# A Survey on Task Scheduling Algorithms in Cloud Computing Environment

Mehwish Awan*, Munam Ali Shah
Department of Computer Science
COMSATS Institute of Information Technology,
Islamabad, Pakistan.
*Email: mehwishawan [AT] rocketmail.com

***Abstract— Cloud Computing is an emerging technology evolving to the new computing framework. Scheduling of task and allocation of virtualized resources in such a widely distributed dynamic environment is a challenging assignment. Attaining proficiency and providing fairness to tasks execution is the primary consideration of the task scheduling algorithm. The objective of this paper is to give a comparative analysis of task scheduling algorithms in cloud platform, where the resources have varying cost and computational efficiency.***

***Keywords— Cloud Computing, Task Scheduling, Algorithm.***

## I. INTRODUCTION

In the present era, Cloud Computing is the state-of-the-art technology advancing with miraculous triumph across the IT globe. Cloud computing is a modern computing paradigm in the spheres of IT. Its groundwork refined from grid computing, parallel computing and distributed computing which provides dynamic services over the Internet [1]. Cloud Computing could be defined as a parallel and distributed system containing a number of dynamically interconnected and virtualized machines that are uniformly allocated over cloud to provide services to consumers according to their need, for which consumers just have to pay as much as they consume [2]. It has achieved the standards of a chief technology to support coherence [3]. Its infrastructure adopts some of idea from utility computing [4], cloud computing gives utility to Business organizations along with individuals as it aims to provide large scale services at minimum cost. Cloud computing gives the following features: elastic resource capacity, management automation, virtualized resources, self-service provisioning, utility pricing, managed operations, and third party owner-ship. It hides the heterogeneity of resources, which are dynamically shared over the cloud and also provides transparency to the users. Virtual machines wrap the divergent resources into virtual pool of computing resources and give more scalability with less management overheads in data centers. The traditional centralized computing task scheduling mechanisms cannot be adjusted in cloud environment where the resources are jumbled and tasks are dynamically scheduled [5].

The goal of Cloud provider is to dynamically render the widely distributed set of services e.g. hardware, software, applications, huge data storage, virtualized resources, high throughput, improved resource utilization, economics of scale, high reliability, Quality of Service (QoS), build-in disaster recovery and high efficiency to the customers. Therefore, in order to achieve these objectives one of the most imperative integral of cloud computing is Task Scheduling Strategy. Task scheduling is an extensive optimization issue and NP-Completeness problem; it gives a major contribution to improve reliable and extensible dynamic systems. The best ever feature Cloud Computing has introduced is monetary and virtualization technology, that puts the task scheduling intricacy of cloud computing to the virtual machines layer through resource virtualization [6].

As cloud computing is largely dependent on virtualization, therefore efficient task scheduling over virtualized machines and heterogeneous physical resources is a challenging task. Attaining proficiency and providing fairness to tasks execution is the primary consideration of the task scheduling algorithm. The scheduling algorithm has got key importance in cloud environment while scheduling the tasks, where the objective is to achieve efficiency, hence to minimize turnaround time, increase throughput and reduce cost under changing scenarios [7]. Task scheduler is a gauging component which functions from the consumer's task request to the selection of apropos resources till the execution. In cloud computing job scheduling is a challenging issue. The goal of task scheduling is to divide the workload over servers in such a manner to maximize utilization in cloud network and minimize total execution time. The service requests at cloud platform have multiplexed resource requirements, as some services are CPU-demanding while other are I/O-demanding, in order to enhance cloud service providers financial benefits not just Quality of Service (QoS), tasks must be scheduled and cloud resources need to be designated to reduce energy usage as well [8]. For Virtualized Multi-tier Application Execution Environments (VAEEs) to achieve progressive provisioning of resources for distinct clients, the most frequent approaches are placed on self-managing techniques [9]. For example, a renowned cloud service providers, Amazon Elastic Compute Cloud (EC2) is an *Infrastructure as a Service (IaaS)* cloud, by the adoption of virtual machines, clients create as many virtual machines as they need and then anchor their all services within VMs after optimized scheduling [10].

Figure. 1 illustrates the General scheduling Mechanism in Cloud computing, where the entire cloud works on 3 basic architecture layers, 1) *Software as a Service (SaaS)*, this is an

application layer which offers applications and services on demand [7], at this layer user requests are recognized, i.e, for which service request is generated e.g. CRM, mailing, games, web browsing, thin-client, mobile apps etc., and then delivered to next architecture layer, 2) *Platform as a Service (Paas),* scheduling of tasks is performed at this layer, PaaS includes following cloud components: web server, development tools, scheduler, database, execution runtime etc. 3) *Infrastructure as a Service (IaaS),* in infrastructure layer virtualization is the fundamental element. Other components infrastructure layer contains are cloud data centers, VM pool, load balancers and servers etc. All the services on cloud are accessed by using *Application Programming Interfaces* (APIs) [11].

The remaining part of the paper is organized in the following sections, Section 2: task scheduling models, Section 3: existing task scheduling algorithms, Section 4: analysis and comparison, Section 5: conclusion.

## II. Task Scheduling Models

In cloud environment task scheduling is used to schedule the tasks fairly at cloud platform and allocate particular tasks to appropriate resources in such a way to maximize the resource utilization, enhance the performance and QoS at the same time, sustaining high efficiency as well as economy at execution time. There are different scheduling models depending upon scheduling criteria. Based on simple taxonomy task scheduling algorithms in cloud computing domain are categorized into following major groups.

*1) Static Scheduling Model:*

It is an easier approach positioned on scheduler viewpoint, as the information of all the available resources and tasks is known and resource is designated once to the task [12].

*2) Dynamic Scheduling Model:*

It is a flexible approach than static scheduling but it has more execution overhead as compared with static [13]. It executes task dynamically by assigning resources at run time.

*3) Hybrid Scheduling Model:*

It is the combination of above both scheduling techniques. This technique is used at virtual machines.

*4) Distributed Scheduling Model:*

This scheduling model is more convenient for cloud platform, it is realistic approach where task requests are received dynamically. This approach is fragile as compared to centralized scheduling but more feasible in context of scheduling in distributed environment.

*5) Centralized Scheduling Model:*

It is more biddable, easy to implement, and more manageable approach. But this approach lacks scalability hence not sufficient for large scale clusters.

*6) Cooperative Scheduling Model:*

It's a scheduling approach in which different schedulers perform their roles in a common system cooperatively in a group according to defined rules. Each scheduler is responsible of assured activity which is predefined in the task scheduling process of the system.

*7) Non-cooperative Scheduling Model:*

It is a scheduling model where schedulers donot have privilege to execute their roles as a group in scheduling process of a common system.

*8) Batch Mode Heuristic Algorithm (BMHA) / off-line Mode:*

In this model of scheduling, scheduler receives the coming tasks and collects them as sets of batches then each batch is executed as a unit [14].

*9) Prompt / On-line Mode:*

In this scheduling model the tasks are scheduled individually as they are received, no waiting for next batch time interval. This model is more compatible with cloud environment, owing to the cloud's non-homogeneity and differing speed of processors [4].

*10) Pre-emptive Scheduling Model:*

This is a type of scheduling where priorities are assigned to the tasks. The task with higher priority is executed first although the other task with low priority may be running; the running process is interrupted for some time and is resumed after the completion of high priority task [15].

*11) Non-preemptive Scheduling Model:*

It doesn't allow to interrupt the executing task until terminated. The resources once allocated to the task do not re-allocate after interrupt completion or I/O wait etc. [15].

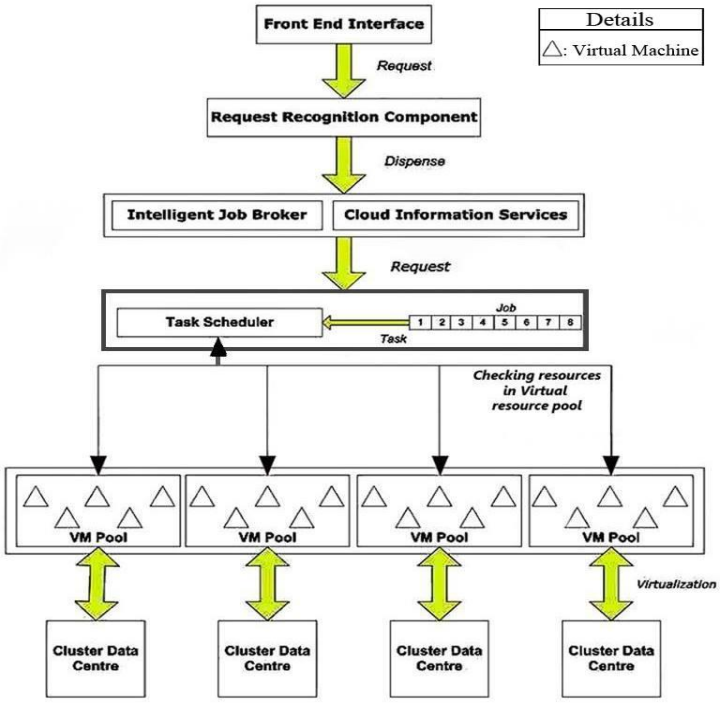


**Figure 1 General task scheduling mechanism in cloud computing environment**

## III. Related Work

Many scheduling algorithms have been proposed so far in cloud environment for task scheduling. In this survey we have categorized some major cloud computing task scheduling algorithms and analyzed them.

### A. Priority Based Task Scheduling Techniques in Cloud

- *A Priority Based Job Scheduling Algorithm In Cloud Computing:* This paper was presented by Shamsollah Ghanbari and Mohamed Othman [16]. In this paper

they proposed a priority based job scheduling algorithm in cloud computing (PJSC) by using mathematical demography. It is a multi-criteria decision-making (MCDM) and multi-attribute decision-making (MCDM) model. The approach is based on theory of Analytical Hierarchy Process (AHP). The bedrock of PJSC is consistent comparison matrix, job requests a resource with a determined priority. The priority of each job is compared with other jobs separately, so comparison matrixes of each job according to resources accessibilities is computed along with comparison matrix of resources. For each of comparison matrixes a priority vector (vector of weights) is computed. This algorithm consists of three level priorities which include: scheduling level (objective level), resources level (attribute level) and job level (alternative level) [16]. This algorithm has complexity issue, while computing priority of comparison matrixes. The worst-case complexity of this algorithm is computed as [16]:

$$\Omega = \alpha^{2.81} + d * m^{2.81}.$$

The other drawback of this algorithm is makespan (finish time) this algorithm only focuses on priority and doesn't give optimal finish time. To handle with inconsistency is another short-coming of this approach.

- *A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment:* Monika Choudhary and Sateesh Kumar Peddoju presented this paper [17]. They merged the concepts of three different approaches i-e Task Grouping, Prioritization and Greedy Allocation to deliver a new scheduling technique which permits improved performance than above approaches in cloud computing environment. It works on two constraint groups 1) deadline constrained tasks 2) cost based tasks. When tasks are received task broker groups the tasks according to above constraint categories, priorities are assigned to each task based on their group type. Then virtual machines are designated according to task groups, waiting times and resource capacity of selected machines are updated accordingly. CloudSim is used to justify the faultlessness of this algorithm, tests are conducted with sequential approach, the jobs are dynamically distributed to establish a generic scheme. It has been seen from results that in comparison to sequential approach this algorithm gives improved cost and better completion time of tasks. The impediment with this algorithm is: it should also consider the tasks type and tasks length for convenient tasks scheduling. Moreover communication overhead should also be minimized by grouping the cost based tasks before resource division according to resource capacity [17].

- *An Improved Adaptive Genetic Algorithm In Cloud Computing:* An improved Adaptive Genetic Algorithm (PAGA) based on priority driven approach for emergency computing task scheduling in cloud computing was proposed by HU Baofang, SUN Xiuli, LI Ying and SUN Hongfeng [18]. Mathematical model is presented to examine lower response time of tasks with higher priority. User tasks are divided into sub-tasks and executed on cloud computing resources. According to the level of sponsor the priority is assigned to each task, and the sub-tasks inherit the priority of their parent task. This algorithm shows faster convergence speed and run-over the drawback of traditional Adaptive Genetic Algorithm (AGA). The experimental results show the effectiveness and feasibility of this algorithm over the other compared algorithms. It gives a new perception of problem-solving for task scheduling algorithm of emergency computing in cloud computing environment. While dealing with dilemma of assigning priorities to tasks and sub-tasks this algorithm faces complexity issue.

### B. *Reduced Makespan Based Task Scheduling Techniques in Cloud*

- *Bees Life Algorithm For Job Scheduling In Cloud Computing:* Salim Bitam [10] proposed this algorithm being inspired from bees in nature. This approach is used for neighborhood search. It basically runs in a cycle which has 13 steps of implementation. It starts with bees population initialization step which has N bees (individual) randomly chosen in search space [10]. Population fitness is evaluated in the next step and roles are classified according to most fitted individuals. The next step is decision making stopping criteria, the Bees Life Algorithm (BLA) cycle's iteration continues as long as the fitness changes and stops when fitness does not change. This is named as stagnation state. To optimize BLA two genetic operators are applied in reproduction phase, Crossover and Mutation. Crossover is a binary operator in which the queen mates with one randomly chosen drone. This process is repeated until reaching 'N' individuals with crossover probability of 'Pc'. Two crossover points are practiced in this paper, parents are isolated at these two points randomly selected to generate children after exchanging the adversity with correction if children do not belongs to the search space. Mutation is a unitary operator it basically choses a task as a randomly selected task which will be placed by another randomly chosen task in the same cloud data center according to mutation probability of 'Pm'. Greedy approach is used in BLA foraging part in which task of one datacenter is randomly selected to be allowed with another task in the nearest datacenter. An implementation of BLA and tests has been performed and compared with Genetic Algorithm (GA) as traditional algorithm in this context, the results collected from those analysis shows efficiency of the algorithm. But the inconvenience with this approach is: it only focused on makespan which could results in unfairness among tasks.

- *Enhanced Max-Min Task Scheduling Algorithm In Cloud Computing:* Upendra Bhoi ,and Purvi N. Ramanuj [19] proposed this algorithm, the concept was borrowed from notion of IMPROVED MAX - MIN TASK SCHEDULING ALGORITHM. The scenario of improved max-min approach is: the task with largest execution time is assigned a slowest resource and vice versa. This will grant the high speed resources to other small tasks concurrently. The flaw in this approach is that sometimes the largest task is too large in Meta-task, in such case overall makespan increases and load imbalance occurs across resources. So author changes the scenario, selects the task with average or nearest greater than average task then assigns to be executed resource with minimum completion time (slowest resource). So the overall makespan is reduced and also balance load across resources.

- *Independent Tasks Scheduling Based On Genetic Algorithm In Cloud Computing:* this paper was presented by Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie and Jicheng Hu. This algorithm model is based on SOA Model (Service Oriented Architecture) where every element of the system is considered as service node. Each service has a hierarchical hyper graph structure [20]. It works on principle of Genetic Algorithm in accordance to meet memory constraint and high performance demand in cloud computing. It assumes the centralized scheduling model as a master processor unit to collect all tasks and dispatch them to other process unit. This algorithm takes into account both time utilization and resource utilization factors to achieve better results. The experiment is constructed across distinct platforms where two tasks with different CRVS to be executed on two process units, one was 32-bit and other was 64-bit.. The expected completion time and responded completion time has 15/8 ratio. At 100th generation optimal fitness value approached 2.9. Experimental result shows optimal resources utilization, although this algorithm showed advantageous results, but some attention should also be paid to curtail solution space in Genetic Algorithm.

- *Host Scheduling Algorithm Using Genetic Algorithm In Cloud Computing Environment:* Tarun Goyal and Aakanksha Agrawal designed an algorithm based on the principle of Genetic Algorithm companioned with Suffrage Heuristic. It aims to schedule multiple jobs on multiple machines dynamically in such a way to minimize the makespan and enhance efficiency. As in Genetic Algorithm population fitness is the most crucial step in order to provoke better generation, in this algorithm suffrage heuristic is implemented for individual generation. Experiments are conducted using cloud-sim as simulator to analyze the performance of this algorithm. It gives better initial population and improved results as correlate with generic Genetic Algorithm [21]. The bottleneck of this algorithm is that it uses the fixed number of virtual machines during experiments, as cloud is a dynamic environment where changes always use to happen according to situations, so different experiments should be performed with varying number of virtual machines to check the vigorous behavior of algorithm. Another drawback is it only focuses to minimize the completion time while neglecting the factors which also influence systems overall performance.

### C. *Energy Efficient Task Scheduling Techniques in Cloud*

- *Job Scheduling Model For Cloud Computing Based On Multi-Objective Genetic Algorithm:* This paper was presented by Jing Liu, Xing-Guo Luo , Xing-Ming Zhang , Fan Zhang and Bai-Nan Li [8]. A macroscopic scheduling model with cognition and decision components is proposed for the cloud computing. This scheduling algorithm is based on Multi-Objective Genetic Algorithm (MO-GA), it focuses on to minimize energy consumption and maximize the profit of service providers under the constraint of deadlines. And implementing a dynamic selection approach to the most convenient scheduling strategy for users according to the real-time demands. Implementation steps of this algorithm include: Population initialization is an important step in whole algorithm, in this paper this constraint is met by combining random and greedy initialization methods. Meanwhile evolution process of MO-GA it customizes the individuals in keeping with the operators marked. The results calculated from MO-GA algorithm are a set of Pareto solutions and give a wide range of options. MO-GA based algorithm is compared to the other algorithms and according to the results of this paper it obtains high profits and consumes less energy. But for the higher arrival rates it gives the worse results, and number of failed applications also increases as the arrival rates get higher, so the whole cloud sops and engages with the high arrival rate, with no or minimum capacity to accept the new arrival applications. Results from MO-GA based algorithm are also compared with three different vectors to examine the impact of selection vectors on MO-GA, the results from two of the vectors V1 and V3 shows compelling progression in scheduling objective, where V2 doesn't give favorable results [8].

### D. *Improved Cost Based Task Scheduling Techniques in Cloud*

- *Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing:* This cost based scheduling model is presented by Mrs.S.Selvarani, and Dr.G.Sudha Sadhasivam [1]. It aims to minimize total tasks completion time and cost. This algorithm works as: tasks are divided into three different lists based on their priority level, different available resources are allocated to those tasks according to their priority lists and grouping algorithm is applied to schedule tasks in each list. CloudSim is used for simulation of experiments, the comparison of this improved Activity Based Costing (ABC) algorithm is conducted with ABC scheduling algorithm. Outcome shows the effectiveness of Improved ABC scheduling algorithm over ABC scheduling algorithm. Shortcoming: As cloud is a dynamically changing environment where a large number of user requests and demands vary among users, this scenario is showing quite limited cloud platform, therefore some dynamic factors should also be entertained for influential cloud environment.

### E. *QoS Based Task Scheduling Techniques in Cloud*

- *A Qos Guided Task Scheduling Model In Cloud Computing Environment:* Haiwen Han, Qi Deyu, Weiping Zheng and Feng Bin proposed this algorithm. This algorithm model is inherited from Sufferage Heuristic Algorithm, Min-min Algorithm, and QoS guided scheduling strategies. It mainly considers the QoS requirements, dynamic and heterogeneity resource properties and low execution complexity in cloud environment [22]. Competition occurs for resources among tasks with different QoS request levels. It schedules the tasks in a manner that on the resources with high QoS capability the group of tasks with higher QoS requests is executed first. The comparison is made between this algorithm and other heuristic approach, results show the improved performance of this algorithm in context of shortened makespan (completion time). The drawback found in this algorithm is: as the group of tasks with higher QoS requests is executed first this may result in starvation of the task groups with low QoS requests.

- *A Task Scheduling Algorithm Based On QoS And Complexity-Aware Optimization In Cloud Computing:* This task scheduling algorithm is proposed by WANG Ning, YANG Yang, MENG Kun, CHEN Yu and DING Hao. This algorithm focuses on achieving higher QoS (quality of service) by the lower service cost. In this paper task scheduling complexity is computed by fuzzy identification method. The complexity of minimum dynamic service-quality cost (MDSCT) is O(K3). It also considers the factors effecting cloud overall performance, including the cloud service cost, node's load rate, bandwidth utilization, and network delay. Service delay from cloud nodes to user nodes is anticipated by mapping cloud nodes to 5-dimensional network coordinate system on Euclidean space [23]. Mathematical model is presented for measuring service quality and service cost by investigating the elements influencing performance and cost. An experimental result demonstrates that MDSCT performs scheduling of tasks across cloud nodes effectively and also attains higher QoS with lower service cost. But this algorithm runs with insufficiency of storage space and computation complexity as in real cloud environment tasks are in larger number than the number of tasks used in experimental analysis of MDSCT.

*F. Improved Consistency Based Task Scheduling Techniques in Cloud*

- *The Analytic Hierarchy Process: Task Scheduling And Resource Allocation In Cloud Computing Environment:* Daji Ergu ·Gang Kou ·Yi Peng ·Yong Shi and Yu Shi [5] presented an approach to schedule tasks and allocate resources in cloud computing that focuses to improve consistency of comparison matrixes. It is an algorithm based on analytical hierarchy process (AHP). It works on the principle of assigning cloud storage and cloud resources to the reciprocal tasks according to the weight of each task [5]. To determine the inconsistent elements in corresponding comparison matrix an induced bias matrix [24] is designed, the induced matrix should be zero matrix if the comparison matrix is approximately consistent. Tasks are pairwise compared according to network bandwidth, complete time, cost of tasks, and reliability of task. Three major steps have been proposed to identify inconsistency among elements. All of the steps are not only used to identify inconsistency of elements when consistency ratio (CR) value of a comparison matrix is above 0.1 but also operateable when CR of a comparison matrix is less than 0.1. Two examples are illustrated showing that the weights are changed with distinct values of CR which are less than 0.1. This approach highly targets inconsistency issue and oversights some other factors that are fundamental while achieving proficiency in cloud environment e.g finish time, complexity of algorithm etc.

*G. Enhanced Load Balancing Based Task Scheduling Techniques in Cloud*

*Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization:* Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, and Dan Wang proposed this algorithm. This scheduling technique utilizes the characteristic of Ant Colony Optimization to decrease the tasks executing computation time and manages workload on each virtual machine in the cloud [25]. It's a heuristic approach to conjunctional optimization problems. Performance evaluation implemented on Load Balancing Ant Colony Optimization (LBACO) concealed by simulation platform with 10 data centers and 100-500 tasks, the comparison results of LBACO with First Come First Serve (FCFS) and Ant Colony Optimization ACO) showed better output as it did minimize the makespan and balance load of the entire system. The inadequacy with this algorithm are the assumptions that had been made during design of this algorithm i-e it has been assumed that tasks are self-dependent, have precedence constraint between tasks, and lastly the tasks are non-preemptive, and which are not practical in cloud environment. Moreover the heterogeneous processing of tasks is not entertained in this algorithm [25].

- *Bar: An Efficient Data Locality Driven Task Scheduling Algorithm For Cloud Computing:* This algorithm was presented by Jiahui Jin, Junzhou Luo, Aibo Song, Fang Dong and Runqun Xiong. In this paper, a data locality driven task scheduling algorithm called BAlance-Reduce (BAR) is presented. Its time complexity is calculated as O(max{m+n,nlogn}*m) [26]. It focuses to give solutions to data locality, network state and cluster workload problems. BAR dynamically regulates the task data locality and cluster workload according to network capacity and precedes a comprehensive vision to schedule tasks. BAR enhances the data locality in poor networks environment. To check performance BAR is compared with some algorithms and simulation results illustrate its effectiveness than other approaches in data locality, network state and cluster workload perspective. As a cloud environment is dynamic in nature, machine failure is expected when cluster is overloaded, this algorithm has complexity issue while handling this problem.

- *A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing:* Yiqiu Fang , Fei Wang and Junwei Ge proposed this algorithm, based on load balancing approach in cloud computing. They evaluate two level task scheduling model [27]. Two journey strategies are applied; 1st is to transfer the virtual machine to a host and designate the supplementary resources for the machine, and the 2nd is to transfer the other virtual machines by the whole of that host for departing the complimentary resources for the virtual machine. In this algorithm the host with lightest load is scheduled to the virtual machine exclusively. It upsurges the resource employment and dynamic requirements of consumers. Cloud Simulator tool kit is used to simulate this algorithm compared with other algorithms. Results show the higher resource utilization and improved load balancing over the cloud. There are certain assumptions in this strategy; i-e tasks are independent and execution of tasks replication is also not entertained. Moreover, only task response time and the demand for resources are considered in this algorithm where bandwidth and cost a not considered.

### *H. Improved Fairness Based Task Scheduling Techniques in Cloud*

- *Job Scheduling Algorithm Based On Berger Model In Cloud Environment:* This paper was presented by Baomin Xu, Chunyan Zhao, Enzhao Hu, and Bin Hu [5]. In this paper Berger Model Theory on distributive justice evaluation function was proposed into the job scheduling algorithm in cloud computing. Its prime focus is on the interpretation of distributive justice to the dynamic tasks requests by different users for various services in cloud environment. For gauging Berger model with task scheduling strategy and resource mapping in cloud computing it is required to perpetuate task classification, fairness function definition of user tasks, the task and resource Berger Model Theory on distributive justice parameterization. User tasks are classified and established on Quality of Service (QoS) parameters which are completion time and bandwidth. Based on idea of Berger Model two fairness constraints are defined: 1) Task justice 2) System justice. General Expectations Constraint restricts to provide fairness among resources allocation to users tasks. Simulation test is performed in order to check the correctness of the proposed algorithm cloud platform. This algorithm is compared with some other scheduling algorithms and through analysis of experimental results it had been seen that it gives effective implementation of user tasks and improved fairness as compared to contrasting algorithms. This algorithm gives better fairness among users task and resources but while meeting all distributive justice constraints it gives complexity to overall system, moreover, this paper gives the empirical value rather than pure logical value of initial vector of general expectation. Furthermore, the accurate vector value can be achieved using fuzzy neural network of QoS but not presented in the paper.

Figure. 2 shows the hierarchal view of categories of scheduling algorithms in cloud computing environment.

## IV. PERFORMANCE EVALUATION

Different task scheduling algorithms in cloud computing with distinct factors and parameters have been discussed, their performance evaluation is shown in Table 1.

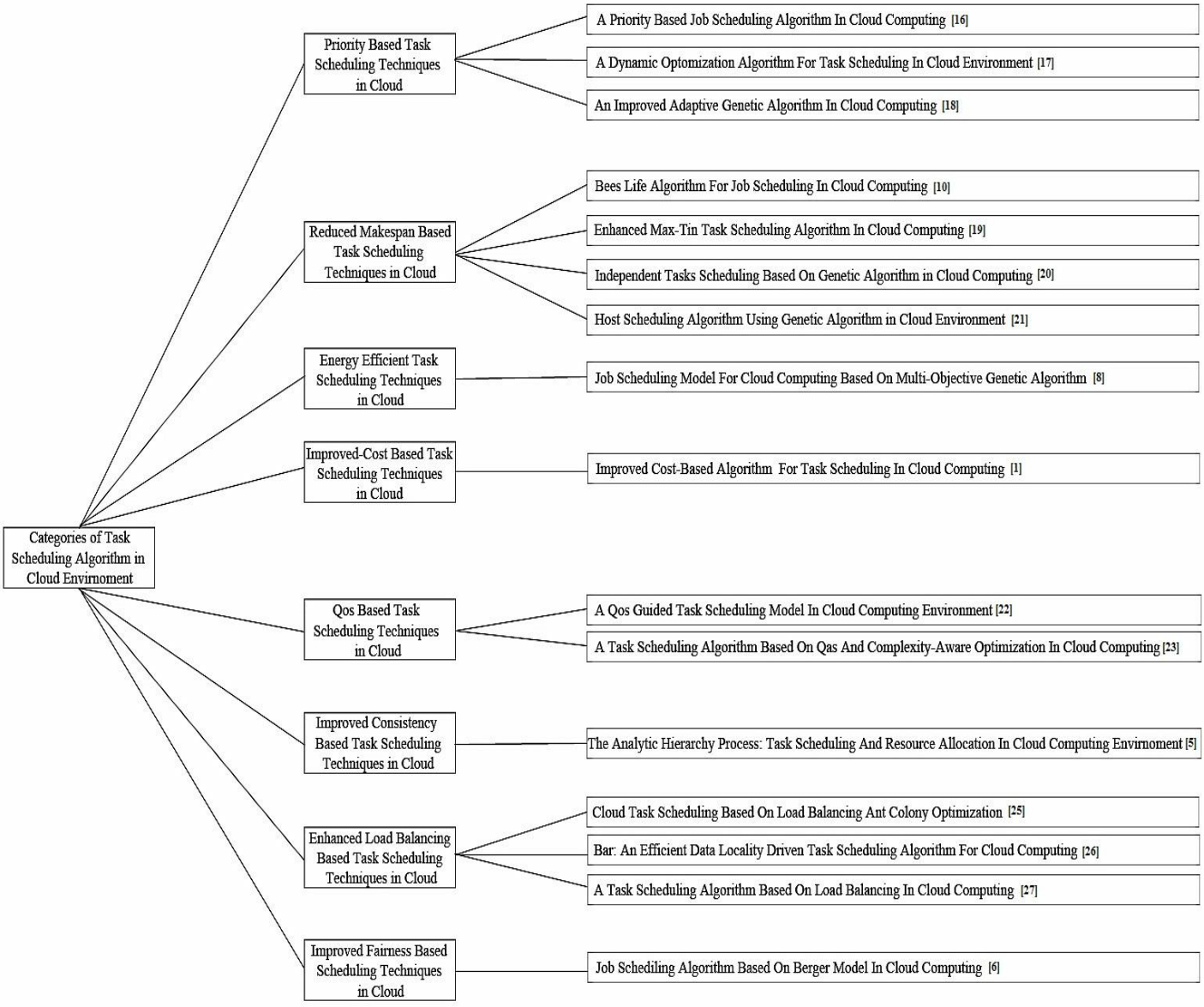


**Figure 2 Hierarchal view: Categories of task scheduling algorithms in cloud computing environment**

**Table 1 Performance evaluation of task scheduling algorithms in cloud computing environment**

| Scheduling Algorithm | Scheduling Category | Scheduling Mode | Scheduling Method | Scheduling Parameter | Algorithm Performance | Limitations |
|---|---|---|---|---|---|---|
| A Priority Based Job Scheduling Algorithm In Cloud Computing [16] | Priority Based Task Scheduling | Dependency mode | Priority using Analytical Hierarchy Process | Equitable priority model in cloud platform | Gives priority based job scheduling | Complexity issue, Doesn't gives optimal finish time and Inconsistency. |
| A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment [17] | Priority Based Task Scheduling | Batch mode | Task grouping, prioritization and Greedy allocation | Improved perfor-mance employing priorities to tasks based on group type | Gives improved cost and better completion time of tasks | Communication overhead |
| An Improved Adaptive Genetic Algorithm In Cloud Computing [18] | Priority Based Task Scheduling | Dynamic batch mode | Users' tasks are divided into sub-tasks, priority is assigned to each task and the sub-tasks inherit the priority of their parent task. | Problem-solving for task scheduling algorithm of emergency computing in cloud computing environment. | Faster conver-gence speed and run-over the drawback of traditional Adaptive Genetic Algorithm. | Complexity issue |
| Bees Life Algorithm For Job Scheduling In Cloud Computing [10] | Reduced Makespan Based Task Scheduling | Batch mode | A novel population-based metaheuristic approach | Efficiency and the performance in terms of the execution time | Shortened finish time | Unfairness among tasks |
| Enhanced Max-Min Task Scheduling Algorithm In Cloud Computing [19] | Reduced Makespan Based Task Scheduling | Online mode | Modification of Improved Max-min task scheduling approach | Minimum makespan, load-balancing, completion-time, Service-time. | Reduced makespan and load balance across resources. | Unfair resource allocation among tasks |
| Independent Tasks Scheduling Based On Genetic Algorithm In Cloud Computing [20] | Reduced Makespan Based Task Scheduling | Dynamic batch mode | Service Oriented Architecture, principle of Genetic Algorithm | Dynamic Scheduling in heterogeneous system to reduced makespan | Optimal time and resource utilization | Large solution space in Genetic Algorithm |
| Host Scheduling Algorithm Using Genetic Algorithm In Cloud Computing Environment [21] | Reduced Makespan Based Task Scheduling | Batch mode | Principle of Genetic Algorithm companioned with Suffrage Heuristic | Minimized makespan and enhanced efficiency | Gives better initial population, improved results then generic Genetic Algorithm | It used the fixed number of virtual machines during experiments. |
| Job Scheduling Model For Cloud Computing Based On Multi-Objective Genetic Algorithm [8] | Energy Efficient Task Scheduling | Batch mode | A macroscopic scheduling model with cognition and decision components | Minimize energy consumption and maximize the profit of service provides under the constraint of deadlines | Gives high profits and consumes less energy. | Gives worse results at higher arrival rates and number of failed applications increases. |
| Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing [1] | Improved Cost Based Task Scheduling | Batch Mode | Three priority lists are defined and grouping algorithm is applied to each list | Minimize total tasks completion time and minimize cost | Better cost minimization results then ABC algorithm. | This algorithm scenario showed limited cloud platform |
| A Qos Guided Task Scheduling Model In Cloud Computing Environment [22] | Qos Based Task Scheduling | Batch Mode | Sufferage heuristic algorithm, Min-min Algorithm, Qos guided scheduling strategies | QoS requirements, dynamic resource properties and low execution complexity | Better QoS and shortened completion time | Starvation of the task groups with low QoS requests |
| A Task Scheduling Algorithm Based On Qos And Complexity-Aware Optimization In Cloud Computing [23] | Qos Based Task Scheduling | Online mode | Fuzzy identification method | Higher quality of service by the lower service cost | Improve service cost, node's load rate, bandwidth utilization, and network delay | Insufficient of storage space and computation complexity |
| The Analytic Hierarchy Process: Task Scheduling And Resource Allocation In Cloud Computing Environment [5] | Improved Consist-ency Based Task Scheduling | Batch mode | Analytical Hierarchy Process using induced comparison matrix | Improved consistency of comparison matrixes | Gives better approach to handle inconsistency | Over sighted finish time and complexity of algorithm. |
| Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization [25] | Enhanced Load Balancing Based Task Scheduling | Batch mode | Heuristic approach to conjunctional optimization problems | Load Balancing, decrease execution time of tasks | Decreased tasks computation time and manages workload on each virtual machine | Certain assumptions had been made in this algorithm design which are not practical in cloud |
| Bar: An Efficient Data Locality Driven Task Scheduling Algorithm For Cloud Computing [26] | Enhanced Load Balancing Based Task Scheduling | Online mode | Data locality driven task scheduling algorithm BAlance-Reduce(BAR) | Data locality, network state and cluster workload problems | Enhanced data locality in poor networks environment | Complexity issue while handling machine failure |
| A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing [27] | Enhanced Load Balancing Based Task Scheduling | Dependency mode | Two level task scheduling model | Enhanced load balancing over dynamic requests | Gives higher resource utilization and improved load balancing | Bandwidth and cost factors are ignored |
| Job Scheduling Algorithm Based On Berger Model In Cloud Environment [6] | Improved Fairness Based Task Scheduling | Batch Mode | Berger Model Theory on distributive justice evaluation function by task classification and fairness function definition | Distributive justice to the dynamic tasks requests. | Effective implementation of user tasks and improved fairness. | 1. Complexity<br>2. Only considered empirical values of initial vector of general expectation. |

## V. Conclusion

In this survey paper, various task scheduling algorithms in cloud computing environment based on distinguishable scheduling parameters have been analyzed. Different scheduling algorithms work on distinct scheduling criteria, all algorithms are efficient in one way or another. These existing algorithms showed enhanced load balancing, minimized make span, energy efficiency, quality of service, consistency, maximum resource utilization, effective implementation, fairness among tasks, high profits and bandwidth utilization over cloud but not all at the same time. There are also some short comings in every scheduling algorithm none of them achieves 100% efficiency. Hence there is a need of some task scheduling algorithm which takes into account all above scheduling parameters to attain maximum efficiency, better tasks management, and proficiency and enhanced profits to cloud service providers.

## References

[1] S. Sadhasivam, "Improved Cost-Based Algorithm For Task Scheduling In," 2010.

[2] A. Agarwal and S. Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment," vol. 9, no. 7, pp. 344–349, 2014.

[3] W. Tan, Y. Sun, G. Lu, A. Tang, and L. Cui, "Trust Services-Oriented Multi-Objects Workflow Scheduling Model for Cloud Computing *," no. 61272036, pp. 617–630, 2013.

[4] C. Science and S. Engineering, "A Survey of Proposed Job Scheduling Algorithms in Cloud Computing Environment," vol. 3, no. 11, pp. 782–790, 2013.

[5] D. Ergu, G. Kou, Y. Peng, Y. Shi, and Y. Shi, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment," *J. Supercomput.*, vol. 64, no. 3, pp. 835–848, May 2011.

[6] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Adv. Eng. Softw.*, vol. 42, no. 7, pp. 419–425, Jul. 2011.

[7] A. V Karthick, E. Ramaraj, and R. G. Subramanian, "An Efficient Multi Queue Job Scheduling for Cloud Computing," 2014.

[8] J. Liu, X. Luo, X. Zhang, F. Zhang, and B. Li, "Job Scheduling Model for Cloud Computing Based on Multi- Objective Genetic Algorithm," vol. 10, no. 1, pp. 134–139, 2013.

[9] A. H. Gade, "A Survey paper on Cloud Computing and its effective utilization with Virtualization," vol. 4, no. 12, 2013.

[10] S. Bitam, "Bees Life Algorithm for Job Scheduling in Cloud Computing," pp. 186–191, 2012.

[11] L. K. Arya and A. Verma, "Workflow scheduling algorithms in cloud environment - A survey," *2014 Recent Adv. Eng. Comput. Sci.*, pp. 1–4, Mar. 2014.

[12] S. Patel and U. Bhoi, "Priority Based Job Scheduling Techniques In Cloud Computing : A Systematic Review," vol. 2, no. 11, 2013.

[13] A. Pawar, M. T. Scholar, and P. D. Kapgate, "A Review on Virtual Machine Scheduling in Cloud Computing," vol. 3, no. 4, pp. 928–933, 2014.

[14] IBM Corporation and I. Corporation, "Mainframes working after hours: Batch processing," *Mainframe concepts*.

[15] "What is pre-emptive and non-preemptive scheduling?" [Online]. Available: http://www.careerride.com/OS-preemptive-scheduling.aspx. [Accessed: 07-May-2014].

[16] S. Ghanbari and M. Othman, "A Priority based Job Scheduling Algorithm in Cloud Computing," vol. 50, no. Icasce, pp. 778–785, 2012.

[17] M. Choudhary, "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment Sateesh Kumar Peddoju," vol. 2, no. 3, pp. 2564–2568, 2012.

[18] B. Hu, X. Sun, Y. Li, and H. Sun, "An Improved Adaptive Genetic Algorithm in Cloud Computing," *2012 13th Int. Conf. Parallel Distrib. Comput. Appl. Technol.*, pp. 294–297, Dec. 2012.

[19] U. Bhoi, P. N. Ramanuj, and W. S. Email, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing," vol. 2, no. 4, pp. 259–264, 2013.

[20] C. Zhao, S. Zhang, Q. Liu, J. Xie, and J. Hu, "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing," *2009 5th Int. Conf. Wirel. Commun. Netw. Mob. Comput.*, pp. 1–4, Sep. 2009.

[21] T. Goyal and A. Agrawal, "Host Scheduling Algorithm Using Genetic Algorithm In Cloud Computing Environment," vol. 1, no. 1, pp. 7–12, 2013.

[22] H. Han, Q. Deyui, W. Zheng, and F. Bin, "A Qos Guided Task Scheduling Model in Cloud Computing Environment," *2013 Fourth Int. Conf. Emerg. Intell. Data Web Technol.*, pp. 72–76, Sep. 2013.

[23] W. Ning, Y. Yang, M. Kun, C. Yu, and D. Hao, "A task scheduling algorithm based on QOS and complexity-aware optimization in cloud computing," *Natl. Dr. Acad. Forum Inf. Commun. Technol. 2013*, pp. 5–5, 2013.

[24] D. Ergu, G. Kou, Y. Peng, and Y. Shi, "A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP," *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 246–259, Aug. 2011.

[25] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization," *2011 Sixth Annu. Chinagrid Conf.*, pp. 3–9, Aug. 2011.

[26] J. Jin, J. Luo, A. Song, F. Dong, and R. Xiong, "BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing," *2011 11th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput.*, pp. 295–304, May 2011.

[27] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing," pp. 271–277, 2010.