

A Survey on Scheduling Algorithms in Cloud Computing

Maheswari.R¹,

¹ PG Scholar, Department of CSE,
Angel College of Engineering and
Technology, India

S.Selvi²

² Asst. Professor, Department of CSE,
Angel College of Engineering and
Technology, India

Abstract

Many organizations are increasingly turning to the cloud because it provides ease of access to resources, cost-effective mechanism, enhanced collaboration, limitless flexibility, security etc., of running resource-intensive applications. Cloud systems are very economical and useful for businesses of all sizes. Resource scheduling and task/job scheduling are the most important confront to be considered in cloud computing. So that the resources must be allocated efficiently to all the jobs submitted to it. In this paper we make a survey as characteristics of various algorithms and provide comparison table by considering various parameters like Cost, Budget Constraints, efficiency etc.,

“A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption” [2].

Cloud computing provides different computing resources and made accessible over the internet to remote user as services. The projected benefits of cloud computing are very compelling both from a cloud consumer as well as a cloud services provider perspective.

Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) are the basic fundamental services of cloud. The other services like Data as a Service (DaaS), Network as a Service (NaaS), Everything as a Service (XaaS) etc., are all provided by the cloud. The main services descriptions are given below:

1. Software as a Service (SaaS): In this model, cloud user can be able to access the software applications and storages. Multiple end users can access one instance at a time which is running in the cloud. In customer side there is no need of any investment for software's licenses or servers.

Example: Xactly, Salesforce.com, NetDocuments, Zoho, Google Apps., eVapt, etc.,

2. Platform as a Service (Paas): In this layer the software platform or software development environment is provided as a service for the cloud users. The cloud users can develop their own software using cloud platform without any cost and complexity of buying and managing the hardware's and software's.

Example: Infobright, Google BigTable, Amazon simpleDB, Force.com etc.,

3. Infrastructure as a Service (Iaas): This layer provides its network, storage, capacity and computing resources as a service. The cloud user can give access to the virtualized components to build their own IT platforms.

Example: Amazon S3, Nirvanix, Terramark, Rightscale, App Nexus etc.,

I. INTRODUCTION

Cloud Computing appears as a new paradigm for managing, deploying and contribute services through a shared infrastructure. The term “Cloud” is the default symbol of internet in diagrams and “Computing” which means computation, coordination logic, Storages etc. Cloud computing environment serves and satisfies user requirements as Pay-Per use basis. Cloud computing becomes visible in a way that provides on-demand resources to the users, so as to precede locally available computational power, delivering new computing resources whenever necessary. With the use of computing resources (hardware and software) that are delivered as a service over a network, it provisioned and released with minimal management effort; dynamically scalable infrastructure for computation, application hosting, content storage and delivery The basic idea of cloud computing is that reuse of IT facilities, ease of organize the services with very low expenses and constant operational expenses leading to variable pricing schemes and reduced opportunity costs. It leverages the economies for both services providers and users of the cloud.

Forrester defines cloud computing as:

SCHEDULING

Scheduling is that allocating resources to the needed jobs, allocating resources according to the budget constraints, etc., in cloud environment. There are many types of scheduling algorithms available in cloud computing. To achieve high performance, efficient use of resources, best system throughput, budget constraints, Quality of Service (QoS) etc., should be considered. Job scheduling algorithms in cloud computing can be categorized into two main groups; Batch Mode Heuristic scheduling Algorithms (BMHA) and Online Mode Heuristic Algorithms (OMHA).

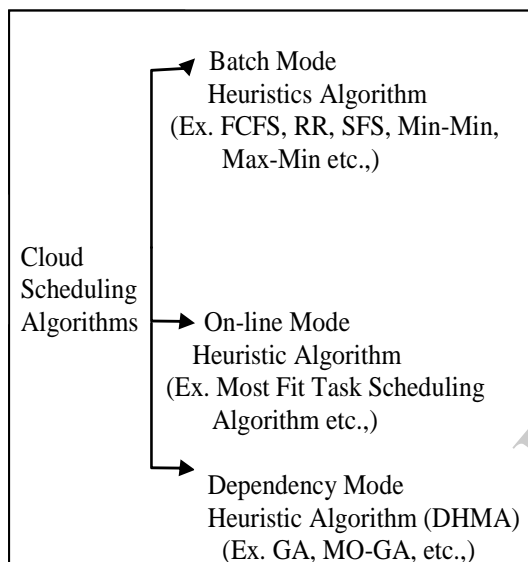


Figure 1. Scheduling algorithm classification

In Batch mode heuristic algorithm, Jobs are queued and collected into a set when they arrive in the system. The scheduling process starts only after a fixed period of time. By OMHA jobs are scheduled immediately they arrive to the system. By DMHA jobs or resources processing depends upon the previous one.

II. SCHEDULING ALGORITHMS

a.) Dynamic Resource Provisioning for Deadline and Budget Constrained Application in Cloud Environment (DRPDBCA) [3]

In this paper, Naimisha S Trivedi ,Devyaniba S Chudasama developed Dynamic Resource Provisioning Algorithm by making cost and deadline as a constraints. It is able to maximize the no. of applications execution under budget and deadlines. Also the author considered the problem of scheduling

prioritized jobs of application with the limit of budget and deadline constraints.

Dynamic Resource Provisioning Algorithm

1. Evaluate the VM's using Number of VM

$$= \frac{TotalBudget}{(Deadlineofjob * MarketPriceofresource)}$$
2. Assign the set of VM processing as V_r and VM completed as V_c .
3. Assign V_t as set of VM to be terminated
4. Assign V_{max} as Max No. of VMs
5. Calculate Budget left = Total Budget – Consumed Budget
6. If the Budget left is less than price of VM's then
7. Calculate the VM to be terminated as $(VM_t) = |V_r| - (Remaining\ Budget/price)$
8. $V_t = \text{select } VM_t \text{ from } V_c$
9. TERMINATE(V_t)
10. Else
11. U as Current VM utilization
12. If Current Utilization > Upper Utilization Threshold and $V_r < V_{max} * \text{Number of VM}$
13. START(new VM)
14. Else
15. $V_i = \text{Set of idle VM's}$
16. $VM_t = |V_i|/2$
 $V_t = \text{select } nT \text{ VMs from } V_i$
 TERMINATE (V_t)
 End
 End

Priority-based scheduling algorithm

1. procedure SCHEDULE
2. $P = \text{empty priority queue}$
3. IdleVMs = set of idle VMs
4. for root task t in all workflows do
5. INSERT(t ; P)
6. end for
7. while deadline not reached do
8. while IdleVMs $\neq 0$; and $P \neq 0$; do
9. $v = \text{SELECTRANDOM}(IdleVMs)$
10. $t = \text{POP}(P)$
11. SUBMIT(t ; v)
12. end while
13. Wait for task t to finish on VM v
14. Update P with ready children of t
15. INSERT(v ; IdleVMs)
16. end while
17. end procedure

This algorithm which consists of two procedures they are:

- *Provisioning procedure* – Based on resource utilization.
- *Scheduling Procedure* –Based on task scheduling.

It schedules the tasks at runtime. It starts its process by finding number of resources depending upon time and budget available. In order to schedule individual tasks onto available VMs, algorithm uses the dynamic cum priority-based scheduling procedure [3]. Initially, the ready tasks are added to a priority queue based on their priority. If there are idle VMs available, and the priority queue is not empty, the next task from the priority queue is submitted to an arbitrarily chosen idle VM. The process is repeated until either there are no idle VMs or the priority queue is empty. The scheduler then waits for a task to finish, add its ready children to the priority queue, marks the VM as idle, and the entire process repeats until the deadline is reached. Algorithm ensures that tasks from lower priority workflows are always deferred when higher-priority tasks are available, but lower-priority tasks can still occupy idle VMs when higher priority tasks are not available.

b.) Optimal Resource Provisioning (ORP) for cloud computing Environment [4]

In this paper the SaaS provider leases resources from cloud providers and also leases software as services to SaaS users. The SaaS providers aim at minimizing the payment of using VMs from cloud providers, and want to maximize the profit earned through serving the SaaS users' requests. The SaaS users purpose to obtain the optimized QoS to accomplish their jobs with a limited budget and deadline. The proposed optimal cloud resource provisioning algorithm includes two sub-algorithms at different levels: interaction between the SaaS user and SaaS provider at the application layer and interaction between the SaaS provider and cloud resource provider at the resource layer. Optimal cloud resource provisioning algorithm (OCRP) is an economic based algorithm, which includes optimization at resource and application level.

Algorithm: Optimization algorithm for SaaS provider and SaaS user

SaaS USER

Input: $sp_m^{(n)}$ the price of the SaaS provider m

Step1: SaaS user calculates optimal payments $up_i^{m*(n+1)}$ for the SaaS provider m

to maximize $U_{su}(up_i^m)$ based on $sp_m^{(n)}$

$$up_i^m = \text{Max}\{U_{su}(up_i^m)\}$$

Step2: SaaS user communicates new optimal payments $up_i^{m*(n+1)}$ to the SaaS provider m .

Output: $up_i^{m*(n+1)}$ to SaaS provider m

SaaS PROVIDER

Input: $up_i^{m*(n+1)}$ from SaaS user i

Step1: Calculates the optimal SaaS service to the SaaS user x_i^{m*} to maximize $U_{sp}(x_i^m)$ based on the payments $up_i^{m*(n+1)}$

$$x_i^{m*} = \text{Max}\{U_{sp}(x_i^m)\}$$

Step2: Computes a new SaaS service price according to the following formula:

$$sp_m^{(n+1)} = \text{Max}\left\{\varepsilon, sp_m^{(n)} + \eta\left(\sum_i x_i^m - S_m\right)\right\}$$

where $\eta > 0$ is a small step size parameter. If the demand for SaaS service exceeds the SaaS provider's service supply S_m , then the price $sp_m^{(n+1)}$ is raised; otherwise, the SaaS service's price is reduced; **Output:** Communicates a new SaaS service's price $sp_m^{(n+1)}$ to all SaaS users.

c.) A Synthetic Heuristic Algorithm for Independent Task Scheduling (SHIS) in Cloud Systems[5]

In this paper Arash Ghorbannia, Delavar, Yalda Aryan proposed Synthetic Heuristics Algorithm for Independent Task Scheduling (SHIS), with goal oriented operations such as, making initial population, dual step evaluation and running the tasks by considering load balancing and quality of service, achieves the optimize makespan. It also decreases the probability of task failure rate on running, based on the resource failure frequency rate, and also decreases the task starvation problem. It supports the scheduling of new entered tasks in system by a dynamic method. The SHIS proposed algorithm, in addition to parameters such as resource access rate, quality of service and load balancing, it also consider order of tasks accordance with workload, wasted time, priority and deadline, and gains an optimal solution by genetic algorithm. This proposed method tries to obtain an optimal tasks mapping on resources by minimizing completion time of tasks or, completion time of the last task (makespan). Furthermore, it decreases the task failure rate on running based on the resource failure frequency rate, and also increases the quality of responding using a special task ordering to run and to decrease the task starvation rate by proper parameters in it.

The features in this approach are:

- 1) Tasks are entered in system, each time.
- 2) All tasks are independent.
- 3) Each resource can process more than one request.

Algorithm: Synthetic Heuristic Algorithm for Independent Task Scheduling (SHIS)

Input: Set of available resources and unmapped tasks.

Output: An optimized generated dynamic schedule.

1. Calculate the available resources characteristics and network workload.
2. Repeat
3. for $\forall t_i \in T_{ready}$ do
4. The tasks are ordered by workload in ascending.
5. Make initial population.
6. Evaluate population.
7. While the stop conditions are met.
8. Cycle crossover operation.
9. Goal oriented mutation operation.
10. Select the best chromosomes.
11. End while.
12. Save the best solution.
13. Sort tasks, according to the candidate resource, in descending.
14. End for.
15. Dispatch all mapped tasks on candidate resources in obtained solution.
16. Update the ready task list.
17. Update the virtual resource list regarding current characteristics from data-center.
18. Until there are unscheduled tasks.

d.) Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm (MO-GA) [6]

In this paper, the overall performance of cloud computing, with the deadline constraint, a task scheduling model is established for reducing the system power consumption of cloud and improving the profit of service providers. For the scheduling model, a solving method based on multi-objective genetic algorithm (MO-GA) is designed and the research is focused on encoding rules, crossover operators, selection operators and the method of sorting Pareto solutions.

Algorithm: Multi Objective – Genetic Algorithm (MO-GA)

Encoding Rule

Each schedule is expressed as a 2 by M matrix, where, M is the length of the chromosome. The first row of the matrix represents the requested applications, and second of the matrix is the corresponding number of the cloud where the application is executed. Fig. 2 shows an example of scheduling result, in which, application 2 is assigned to cloud 0, and application 1 is allocated to cloud 5.

0	1	2	3	4	5
2	5	0	2	0	1

↙ Application Number
↙ Cloud Number

Fig. 2 Encoding example of a Scheduling

According to the above rule, we can see that each application can only be assigned to one cloud, while a cloud may be able to process several applications.

Population Initialization

The population initialization affects the quality of the future generations, and is an important step in the whole algorithm. In this paper, this step is conducted by combining the random and greedy initialization methods. Owing to the greedy initiation method, the scheduler rejects the applications not meeting the deadline constraint which may cause the whole scheduling failed. This kind of initialization method supports initial population and biasing search avoidance of MO-GA.

Genetic Algorithm

Genetic algorithm is a search heuristic that mimics the process of natural evolution based on a population of candidate solutions. It is routinely used to generate useful solutions to optimization and problems. In the process of evolution, a modification is performed by the operators on each individual. Each chromosome represents a scheduling result, and an evaluation operator (fitness) is called to evaluate the offspring. The following are the operations under Genetic algorithm:

- (1) Individual Evaluation
- (2) Selection operation
- (3) Crossover Operation
- (4) Mutation Operation

Optimal Selection in Pareto Archive

The results of MO-GA algorithm are a set of Pareto solutions, providing a wide range of possible options, while reducing the efficiency of scheduling process. In practice, users sometimes need to adjust the degree of preference for a particular objective dynamically. This step provides an approach to pick up an optimal solution among the external Pareto archive according to the current requirement. A two dimensional vector is introduced to represent the weighting for a particular objective, whose direction points to the most favorable solution.

Implementation Steps

Based on the above, the implementation steps of this algorithm are listed as follows:

- (1) Initial the population by greedy and random methods;
- (2) Modify the individual during the evolution process of the MO-GA algorithm according to the operators indicated in population initialization, and store the results to external Pareto archive;
- (3) Select the optimal solution according to the vector and implement the scheduling result to distributed cloud federation.

e.) Market-oriented hierarchical Scheduling strategy (MOHS) [7]

In this paper the service-level scheduling deals with the Task-to-Service assignment where tasks of individual workflow instances are mapped to cloud services in the global cloud markets based on their functional and non-functional QoS requirements; the task-level scheduling deals with the optimization of the Task-to-VM (virtual machine) assignment in local cloud data centers where the overall running cost of cloud workflow systems will be minimized given the satisfaction of QoS constraints for individual tasks. Based on our hierarchical scheduling strategy, a package based random scheduling algorithm is presented as the candidate service-level scheduling algorithm and three representative meta heuristic based scheduling algorithms including genetic algorithm (GA), ant colony optimization(ACO), and particle swarm optimization (PSO) are adapted, implemented and analyzed as the candidate task-level scheduling algorithms.

Algorithm: Market Oriented hierarchical Scheduling strategy overview

Service-level scheduling

Global scheduler: platform layer, static scheduling, workflow instances, service providers.

Input: Cloud workflow specifications, cloud service providers, unified resources

Output: Service-level scheduling plan and task-level scheduling plan

Step1: Search and produce the collection of suitable services based on functional requirements

Step2: Assign fine-grained QoS constraints based on the non-functional QoS requirements

Step3: Allocate suitable services and make reservations.

Task-level Scheduling

Local scheduler: unified resources layer, dynamic scheduling, workflow tasks, unified resources

Step1: Obtain the QoS constraints for each individual tasks (including both workflow and non-workflow tasks)

Step2: Optimize the Task-to-VM assignment

Step3: Implement the optimal scheduling plan.

Consider the algorithm Package based random scheduling algorithm in the given [5].

III. COMPARISON TABLE

SCHEDULING ALGORITHMS	DEADLINE CONSTRAINTS	BUDGET CONSTRAINTS	QOS (QUALITY OF SERVICE)	JOB SCHEDULING	TASK FAILURE REDUCTION	LOAD BALANCING	PRIORITY
DRPDBCA [3]	C	C	NC	C	NC	NC	C
ORP [4]	C	C	C	C	NC	NC	NC
SHIS [5]	NC	NC	C	C	C	C	NC
MO-GA [6]	C	NC	C	C	NC	NC	NC
MOHS [7]	NC	C	C	C	NC	NC	NC

*C- Considered parameter, *NC – Not Considered parameter

IV. CONCLUSION

Scheduling is one of the most important tasks in cloud computing environment. In this paper we have analyze various scheduling algorithms. Existing scheduling algorithm gives high throughput, high profit, efficiency, Deadline constraints. But still we need to improve the scheduling of jobs by reducing the load. So we have to improve many more algorithms by reducing the load, access time in this cloud computing environment

V. REFERENCES

- [1] Cloud Computing its basics, <https://www.ibm.com/developerworks/cloud/library/cl-cloudintro/>
- [2]http://www.iqynamics.com/cloud_computing.html
- [3] Naimisha S Trivedi ,Devyaniba S Chudasama, "Dynamic Resource Provisioning for Deadline and Budget Constrained Application in

- Cloud Environment", Int.J.Computer Technology &Applications, Vol 4 (3),462-465, May-June 2013
- [4] Chunlin Li · La Yuan Li, "Optimal resource provisioning for cloud computing Environment", Published online: 25 April 2012 © Springer Science+Business Media, LLC 2012
- [5] ArashGhorbanniaDelavar, Yalda Aryan, "A Synthetic Heuristic Algorithm for Independent Task Scheduling in Cloud Systems", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 2, November 2011
- [6] Jing Liu, Xing-GuoLuo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li, "Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013
- [7] ZhangjunWu, Xiao Liu,ZhiweiNi,Dong Yuan, Yun Yang, "A market-oriented hierarchical scheduling strategy in cloud workflow systems" , © Springer Science+Business Media, LLC 2011

AUTHOR'S BIOGRAPHY

Miss. R. Maheswari has received B.E. degree in Computer Science and Engineering from Anna University, Coimbatore and pursuing her M.E. degree in Computer Science and Engineering from Anna University, Chennai. She has published 3 papers in national conferences. She has presented 3 papers in technical symposiums. She has attended 3 Seminars and 5 workshops. She is one of the University rank holder during the year 2007-2011. She has secured centum in Java and Digital principals and system design. She has undergone training in MIT- FOSS Club for a period of 7 days. She is a member of MISTE. She has undergone in-plant training for 15 days in several organizations.



Mrs. S. Selvi, pursuing her Ph.D in the area of cloud computing. She has received her B.E degree in Computer Science and Engineering from Anna University, Chennai. She completed her M.E. degree in computer science and engineering from Anna University, Chennai. Her area of interest includes Cloud computing, Network security and Compiler design. She has published 8 papers in national conferences and 3 papers in international conferences. She has attended 6 national workshops, 1 seminar and 1 FDP program from various colleges. Also, she has organized 3 workshops, 3 Seminars, 2 guest lectures, 1 symposium and 1 national conference on various topics. Also she published papers in 3 journals.