloud Workflow management system are evaluated. The proposed algorithms are implemented on real time cloud. The algorithms are compared with each other on the basis of parameters like total execution time, execution time for algorithm, and estimated execution time.

3.8 GENETIC ALGORITHM

3.8.1 A Genetic Algorithm Scheduling Approach for Virtual Machine Resources

In the present cloud computing environment, the scheduling approaches for Virtual Machine (VM) resources only focus on the current state of the entire system. Most often they fail to consider the system variation and historical behavioural data which causes system load imbalance. To present a better approach for solving the problem of VM resource scheduling in a cloud computing environment, a genetic algorithm based VM resource scheduling strategy was demonstrated that focuses on system load balancing.

The genetic algorithm approach computes the impact in advance, that it will have on the system after the new VM resource is deployed in the system, by utilizing historical data and current state of the system. It then
picks up the solution, which will have the least effect on the system. By doing this it ensures the better load balancing and reduces the number of dynamic VM migrations. This approach solves the problem of load imbalance and high migration costs. Usually load imbalance and high number of VM migrations occurs if the scheduling is performed using traditional algorithms.

3.9 ENERGY AWARE SCHEDULING

3.9.1 Hybrid Metaheuristic for Energy-Aware Scheduling

Precedence-constrained parallel applications are one of the most typical application model used in science and engineering fields. The main objective is finding the right compromise between the resolution time and the energy consumed of a precedence-constrained parallel application. Many techniques are used for energy consumption including Dynamic Voltage Scaling (DVS), resource hibernation, and memory optimizations. DVS is a very promising technique with its demonstrated capability for energy savings.

DVS enables processor to dynamically adjust Voltage Supply Levels (VSLs) aiming to reduce the power consumption. Hybrid Genetic Algorithm is used to reduce the completion time and power. The performance of the Hybrid Genetic Algorithm is comparatively better than Task Duplication based Bottom up Scheduling algorithm and Heterogeneous Earliest-Finish-Time algorithm. The cooperative approach of the island model and the hybridization improves the quality of the output and the multi start model reduces the running time of a resolution.

Mezmaza et al (2012) analyzed the problem of scheduling precedence constrained parallel applications on Heterogeneous Computing Systems (HCSs) like cloud computing infrastructures. New parallel bi-objective hybrid genetic algorithm not only takes makespan into account, but also energy consumption. Hybrid approach is based on the Dynamic Voltage
Scaling (DVS) to minimize energy consumption. The insular approach shows that the more islands used, the more the resources are needed to minimize the energy consumption and makespan.

3.10 SUMMARY

Here, different types of Cloud Computing algorithms have been discussed for various types of scheduling with the relevant theoretical study. The Scheduling optimization have been addressed and analyzed for the resource allocation issues when the resource is on demand in the distributed environment. The solution was identified based on the theoretical analysis made and using eight schemes to alleviate the scheduling optimization problem.