Abstract— Cloud computing is a general term used to describe a new class of network based computing that takes place over the internet. The primary benefit of moving to Clouds is application scalability. Cloud computing is very beneficial for the application which are sharing their resources on different nodes. As Cloud computing is based on the concepts of distributed computing, grid computing, utility computing and virtualization. Scheduling the task is quite a challenging in cloud environment. Usually tasks are scheduled by user requirements. New scheduling strategies need to be proposed to overcome the problems proposed by network properties between user and resources. New scheduling strategies may use some of the conventional scheduling concepts to merge them together with some network aware strategies to provide solutions for better and more efficient job scheduling. Scheduling strategy is the key technology in cloud computing. This paper compared different types of workflow scheduling algorithms. There working with respect to the resource sharing. We systemize the scheduling problem in cloud computing, and present a cloud scheduling hierarchy, mainly splitting into user-level and system-level.

Keywords- Cloud Computing, Scheduling, Task, Static, Dynamic, Workflow

I. INTRODUCTION

Amazon played a key role in Cloud Computing development by launching Amazon web service on utility basis in 2006.

Cloud computing emerges as a new computing paradigm which aims to provide reliable, customized and QoS (Quality of Service) guaranteed computing dynamic environments for end-users. Distributed processing, parallel processing and grid computing together emerged as cloud computing. Managing resources at large scale while providing performance isolation and efficient use of underlying hardware is a key challenge for any cloud management software.

Optimal resource allocation or task scheduling in the cloud should decide optimal number of systems required in the cloud so that the total cost is minimized and the SLA is upheld. Scheduling is a critical component of the cloud resource management. Scheduling is responsible for resource sharing/multiplexing at several levels; a server can be shared among several virtual machines, each virtual machine could support several applications, and each application may consist of multiple threads.

As Cloud computing is highly dynamic, and hence, resource allocation problems have to be continuously addressed, as servers become available/nonavailable while at the same time the customer demand fluctuates.

The objective of this paper is to be focus on various workflow scheduling algorithms.

II. NEED OF SCHEDULING IN CLOUD

The primary benefit of moving to Clouds is application scalability. Unlike Grids, scalability of Cloud resources allows real-time provisioning of resources to meet application requirements. Cloud services like compute, storage and bandwidth resources are available at substantially lower costs. Usually tasks are scheduled by user requirements. New scheduling strategies need to be proposed to overcome the problems posed by network properties between user and resources. New scheduling strategies may use some of the conventional scheduling concepts to merge them together with some network aware strategies to provide solutions for better and more efficient job scheduling [1]. Usually tasks are scheduled by user requirements.
Due to the reduced performance faced in grids, now there is a need to implement scheduling in cloud. The primary benefit of moving to Clouds is application scalability.

Traditional way for scheduling in cloud computing tended to use the direct tasks of users as the overhead application base. The problem is that there may be no relationship between the overhead application base and the way that different tasks cause overhead costs of resources in cloud systems. For large number of simple tasks this increases the cost and the cost is decreased if we have small number of complex tasks.

### III. SCHEDULING ALGORITHM FOR COMPUTING CLOUDS

Scheduling is a critical component of the cloud resource management. Scheduling is responsible for resource sharing/multiplexing at several levels; a server can be shared among several virtual machines, each virtual machine could support several applications, and each application may consist of multiple threads. CPU scheduling supports the virtualization of a processor, the individual threads acting as virtual processors; a communication link can be multiplexed among a number of virtual channels, one for each flow. In addition to the requirement to meet its design objectives, a scheduling algorithm should be efficient, fair, and starvation-free. The objectives of a scheduler for a batch system are to maximize the throughput (the number of jobs completed in one unit of time, e.g., in one hour) and to minimize the turnaround time (the time between job submission and its completion); for a real-time system the objectives are to meet the deadlines and to be predictable. Schedulers for systems supporting a mix of tasks, some with hard real-time constraints, others with soft, or no timing constraints, are often subject to contradictory requirements. Some schedulers are preemptive, allowing a high-priority task to interrupt the execution of a lower priority one, others are non-preemptive.

Two distinct dimensions of resource management must be addressed by a scheduling policy: (a) the amount/quantity of resources allocated; and (b) the timing when access to resources is granted. Hard-real time systems are the most challenging as they require strict timing and precise amounts of resources. There are multiple definitions of a fair scheduling algorithm. Consider a resource with bandwidth $B$ shared among $n$ users who have equal rights; each user requests an amount $b_i$ and receives $B_i$.

### IV. TASK SCHEDULING TYPES

Cloud service scheduling is categorized at user level and system level. At user level scheduling deals with problems raised by service provision between providers and customers. The system level scheduling handles resource management within datacenter. Datacenter consists of many physical machines. Millions of tasks from users are received; assignment of these tasks to physical machine is done at datacenter. This assignment or scheduling significantly impacts the performance of datacenter. In addition to system utilization, other requirements like QoS, SLA (Service Level Agreements), resource sharing, fault tolerance, reliability, real time satisfaction etc should be taken into consideration.

### A. User Level Scheduling

Market-based and auction-based schedulers are suitable for regulating the supply and demand of cloud resources. Market based resource allocation is effective in cloud computing environment where resources are virtualized and delivered to user as a service. Development of a pricing model using processor-sharing for clouds, the application of this pricing model to composite services with dependency consideration and the development of two sets of profit-driven scheduling algorithms are proposed in [2].

Service provisioning in Clouds is based on Service Level Agreements (SLA). SLA represents a contract signed between the customer and the service provider stating the terms of the agreement including non-functional requirements of the service specified as Quality of Service (QoS), obligations, and penalties in case of agreement violations. Thus there is a need of scheduling strategies considering multiple SLA parameters and efficient allocation of resources. The scheduler algorithm that allows re-provisioning of resources on the cloud in the event of failures. The focus of model is to provide fair deal to the users and consumers, enhanced quality of service as well as generation of optimal revenue. A novel cloud scheduling scheme uses SLA along with trust monitor to provide a faster scheduling of the over flooding user request with secure processing of the request. A novel approach of heuristic-based request scheduling at each server, in each of the geographically distributed data centers, to globally minimize the penalty charged to the cloud computing system. This approach considers two variants of heuristics, one based on the simulated annealing method of neighborhood searches and another based on gi-FIFO scheduling. Based on the queuing model and system cost function, considering the goals of both the cloud computing service users and providers. This approach guarantees the QoS requirements of the users’ as well as the maximum profits for the cloud computing service providers. To deal with dynamically fluctuating resource demands, market-driven resource allocation has been proposed and implemented by public Infrastructure-as-a-Service (IaaS) providers like Amazon EC2. In this environment, cloud resources are offered in distinct types of virtual machines (VMs) and the cloud provider runs an auction-based market for each VM type with the goal of achieving maximum revenue over time. A case study of single cloud provider and how to best match customer demand in terms of both supply and price in order to maximize the providers revenue and customer satisfactions while minimizing energy [3]. Another auction-based mechanism for dynamic VM provisioning and allocation that takes into account the user demand for VMs when making VM provisioning decisions.

### B. HEURISTIC MODELS FOR TASK EXECUTION SCHEDULING

This section intensively researches two types of strategies, static and dynamic heuristics. Static heuristic is suitable for the situation where the complete set of tasks is known prior to execution, while dynamic heuristic performs the scheduling when a task arrives.

1) **Static strategies:** Static strategies are performed under two assumptions. The first is that tasks arrive simultaneously $ci = 0$. The second is that machine available time $aj$ is updated after each task is scheduled.

2) **Dynamic strategies:** Dynamic heuristics are necessary when task set or machine set is not fixed. For example, not all tasks arrive simultaneously, or some machines go offline at
intervals. The dynamic heuristics can be used in two fashions, on-line mode and batch mode. In the former mode, a task is scheduled to a machine as soon as it arrives. In the latter mode, tasks are firstly collected into a set that is examined for scheduling at prescheduled times.

C. STATIC AND DYNAMIC SCHEDULING

Static scheduling allows for pre-fetching required data and pipelining different stages of task execution. Static scheduling imposes less runtime overhead. In case of dynamic scheduling information of the job components/task is not known beforehand. Thus execution time of the task may not be known and the allocation of tasks is done on fly as the application executes. A job execution environment flextic that exploits scalable static scheduling techniques to provide the user with a flexible pricing model and at the same time, reduce scheduling overhead for the cloud provider. The service request scheduling strategies in three-tier cloud structure, which consists of resource providers, service providers and consumers, should satisfy the objectives of the service providers and consumers. A new fault tolerant scheduling algorithm MaxRe incorporates the reliability analysis into the active replication schema, and exploits a dynamic number of replicas for different tasks. Trust relationship is built among computing nodes, and the trustworthiness of nodes is evaluated by utilizing the Bayesian cognitive method. A preemptable scheduling improves the utilization of resources in clouds and feedback procedure in above algorithms works well in the situation where resource contentions are fierce. In cloud computing, traditional way for task scheduling cannot measure the cost of cloud resources accurately by reason that each of the tasks on cloud systems is totally different between each other. A scheduling algorithm is proposed that measures both resource cost and computation performance and also improves the computation/communication ratio by grouping the user tasks according to a particular cloud resource’s processing capability and sends the grouped jobs to the resource. Due to job grouping, communication of coarse-grained jobs and resources optimizes computation/communication ratio. A large number of cloud computing servers waste a tremendous amount of energy and emit a considerable amount of carbon dioxide. Green task scheduling is necessary to significantly reduce pollution and substantially lower energy usage. A fully decentralized scheduler aggregates information about the availability of the execution nodes throughout the network and uses it to allocate tasks to those nodes those are able to finish them in time. As study considering the realistic network topology and communication model, proposes the Deadline, Reliability, Resources-aware (DRR) scheduling algorithm. Considering the failure and recovery scenario in the Cloud computing entities, a Reinforcement Learning (RL) based algorithm to make job scheduling fault-tolerable while maximizing utilities attained in the long term.

D. HEURISTIC SCHEDULING

Optimization problems are in Class NP-hard. These problems can be solved by enumeration method, heuristic method or approximation method. In enumeration method, an optimal solution can be selected if all the possible solutions are enumerated and compared one by one. When number of instances is large, exhaustive enumeration is not feasible for scheduling problems. In that case heuristic is a suboptimal algorithm to find reasonably good solutions reasonably fast. Approximation algorithms are used to find approximate solutions to optimized solution. These algorithms are used for problems when exact polynomial time algorithms are known. Enhancing task data locality in large scale data processing systems is crucial for the job completion time. Most of the approaches to improve data locality are either greedy and ignore global optimization, or suffer from high computation complexity. This problem is addressed by proposing a heuristic task scheduling algorithm called Balance-Reduce (BAR). Load balancing task scheduler balance the entire system load while trying to minimizing the make span of a given tasks set. Two different load balancing scheduling algorithms based on ant colony. Cloud Loading Balance algorithm, adds capacity to the dynamic balance mechanism for the cloud environment. The decision, which workloads to outsource to what cloud provider, should maximize the utilization of the internal infrastructure and minimize the cost of running the outsourced tasks in the cloud, while taking into account the applications’ quality of service constraints. A set of heuristics, to cost-efficiently schedule deadline constrained computational applications. Multi-objective meta-heuristics scheduling algorithm for multi-cloud environment tries to achieve application high availability and fault-tolerance while reducing the application cost and keeping the resource load maximized. Because of the increasing large Web graph and social networks, cost-conscious large graph processing scheduling is important. COA (Course of action) planning involves resource allocation and task scheduling. Reducing energy consumption is an increasingly important issue in cloud computing, more specifically when dealing with High Performance Computing (HPC). A multi-objective genetic algorithm (MO-GA), optimizes the energy consumption, carbon dioxide emissions and the generated profit of a geographically distributed cloud computing infrastructure. Simulated annealing is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. The scalability of a computing system can be mainly identified by size, geographical distribution, administrative constraints, heterogeneity, energy consumption and transparency. A low complexity energy efficient heuristic algorithm for scheduling performs efficiently demonstrating their applicability and scalability.

In batch mode, tasks are scheduled only at some predefined time. This enables batch heuristics to know about the actual execution times of a larger number of tasks. Min-min and Max-min are heuristics used for batch mode scheduling. Bag of tasks (BoT) applications are the one which execute independent parallel tasks. Heuristics aims to maximize resource utilization while executing BoTs in heterogeneous sets of Cloud resources allocated for different numbers of hours. Another budget constraint scheduler schedules large bags of tasks onto multiple clouds with different CPU performance and cost, minimizing completion time while respecting an upper bound for the budget to be spent. When providers cannot disclose private information such as their load and computing power, which are usually heterogeneous, the meta-scheduler needs to make blind scheduling decisions.

E. REAL TIME SCHEDULING FOR CLOUD COMPUTING

There are emerging classes of applications that can benefit from increasing timing guarantee of cloud services. These mission critical applications typically have deadline requirements, and any delay is considered as failure for the whole deployment. For
instance, traffic control centers periodically collect the state of roads by sensor devices. Database updates recent information before next data reports are submitted. If anyone consults the control center about traffic problems, a real-time decision should be responded to help operators choose appropriate control actions. Besides, current service level agreements cannot provide cloud users real-time control over the timing behavior of the applications, so more flexible, transparent and trust-worthy service agreement between cloud providers and users is needed in future. Given the above analysis, the ability to satisfy timing constraints of such real-time applications plays a significant role in cloud environment. However, the existing cloud schedulers are not perfectly suitable for real-time tasks, because they lack strict requirement of hard deadlines. A real-time scheduler must ensure that processes meet deadlines, regardless of system load or make span. Priority is applied to the scheduling of these periodic tasks with deadlines. Every task in priority scheduling is given a priority through some policy, so that scheduler assigns tasks to resources according to priorities. Based on the policy for assigning priority, real-time scheduling is classified into two types: fixed priority strategy and dynamic priority strategy.

F. WORKFLOW SCHEDULING

A workflow enables the structuring of applications in a directed acyclic graph, where each node represents the constituent task and edges represent inter task dependencies of the applications. A single workflow generally consists of a set of tasks each of which may communicate with another task in the workflow. Workflow scheduling is one of the key issues in the management of workflow execution. A Survey of various workflow scheduling algorithms in cloud environment is documented in [4]. Workflows constitute a common model for describing a wide range of scientific applications in distributed systems. Workflow scheduling is the problem of mapping each task to appropriate resource and allowing the tasks to satisfy some performance criterion. Workflow is processes that consist of a series of steps which simplifies the complexity of executions and management of applications. Workflow mainly focused with the automation of procedures and also in order to achieve a overall goal thereby files and data are passed between participants according to a defined set of rules. A single workflow consists of a set of tasks and each task communicates with another task in the workflow. Workflows are supported by Workflow Management Systems. Workflow scheduling discovers resources and allocates tasks on suitable resources. Workflow scheduling plays a vital role in the workflow management. Proper scheduling of workflow can have an efficient impact on the performance of the system.

V. LITERATURE SURVEY

A Particle Swarm Optimization based Heuristic for Scheduling Workflow Applications: Pandey, Linlin Wu, Guru, Rajkumar Buyya[6] presented a particle swarm optimization (PSO) based heuristic to schedule the applications to cloud resources that takes both computation and data transmission cost. It is used for workflow applications by varying its computation and communication costs. The evaluation results shows that PSO can minimize the cost and good distribution of workload onto resources. Heterogeneous Earliest Finish Time algorithm (HEFT): Topcuoglu et. Al[7], presented the HEFT algorithm. This algorithm finds the average execution time of each task and also the average communication time between the resources of two tasks. Then tasks in the workflow are ordered on a rank function. Then the task with higher rank value is given higher priority. In the resource selection phase tasks are scheduled in priorities and each task is assigned to the resource that complete the task at the earliest time.

Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi[8] worked on multiple workflows and multiple QoS. They implemented a strategy for multiple workflow management system with multiple Quality of Service. The access rate for scheduling is increased by using this strategy. This strategy minimizes the make span and cost of workflows.

Market-Oriented-Hierarchical Scheduling: Salehi, M.A. and Buyya, R.[9] proposed a market oriented hierarchical scheduling strategy which consists of both service level scheduling and task level scheduling. The service level scheduling deals with the Task to Service assignment and the task level scheduling deals with the optimization of the Task to Virtual Machine assignment in local cloud data centers.

Scheduling workflows with budget constraints: Sakellariou, R., Zhao, H., Tsiakkouri, E. and Dikaikos, M.D[10] proposed a basic model for workflow applications that modelled as directed acyclic graph (DAGs) and that allow to schedule the nodes of DAG onto resources in a way that satisfies a budget constraint and is optimized for overall time.

Cost based scheduling of scientific workflow applications on utility grids: Yu, J., Buyya, R. and Tham, C.K.[11] proposed a cost based workflow scheduling algorithm minimizes the execution cost while meeting the deadline for delivering results. It can also adapt to the deays of service executions by rescheduling unexecuted tasks.
The objectives of a scheduler for a batch system are to maximize the throughput and to minimize the turnaround time. This paper explores methods of scheduling done in cloud computing. It helps to understand the wide task scheduling options in order to select one for a given environment. Workflow scheduling is one of the major issues in cloud computing environment. The various existing workflow scheduling algorithms in cloud computing and compare their various parameters. Existing workflow algorithms does not consider the execution time. Therefore there is a need to implement a new scheduling algorithm that can minimize the execution time in cloud environment. Moving workflows to a cloud computing environment enables the use of various cloud services to facilitate workflow execution.

VI. CONCLUSION

REFERENCES


Author’s Profile:

Miss. Priya R. Lodha is doing M.E (CSE) from G.H Raisoni College of Engineering and Management Amravati and has done B.E in Information Technology from SGBAU, Amravati.

Prof. Avinash P. Wadhe: Received the B.E and from SGBAU Amravati university and M-Tech (CSE) From G.H Raisoni College of Engineering, Nagpur (an Autonomous Institute). He is currently an Assistant Professor with the G.H Raisoni College of Engineering and Management, Amravati SGBAU Amravati university. His research interest include Network Security, Data mining and Fuzzy system .He has contributed to more than 20 research paper. He had awarded with young investigator award in international conference.