A Review on Task Model and Task Scheduling in Cloud Computing

Sandeep kaur[1],
[1]Research Scholar
Rayat Institute Of Engineering and Information Technology, Punjab, India.

Pooja Nagpal[2]
[2] Faculty of Computer Science and Information Technology, Rayat Institute Of Engineering and Information Technology, Punjab, India.

Abstract: *Cloud computing is an emerging technology, that based upon internet computing and share resources (software and hardware) depends upon their demand. Cloud computing works on its important feature known as virtualization in order to access remote and geographically distributed resources. Depending upon cloud service provider and user requirements, a number of virtual machines are used. So it is necessary to schedule VM request. Nowadays scheduling task becomesa challenge for researchers. So a number of algorithms are used to provide proficiency of task and resource scheduling. In this paper, we have discussed different scheduling algorithms along with task model known as DAG (Directed Acyclic graph).*

Keywords: *Cloud computing, scheduling, DAG, HEFT.*

I. Introduction

Recently, the emerging cloud computing has offers new computing models where resources such as online applications, computing power, storage and network infrastructure can be shared as services through the internet [1].

1.1 CRUCIAL ISSUES IN CLOUD COMPUTING

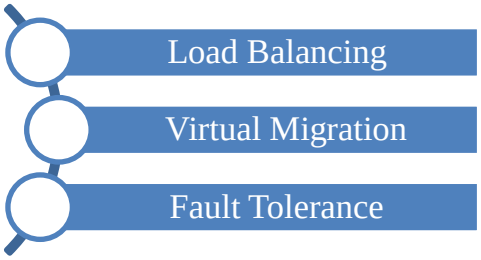


Figure Error! No text of specified style in document.**: Issues in Cloud Computing**

Load Balancing: It is the way toward circulating employments or load evening among different hubs of the framework so that time effectiveness has been expanded. Additionally appropriate use of resources happens [2].

Virtual Machine Migration: Virtual Machine (VM) relocation is an effective administration method that gives server farm administrators the capacity to adjust the position of VMs so as to better fulfill execution goals, enhance asset usage and correspondence area, alleviate execution hotspots, accomplish adaptation to non-critical failure, decrease vitality utilization. A virtual machine gives interface indistinguishable to basic uncovered equipment ▫ i.e. all gadgets, interferes with, memory, page tables and so on [3].

Points of interest of Virtualization:

- Efficient utilization of resources
- Superior level of reflection
- Replication
- Scalable and adaptable base

Adaptation to non-critical failure: Fault resistance permits the virtual machines to proceed with its employment even any piece of framework comes up short. This procedure moves the virtual machine starting with one physical server then onto the next physical server based upon the expectation of the disappointment happened, shortcoming tolerant relocation method is to enhance the accessibility of physical server and stays away from execution debasement of uses.

1.2 Directed Acyclic graph (DAG)

DAG modules have been mainly used in multi-processor based on parallel system. It is a directed graph which have no cycles [4].

A directed graph has a direction as well as a lack of cycles. The parts of the above graph are:

Integer = the set of the vertices.
Vertices set= {1,2,3,4,5,6,7,8,9,10}

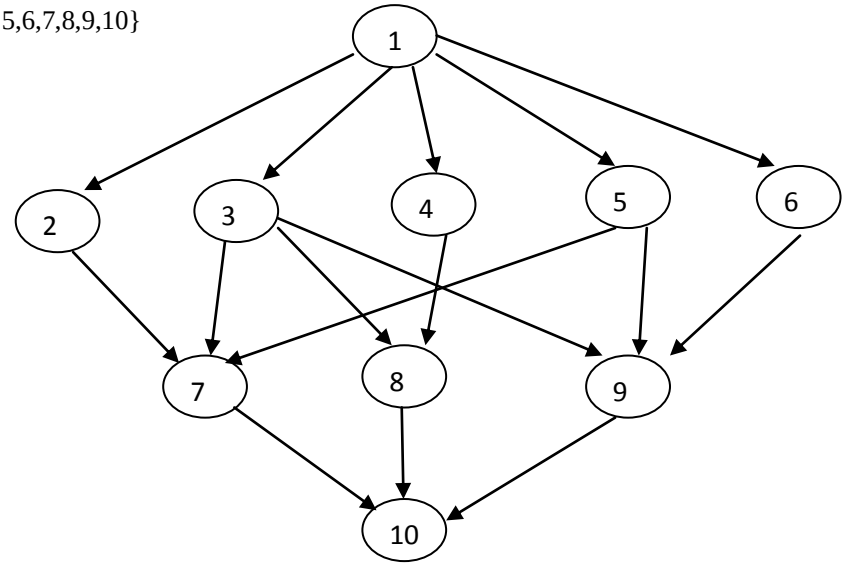


Figure 2: An example of DAG

Edge set= { (1,2,3,4,5,6), (3,7,8,4,9), (4,8), (5,9,7), (6,9), (9,10), 98,10), (7,10)}

A DAG graph was the combination of vertices and edges. As shown in figure 2 the vertices representing subtasks and the edges denote execution precedence between subtasks [5].

Each process is an indivisible unit of execution, expressed by node. A node has one or more inputs and can have one or more output to various nodes. When all inputs are available, the node is triggered to execute. After its execution, it generates its output. In this model, a set of node $(n_1, n_2, n_3 \ldots\ldots n_n)$ are connected by a set of a directed edges, which are represented by (n_i, n_j) where $n_i$is called the Parent node and n_j is called the child node. A node without parent is called an Entry node and a node with no child is an Exit node. The weight of a node is denoted by w (n_i) which represents the process execution time of a process. Since each edge corresponds to a message transfer from one process to another, therefore, the weight of an edge which is equal to the message transmission time from node $n_i ton_j$. Thus, w (n_i, n_j) becomes zero when $n_i and n_j$ are scheduled to the same processor because intra processor communication time is negligible compared with the inter processor communication time [6].

The granularity of DAG is

$$f(F) = min_{x=1,..w}\{f(x)\} \quad (1)$$

The DAG coarse grain is

$$f(L) \geq 1 \quad (2)$$

The grain of the task can be written as

$$F_x = \max\{F(k_x)F(L_x)\} \quad (3)$$

And the DAG's Granularity is

$$F(F) = max_{x=1,\ldots v}\{F_x\} \quad (4)$$

The DAG's fine grain is introduced as

$$F(F) \leq 1$$

DAGs can model many various kinds of information. A spreadsheet can be mold as a DAG, with a vertex for each cell and an edge whenever the principle in one cell uses the value from another; a topological ordering of this DAG can be utilized to modify all cell values when the spreadsheet is varied. Similarly, topological orderings of DAGs can be usage to order the anthology operations in a make file [7]. The program evaluation and review method uses DAGs to model the milestones and activities of large human projects, and schedule these projects to use as a little total time as possible. Combinational logic blocks in electronic circuit propose, and the operations in dataflow programming languages, engage acyclic networks of processing elements. DAGs can also symbolize collections of events and their influence on each other, moreover, in a probabilistic structure such as a Bayesian network or as a evidence of historical data such as family trees or the version histories of distributed revision control systems. DAGs can also be used as a compact representation of sequence data, such as the directed

acyclic word graph illustration of a collection of strings, or the binary decision diagram representation of sequences of binary choices. More abstractly, the ability relation in a DAG forms a partial order, and a limited partial order that may be represented by a DAG using ability.

Computational trouble on DAGs consists of topological sorting (finding a topological ordering), construction of the transitive closure and transitive reduction , and the closure problem, in which the aim is to find a minimum-weight subset of vertices with no edges link between them to the rest of the graph [8].

1.3 Job scheduling in cloud computing

The scheduling in multiprocessors is the method to allot the tasks in the system in such a manner that all utilized the resources and give better output [9]. Proper scheduling of the resources is possible only when proper load balancing and effectively shares the available resources, thus we will obtained better quality service. Scheduling algorithms are mainly used schedule all the subtasks, so that one can obtained better output without violating proceeding constraints. There are a number of scheduling techniques that are used to solve the problem occur in cloud computing [10]. Some of them are discussed below:

1.3.1 Heterogeneous Earliest Finish Time Algorithm (HEFT)

The HEFT Algorithm is one of the best and well accepted list deployed heuristics. The HEFT calculation is a compelling answer for the DAG Scheduling issue on heterogeneous framework. The confinement of HEFT calculation is that it utilizes procedures that are all static methodologies of the mapping issue that accept static conditions for a given time. HEFT is a two- phase scheduling algorithm with the heterogeneous processors. The phase which is used to assign the priority to tasks is the first phase, known as the task prioritization phase and for assigning the priority, the upward rank of every task is calculated. The upward rank is the critical path of the task that is the maximum amount of communication time and the average implementation time right from the start of any task to the end of any task. The phase which is used to schedule the tasks into the process which provides the EFT of the task is the Processor Selection phase (second phase). This phase has the insertion based policy that takes the maximum possibility of the insertion of the tasks in the less time i.e. the earliest idle time among the tasks that are already scheduled on the processor. The time complexity of the HEFT algorithm is$O(v^2xp)$, where $v$the number of tasks in the graph and p is the number of processors [11].

Furthermore, in complex circumstances it can undoubtedly neglect to locate the ideal scheduling;

1) The HEFT calculation first computes normal execution time for every errand (task mean) and normal correspondence time between assets of two progressive errands.

Let time (Ti, r) be the execution time of undertaking $T_i$on asset r and let Ribe the arrangement of every accessible asset for processing Ti and ω is the estimated time. Normal execution time of an errand $Ti$is characterized as

$$|\omega| = \sum r\epsilon Ritime\ (Ti, r)\ /|R| \quad (6)$$

2) Let time (eij, ri, rj) be the data transfer time between resources $ri$and rjwhich process the tasks $Ti$and Tjrespectively. Let $Ri$and Rjbe the set of all available resources for processing $Ti$and Tjrespectively. The average transmission time $T_i$to $Tj$is defined by:

$$|C1| = \sum r\epsilon Ri,\ r\epsilon Ritime\ (eij, ri, rj)\ /|Ri|\ |Rj| \quad (7)$$

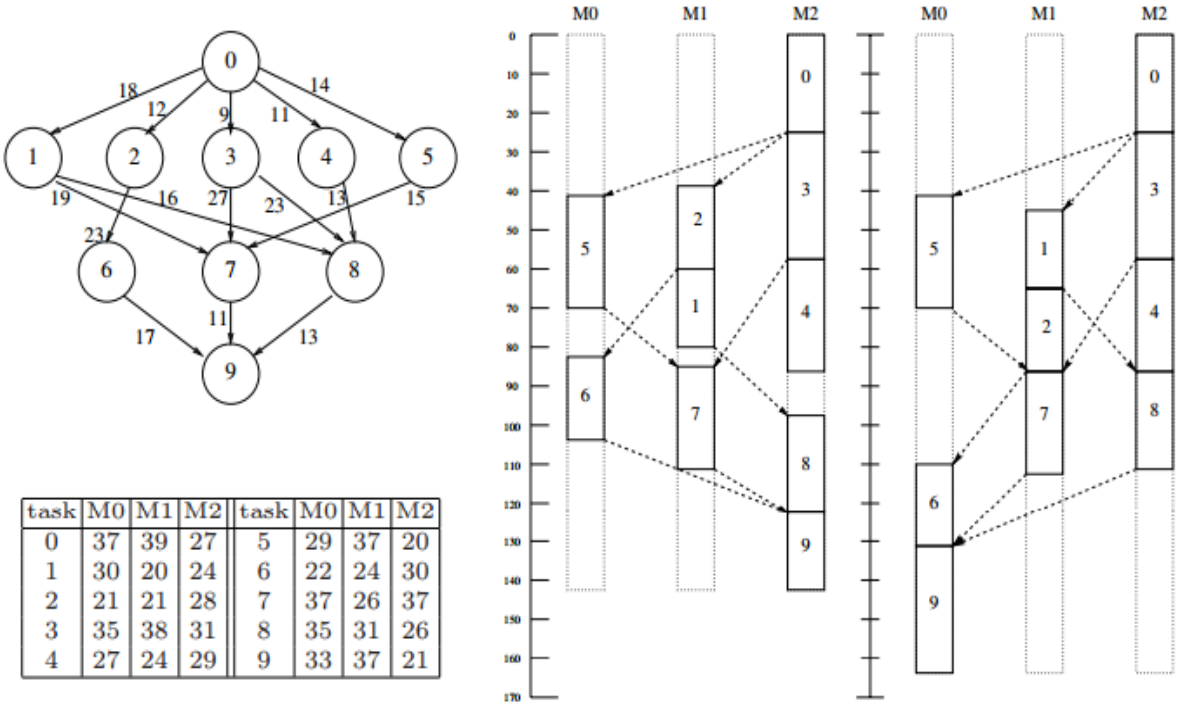
3) Then tasks in the workflow are ordered in HEFT based on a rank function.ωi is the estimated time of each task and i is for each task range from i to n .For an exit task T_i, the rank value is:

$$Rank\ (Ti) = |\omega i| \quad (8)$$

4) The rank values of other tasks are computed as:

$$Rank\ (Ti) = |\omega i| + maxTj\epsilon succ\ (Ti)\ (C1 + rank\ (Ti)) \quad (9)$$

Where, $succ\ (Ti)$is the location of prompt successors of assignment Ti .The calculation then sorts the assignment by diminishing request of their rank qualities. The undertaking with higher rank worth is given higher need. In the asset determination stage, errands are planned by needs and every assignment is allocated to the asset that can complete the undertaking at the soonest time [12].



task	M0	M1	M2	task	M0	M1	M2
0	37	39	27	5	29	37	20
1	30	20	24	6	22	24	30
2	21	21	28	7	37	26	37
3	35	38	31	8	35	31	26
4	27	24	29	9	33	37	21

Fig.3: A Motivating Example: Two different schedules for two different rank schemes

Table 1: comparative study of different scheduling algorithms based on various factors

Scheduling algorithm	Scheduling factors	Environment	Objective criteria	Advantages	disadvantage
First come first serve (FCFS)	time	Cloud computing	To provide efficient energy	simple to implement	Non- pre emptive algorithm
Round Robin	Time	Cloud computing	To execute tasks in a given time period	Better utilization of resources in balances order	Consume high power
Heterogeneous Earliest Finish Time (HEFT)	Highest upward rank	Grid environment	Select thetask with highest upward rank value	It decreases makespan of tasks in a DAG	No load balancing
Resource-Aware acheduling algorithm (RASA)	Grouped task	Grid Environment	To provide fastest execution	This algorithm aslso decrease th makespam and it recovers the disadvantages occurs in Max-Min and Min-Min algorithm	It ignores the waiting time of the large task in Min-Min scheduling and only consider the resource demand not user preferences.

Critical-path-on a processor algorithm (CPOP)	Highest upward and downward rank	Grid environment	It works into three phases namely: task prioritizing, task selection and processor selection phase.	It allocates all the tasks that follow a critical path to a single processor. It has better scheduling quality performance and running time	Execution time is high
Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT)	A group of task	Cloud environment	It consist of teo mechanism namely: Listing mechanism and machine assignment mechanism	It increase workflow execution time, additionally it enables resources to scale elastically at runtime.	It does not concentrate on multiple parameters like rate, cost, time, makespan, availability and reliability.

II. Related work

Cvadaret. Al [12] has focused on computing and networking proposals for green data centers, even though they briefly describe some other green research related to data centers such as cloud computing, cooling. **A.Jain [13]** discussed that the large amount of CO_2 dissipation in environment has generated the necessity of Green computing (saving energy by recycling it and reusing it over a period of time and minimizing the wastage in terms of usage of resources). More processor chips generates more heat, more heat requires more cooling and cooling again generates heats and thus they come to a stage where author want to balance the system by getting the same computing speed at decreased energy consumption. **Fumiko Satoh et al. [15]** focussed on reducing the usage of energy in data centers. But for the future energy management, the author has developed an energy management System for cloud by the use of sensor management function with an optimized VM allocation tool. This system has helped to reduce the energy consumption in multiple data centres and results shows that it will save 30% of energy. This system also used to reduce the energy in carbon emissions. **Dzmitry Kliazovich et al** proposed a new communication model for cloud computing applications, the proposed model is known as CA-DAG. The proposed model is based on DAG (Directed Acyclic Graphs. The simulation results shows that the system have improved efficiency.**Liu Yuan, et al.**10

Table 2: Comparative study of the existing techniques in task scheduling

Author	Title	Description
Xiu-Jie XU and Chuang-BAi XIAO	Hybrid Scheduling Deadline-Constrained Multi-DAGs Based on Reverse HEFT	Author proposed a reverse HEFT scheduling policy, which based upon the highest priority in each DAG
Karan R. Shetti et al.	Optimization of the HEFT algorithm for a CPU-GPU environment	Proposed a novel HEFT algorithm. Author also proposed a method for task ranking.
Zhaobin Liu et al.	DAG Cluster Scheduling Algorithm for Grid Computing	Author combined two scheduling list and task scheduling and proposed a method DAG cluster

		algorithm known as CFTD.
L. F. Bittencourt et al.	DAG Scheduling Using a Lookahead Variant of the Heterogeneous Earliest Finish Time Algorithm	Proposed an improved HEFT algorithm that will effectively resuce the makespan.

III. Conclusion

When we are using cloud computing technology, we have to face a lot of new challenges. The main problem occur in cloud computing is task scheduling. So, the main aim of the scheduling is to utilize more and more resources, so that makespan get reduced. A comparative study of different scheduling methods ahs been explained above. A number of researchers has researched to provide a better solution for task scheduling.In this paper a number of scheduling algorithm of cloud environment based on different parameters like objective, scheduling factors their advantages and disadvantages has been discussed.

References

1. Chauhan N. and Saxena A. (2013), "A Green Software Development Lifecycle for Cloud Computing", IEEE's IT Pro, pp. 28-34.
2. Kaur, Rajwinder, and Pawan Luthra, "Load balancing in cloud computing," *Second Symposium on Cloud computing*. 2012.
3. Xiao, Zhen, Weijia Song, and Qi Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE transactions on parallel and distributed systems* 24.6 (2013): 1107-1117.
4. D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan and A. Y. Zomaya, "CA-DAG: Communication-Aware Directed Acyclic Graphs for Modeling Cloud Computing Applications," *2013 IEEE Sixth International Conference on Cloud Computing*, Santa Clara, CA, 2013, pp. 277-284.
5. X. J. Xu, C. B. Xiao, G. Z. Tian and T. Sun, "Hybrid Scheduling Deadline-Constrained Multi-DAGs Based on Reverse HEFT," *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, Hong Kong, 2016, pp. 196-202.
6. Selvarani, S., and G. Sudha Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," *Computational intelligence and computing research (iccic), 2010 ieee international conference on*. IEEE, 2010.
7. Z. Liu, T. Qin, W. Qu and W. Liu, "DAG Cluster Scheduling Algorithm for Grid Computing," *2011 14th IEEE International Conference on Computational Science and Engineering*, Dalian, Liaoning, 2011, pp. 632-636.
8. L. F. Bittencourt, R. Sakellariou and E. R. M. Madeira, "DAG Scheduling Using a Lookahead Variant of the Heterogeneous Earliest Finish Time Algorithm,"*2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, Pisa, 2010, pp. 27-34.
9. Patil, Neeta, and Deepak Aeloor, "A review-different scheduling algorithms in cloud computing environment," *Intelligent Systems and Control (ISCO), 2017 11th International Conference on*. IEEE, 2017.
10. Kavyasri, M. N., and B. Ramesh, "Comparative study of scheduling algorithms to enhance the performance of virtual machines in cloud computing,"*Emerging Trends in Engineering, Technology and Science (ICETETS), International Conference on*. IEEE, 2016.
11. K. R. Shetti, S. A. Fahmy and T. Bretschneider, "Optimization of the HEFT Algorithm for a CPU-GPU Environment,"*2013 International Conference on Parallel and Distributed Computing, Applications and Technologies*, Taipei, 2013, pp. 212-218.
12. Abdelkader, Doaa M., and Fatma Omara, "Dynamic task scheduling algorithm with load balancing for heterogeneous computing system," *Egyptian Informatics Journal* 13.2 (2012): 135-145.