Volume 8, No. 3, March – April 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
REVIEW ARTICLE
Available Online at www.ijarcs.info

# Task Scheduling in Cloud Computing

Vivek Manglani, Abhilasha Jain and Prof. VivekPrasad
Institute of Technology
Nirma University,
Ahmadabad, Gujarat, India

***Abstract*:** Cloud computing is considered to be a buzzword in today's IT industry with the help of which users can get access to software, hardware, applications, platform by the means of just an internet connection. It is based on the concept of utility computing wherein customer has to pay as per the use. The necessary requirement in the world of cloud computing is the scheduling of tasks under some restrictions. The job of the scheduler is to take care of efficiency of the algorithm while maintaining the Quality of service parameters. However there is no single algorithm existing which performs better in terms of all Quality of service parameters which is why in this paper we will be describing some popular scheduling algorithms. Various scheduling algorithms discussed in this paper are Balance-Reduce, Dynamic priority based, cost-deadline based and Genetic algorithm.

***Keywords:*** cloud, computing, algorithm, software

## *1.* INTRODUCTION

Cloud computing has become one of the most important service which is being utilized by both technical and non-technical people. Non-technical users who are not even aware of the concepts of cloud computing are taking advantage of the cloud services by making their computational tasks handled by the cloud. In order to gain the least execution time, the development of the Cloud applications are executed in parallel. Cloud can be either public or private. In public cloud, the data centres are managed and monitored by a third party which is why security turns out to be a major issue. In private cloud security is not compromised since data centre is within an organization.

As the number of cloud users is increasing with time, the need for providing a quality service has become of great importance. In this aspect, the notion of scheduling becomes an important task which is to be provided to the user. Scheduling refers to the execution sequence of the tasks which are allocated to machines. By the means of scheduling, one of the most important quality of service, i.e. latency, is minimized.[1] The cloud cannot restrict its working like the conventional methods (Distributed Systems) do that is why it is dynamic in nature. This implies that the scheduling algorithms should also be dynamic and not static. The dynamic scheduling differs from static in a way that the prior information regarding the incoming tasks and the number of resources available for them, is known.

Scheduling can be implemented using two methods- one is by scheduling the VM and the other is by scheduling the host.[1] The latter one is implemented by scheduling the number of VMs considering the requirements of the host. The scheduling by VM will depend on the incoming tasks and the VM assigned to the user serves as an input container for their data.

In this paper, the concept of task scheduling and its classification is discussed along with the comparative study of various task scheduling algorithms on the basis of various quality of service parameters. Later, the advantages and disadvantages of these scheduling algorithms are also mentioned.

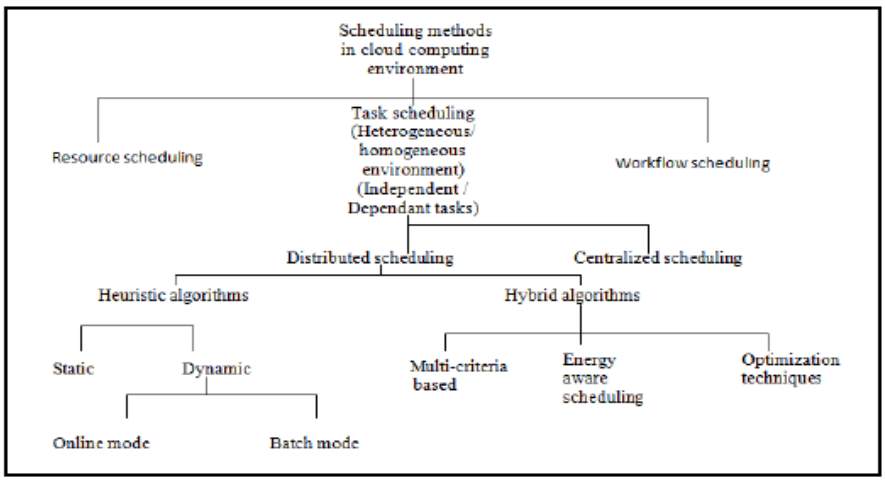


Fig. 1. Classification of Scheduling Methods in Cloud Computing Environment

## 2. CLASSIFICATION OF SCHEDULING:

Scheduling algorithms can be categorized into three aspects: workflow scheduling, task scheduling and resource scheduling. In this paper, the prime focus is on task scheduling. Resource Scheduling means allocation of resources to different Virtual Machines. Workflow Scheduling refers to the scheduling of workflows which will constitute a task, if executed in a specific order.[2] The tasks, thus obtained are sequenced using task scheduling algorithms. Task scheduling is further divided into distributed or centralized. In distributed scheduling, partitioning of scheduling is done between different schedulers while in centralized scheduling, a single scheduler is responsible for mapping the resources. Task scheduling can be applied on independent or dependent tasks in either homogeneous or non-homogeneous environment.

Centralized scheduling is simple to implement but it lacks fault tolerance and scalability since it is highly dependent on single server. While in distributed scheduling, the load is divided among various nodes resulting in reduced number of processing cycles. Despite this advantage, the distributed architecture is complex and difficult to implement.

Distributed scheduling is divided into heuristic and hybrid scheduling.[2] Heuristic method can be further categorized as static or dynamic. The static scheduling can be applied only when we know the information about the tasks in advance whereas dynamic scheduling is used for executing tasks in an instant. Although the performance of dynamic scheduling is better than static, overhead of static algorithms is less since in dynamic algorithms there is a need of change in system's information instantaneously. There are three types of hybrid algorithms: Energy aware scheduling, minimization-maximization approach and multi-objective methods.

## 3. WHAT IS TASK SCHEDULING?

Resource allocation and task scheduling are the most important aspects of the cloud environment as they directly affect the system's performance. Task scheduling focuses on assigning tasks to available resources at a particular instant of time.[3] The incoming tasks to the cloud has to be run timely using the available resources in order to attain efficiency, possible minimum time, lesser makespan and proper utilization ofresources which calls for the need of an effective task scheduling algorithm. The performance of the system is determined by the algorithm chosen.

Task scheduling is a NP complete problem. It is the one of the prime processes of IAAS(Infrastructure as a service). Virtual machines are considered to be scheduling machines when task scheduling is being done. The ultimate goal of task scheduling is to balance the load on processors in regards of network bandwidth.

### 3.1 NON-LINEAR PROGRAMMING MODEL:

In this algorithm we will be considering 'n' number of computer hosts h1,h2…hn and each of them are allocated with virtual machines along with VMM(Virtual Machine Monitor) to control and monitor them.[3] There will be 'm' number of virtual machines each having computational power. The time required for each task is t1,t2…tn. In this model, virtual machines are assigned a correct number of tasks depending upon the network bandwidth available.
Fig 2 shows an effective algorithm for task scheduling which, on the basis of computational power of VMs, bandwidth available and resources, allocate tasks to minimize the waiting time. The steps involved are:

1. Give in tasks to VMs
2. Allocating bandwidth to tasks
3. Allocating bandwidth to VMs
4. Building a non-linear programming model
5. Using the output from the model, tasks are assigned to each VM
6. Now, tasks are present in a queue
7. Idle virtual machines are searched by tasks
8. Required bandwidth is then checked by tasks
9. Finally, the tasks are executed

## 4. SCHEDULING ALGORITHMS:

### 4.1 Balance-Reduce Algorithm:

In this task scheduling algorithm, scheduling is done on data locality.[4] After finding a suitable solution, the problem faced is that remote task number is required for calculating remote cost. Since the remote cost is changing quite frequently, it is difficult to find an optimal solution. For instance, after a remote task is allocated, its number as well as cost increases by one. Also updating servers which were assigned remote tasks is required. This algorithm is divided into two phases: Balance and Reduce.

**Balance**: The inputs given to this phase are Linit (initial load set), Cloc (Local cost) and G(data placement graph) which gives the output**B** (total allocation). Even distribution of tasks to their favourable servers is under **B**.

**Reduce**: Here the inputs are Cloc, B which was computed in the balance phase, a remote cost function Crem () and Linit. In iterative manner, this phase results in a collection of total allocations and the best one is returned. The remote cost is computed prior to every iteration since we already have **B**.

### 4.2 Dynamic priority based scheduling algorithm:

In this algorithm scheduling of virtual machines is done based on priority given to the nodes dynamically.[5] VMs are assigned to the nodes based on their priority which changes dynamically depending on their load factor and capacity. With this concept of dynamic priority, utilization of resources is done efficiently. This algorithm help in achieving power efficiency and better performance.

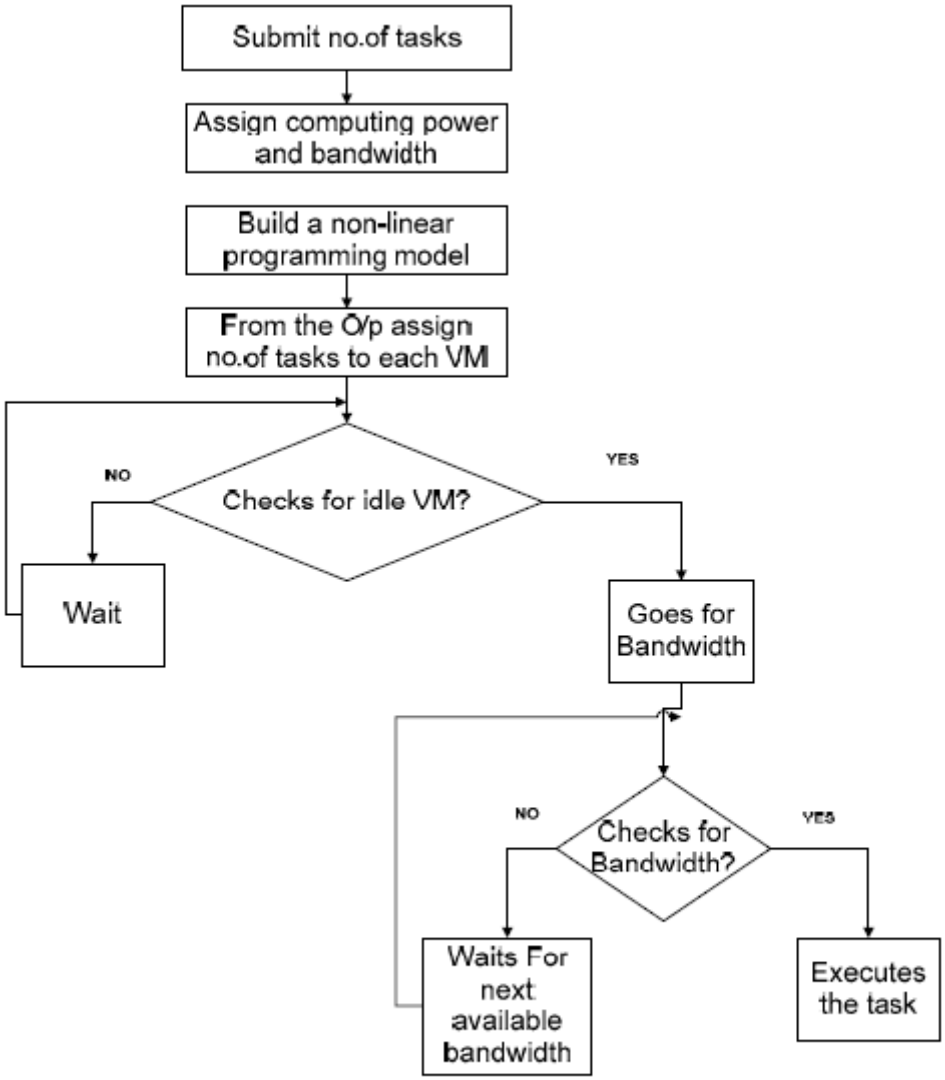


Fig 2 : Showing task scheduling to VM's

**WORKING OF ALGORITHM**

When VMs are requested to be scheduled, it sees to it whether the node with maximum resource has been already identified and prioritized. In case the node is not identified then it is done now. The maximum resource node is checked for its load factor (should be less than 80%. This is done so that the node is not overloaded. After a preferable node is identified, it is given the highest priority and assigned with the VM. If the load factor is above 80% than the node with second maximum resource is checked. If it is identified before and has a load factor less than 80%, then the VM is scheduled on it and node with next maximum resource is searched, which has load factor less than 80%. Firstly the current maximum resource node is given to previous maximum resource node and the previous maximum node is given the next highest priority. After this highest priority is given to the new node. In order to prevent the variations in load factor in some cases, the track of previous maximum resource node, is kept.

If the node having highest priority has load factor greater than 80%, then it continuously assign the VMs to the node with second highest priority until its load factor is increased to 80%. Doing this, we will buy some time for the node with highest priority to complete its work and shutdown the VM and hence it will become less than the load factor. Even after this process, the load factor of the node remains above 80% then its priority is reduced and another node is selected for assigning higher priority.

The performance of this algorithm is extraordinary in regards of time required for scheduling a VM on a node, after all the nodes have been identified. The identified nodes would receive scheduling directly while other nodes are not considered. The idle nodes (nodes which have no VM allocated.) are also checked when the request for scheduling is made.

4.3**Cost-Deadline Based Scheduling algorithm:**

This algorithm considers two parameters: cost and deadline.Cloudlets are scheduled based on these parameters only,[6] The cost of the algorithm is computed on the basis of length of the cloudlets and their deadline. The order of the cloudlet is decided after applying sorting.A min-min algorithm is applied for mapping of cloudlets with virtual

machines. Hence, this algorithm defines the cost in user as well as provider's point of view also minimizes the skipped deadline.

ALGORITHM:
START

step1: Define cloudlet length Lj and deadline Dj for every task Tj.

step2: Compute UserMaxPay Uj for every Tj by using the given equation:

Uj = k*Lj/Dj

step3: All the task Tj which are accepted are placed in a queue Q.

step4: Depending on the earliest deadline first, sort the queue.

step5: In case two task have same deadline then sort the queue according to UserMaxPay Uj.

step6: Use min-min algorithm for mapping tasks to VMs.

step7: Use Space-shared policy for task scheduling.

step8: Compute the net profit NPj using a model which is depending on a parameter called Lateness G given as:-

IF A<0
NPj = Uj – PerSecCost*RealCpuTime
Else
NPj = Uj*(1-0.01*A) – PerSecCost*RealCpuTime

Return NPj
End

**4.4 Genetic Algorithm:**
The GA is based on the biological concept of producing population.[7] The term from Charles Darwin's theory 'survival of the fittest' is used as the basis for scheduling in which resources are assigned with tasks depending on the fittest function value for every parameter in the process. The basic principles of the algorithm are:

***1) INTITIAL POPULATION***:
This refers to the set of individuals used in the algorithm for finding out the best possible solution (individual). The individuals from initial population are selected and some computations are made on them to construct the next generation.

***2) FITNESS FUNCTION***:
It is a measure which demonstrates the performance of a solution in the population. The value of this function determines whether an individual will survive or die.

***3) SELECTION***:
This method is used for selecting a solution for the next generation according to the Darwin's law of survival. Various selection strategies can be applied for selecting the best chromosomes like Boltzmann strategy, selection based on rank or tournament selection.

***4) CROSSOVER***:
This operation proceeds by selecting two individual (parents) and then creating their offspring by changing the properties of those parents.

***5) MUTATION***:
This method is applied after crossover, whenever the population has become very homogeneous. It works by altering one or more values of the gene in chromosome. This results in an entirely new gene getting added.

**TOURNAMENT SELECTION GENETIC ALGORITHM:**

The prime objective of this algorithm is that after the selection process we might have a solution which is satisfying the fitness function but we will not pass it on to crossover process. The solution is not deleted from the population rather added to the population in the beginning of next iteration. Some iterations can result in the optimal solution which is why this step is regarded as a good step.

***1) INTITIAL POPULATION***:
Binary representation is used for randomly generating the population. The solutions in task scheduling are represented using VM ID and ID for every task which is to be run on those VMs. The tasks and associated VMs are encoded in a binary bit.

***2) FITNESS FUNCTION***:
The motive of task scheduling is to minimize the completion time for all tasks. The completion time CTij of task Ti on VMj is defined as:

$$CompTime = Cmax[i,j]$$

$$j \in VM, j = 1,2,3,...m$$

$$i \in T, i = 1,2,3,...n$$

Where m, n and CompTime represents number of virtual machines, number of tasks and completion time respectively.

The processing time Pi for very task can be calculated if the processing speed Sj of every VMj is known, is using the following equation:

$$P_{ij} = C_i / S_j$$

Where Ci represents task Pi's computational complexity and Pij represents the processing time for the task Ti.
The collective processing time for every task is calculated using the equation:

$$P_j = \sum_{i=1}^{n} P_{ij}$$

***3) SELECTION***:
First of all, two individuals from population are chosen at random. After this a random number Rn is selected between 0 and 1. If Rn < m (where m refers to a parameter), then the fitter of the two solutions is chosen as the parent else the less fit solution is chosen. The

solution which is not chosen is added back to the initial population so that it could be chosen again.

***4) CROSSOVER***:
This method is slightly different from the one defined in genetic algorithm as the two chromosomes which are chosen for crossover process are considered as offspring. Hence the crossover generates 4 children out of which 2 best are selected.

***5) INITIALIZE SUBPOPULATION***:
This is an added method be the TS-GA algorithm to create a diversified population. Here subpopulations (created after crossover) are added to old populations (parents) after every iteration.

***6) KEEP BEST SOLUTION***:
Their might be a solution which satisfies fitness function but is not chosen in crossover process. The solution is not deleted from the population rather added to the population in the beginning of next iteration. Some iterations can result in the optimal solution which is why this step is regarded as a good step.

## 5. CONCLUSION

Effective scheduling algorithms have a crucial rule in the performance of cloud computing. We can develop an improvised scheduling algorithm by enhancing the existing scheduling methods. The table created below combines the scheduling algorithms that we have studied.

| Scheduling Algorithm | Parameters involved | Advantages | Disadvantages |
|---|---|---|---|
| Balance-Reduce algorithm | Remote cost | Data available locally is considered | Can't find optimal solution, Remote task number is required |
| Dynamic priority based algorithm | Priority of task, Load factor | Power efficiency and better performance | Consistency, Makespan, Complexity |
| Cost-deadline based algorithm | Cost, deadline | Efficient mapping of cloudlets, resource cost and skipped deadline are minimized | Load balancing problem, cost of regeneration of dataset is not considered. |
| Genetic algorithm | Efficiency, Makespan, Optimization, Performance | Better efficiency and performance in regards of makespan | Time consuming and complex |

Fig 3: Comparative study of scheduling algorithms

## 6. REFERENCES

[1] Sujan, S., and R. Kanniga Devi. "A batchmode dynamic schedulingscheme for cloud computing." Communication Technologies (GCCT),2015 Global Conference on. IEEE, 2015.

[2] Mathew, Teena, K. Chandra Sekaran, and John Jose. "Study andanalysis of various task scheduling algorithms in the cloud computingenvironment." Advances in Computing, Communications and Infor-matics (ICACCI, 2014 International Conference on. IEEE, 2014

[3] Mittal, Shubham, and Avita Katal. "An Optimized Task SchedulingAlgorithm in Cloud Computing." Advanced Computing (IACC), 2016IEEE 6th International Conference on. IEEE, 2016.

[4] Thawari, Vaishali W., Sachin D. Babar, and Nitin A. Dhawas. "Anefficient data locality driven task scheduling algorithm for cloudcomputing." International Journal in Multidisciplinary and AcademicResearch (SSIJMAR) 1.3 (2012).

[5] Subramanian, S., et al."An adaptive algorithm for dynamic prioritybased virtual machine scheduling in cloud." International Journal ofComputer Science Issues(IJCSI) 9.6 (2012).

[6] Sidhu, Harmanbir Singh. "Cost-Deadline Based Task Scheduling inCloud Computing." Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on. IEEE,2015.

[7] Hamad, Safwat A., and Fatma A. Omara. "Genetic-Based TaskScheduling Algorithm in Cloud Computing Environment." International Journal of Advanced Computer Science & Applications 1.7(2016): 550-556.