

Design and Development of Efficient Cloud Scheduling Algorithm based on Load Balancing Analytics

Ch. Sridevi
Dept of CSE
Adikavi Nannaya University
Rajamahendravaram

P. Suresh Varma
Dept of CSE
Adikavi Nannaya University
Rajamahendravaram

S. Rama Krishna
Dept of Chemistry
Adikavi Nannaya University
Rajamahendravaram

ABSTRACT

Cloud computing involves sharing computing resources rather than having individual servers or personal devices to handle applications. Cloud computing architectures include the delivery of software, infrastructure, storage and technology enabled services over internet to the people and organizations on demand. Cloud scheduling is the process of allocating resources to the job requests in the form of Virtual Machines. In this paper we designed and developed a novel, efficient cloud scheduling algorithm based on load balancing analytics for allocation of physical resources in the form of virtual machine to the incoming job requests. In this paper we measured various cloud performance metrics like mean turnaround time and mean waiting time. The results obtained with this method compared with traditional methods like First Come First Serve (FCFS), two stage scheduling algorithms and observed considerable increase in the performance Metrics.

General Terms

Scheduling algorithm, resource allocation, Cloud computing, optimization, Virtual Machine.

Keywords

Cloud computing, Virtual Machine, Load Balance, Two stage scheduling.

1. INTRODUCTION

The definition of Cloud Computing, as provided by National Institute of Standards and Technology (NIST), per work [1] is, "A model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources (e.g. Networks, Servers, Storage, application and services) that can be rapidly provisioned and released with minimal management effort or the service provider interaction".

The majority of cloud computing infrastructure currently consists of reliable services delivered from datacenter which is built on servers with different levels of virtualization technologies. Many companies provide the cloud computing platform such as Amazon, Microsoft, Google, Rackspace, IBM, VMware etc. Cloud computing system provides the service to the user and is characterized by high scalability, portability, elasticity and reliability. The resources of the cloud computing system are transparent to the application and the user does not know the location or configuration or capacity details of the resources. The users can access data and applications from anywhere through internet which means great portability.

A major challenge in cloud computing is scheduling that is to allocate tasks to available resources on the basis of task qualities, metrics and requirements without affecting the

services provided by the cloud. Scheduling in cloud computing system decides how to allocate the resources such as CPU, memory, secondary storage space, I/O, network bandwidth etc. between users and tasks. A good scheduler adapts its scheduling strategy according to the changing environment over time, the nature of tasks and relative metrics of the tasks involved in batch of tasks submitted for processing.

Scheduling refers to the set of policies to control the order of work to be performed by a computing system. There are various types of scheduling algorithms existing in distributed computing system, and job scheduling is one of them. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. Scheduling manages availability of resources like CPU, memory and good scheduling policy gives maximum utilization of resources. Virtual Machine (VM) is a process of mapping virtual machines to Physical Machines. As Virtualization is a core technology of cloud computing, allocation of Virtual Machines has become important challenge in cloud computing. Several research works addressed the importance of placing VMs appropriately. Recently in works [2] and [3] two stage scheduling model is designed and developed for allocation of resources in cloud to the incoming jobs in the form of Virtual machines. Load balancing affects cloud computing and improves the performance by re-distributing the load among the computing resources. Jobs are transferred from one node to another through the network.

Little work is reported till date regarding cloud scheduling with load balancing strategy. Hence in this paper we designed and developed a novel efficient cloud scheduling algorithm based on load balancing analytics for allocation of physical resources in the form of virtual machine to incoming job requests.

2. REVIEW OF LITERATURE

Cloud computing is a model for providing information technology services over internet through web based tools and applications. Cloud scheduling is the process of allocation of resources to the incoming jobs to improve resource utilization and increase the throughput and performance of the cloud. Cloud service scheduling can be categorized as user level and system level. User Level scheduling solves problems raised by service provision between providers and customers. The system level scheduling handles resource management within the datacenter.

Amazon played a key role in cloud computing development by launching Amazon web services on utility basis in 2006. Millions of users share cloud resources by submitting jobs to the cloud system. Scheduling of these millions of jobs is a

challenging activity in cloud computing. Various scheduling strategies are proposed in works [4], [5], [6], [7], [8], [9], [10], [11] and [12] to list a few.

Work [13] in cloud scheduling methods has presented a study on scheduling methods in cloud computing, and it has reported different scheduling strategies and algorithms suitable to different cloud computing environments. Also the work [14] has presented scheduling in cloud, where the authors have designed a protocol to minimize the switching time, and improve the resource utilization by using back filling and by assigning the shortest distance resources to the job to minimize the make span and also improve the server performance and throughput.

In work [15], Bin Lin et al (2014) has presented user-driven scheduling of interactive virtual machines. The authors developed a distributed computing system called Virtuoso, which presents virtual machines (VMs) as its fundamental abstraction to the end users. Long-running and non-interactive VMs may coexist on the same host which is used to run VMs being used by highly interactive. The work used direct user feedback to balance between the non-interactive VMs and interactive VMs by providing good average computation rates and fair service to both of them. In another work [16], Arabi E. Keshk(2014) has presented an online cloud task scheduling model based on virtual machine adaptive fault tolerance and load balancing using ant colony algorithm . The main contributed work is the addition of load balancing factor so that the system tolerates the faults by tracking the decision on the basis of reliability of the virtual machines in scheduling process.

Work [17] highlighted a new scheduling framework In this work the authors have presented a novel, energy-aware task scheduling framework that makes integrated exploitation of the VM reuse technique, in cloud task scheduling in a datacenter and achieved better energy efficiency without sacrificing the Quality of Service (QoS) of the system. This resource utilization model is much more accurate and suitable for cloud datacenters over traditional models. Work [18] has presented optimal load balancing in cloud computing by efficient utilization of virtual machines by allocating the incoming requests to the available virtual machines in an

efficient manner and analyzed the performance using cloud simulator.

Recently work [2] presented two stage scheduling algorithm and performance evaluation of cloud. They reported a novel approach for scheduling given job requests through two stage scheduling algorithm. Simulation results show that reduced average turnaround time, average waiting time and total lapsed time when compared to other scheduling algorithms.

Very little work has been reported in the literature regarding cloud scheduling using load balancing strategy. Hence in this paper proposed an optimal dynamic scheduling algorithm to schedule job requests in cloud computing by extending earliest deadline first algorithm.

3. CLOUD SCHEDULING ALGORITHM BASED ON LOAD BALANCING ANALYTICS

Cloud computing datacenter offers, information technology (IT) resources as services. The Hardware Systems and Software systems represent the resources the data center provides as Infrastructure as a service and platform as a Service respectively. Virtual Machines provided by the IaaS and PaaS.

In cloud computing, each job has requests, of physical resources in the form of VM's in sequence to complete its task. . Scheduling is the process of allocation of physical resources in the Form of Virtual machine to the incoming jobs. In this section we consider $\{J_1, J_2, \dots, J_n\}$ be the set of job-requests arrived at the scheduler in the cloud at a particular instance of time. In this model, the role of scheduler is to allocate the resources of VM1 and VM2 in the form of virtual machine instances for each job-request to execute sequentially on VM1 and VM2, the scheduler also considers the dead line period of each job-requests as a criteria for allocation of the resources. If each job-request, J_i requires an execution time of ET_{1i} on virtual machine of type-1(VM1) and ET_{2i} on virtual machine of type-2 (VM2), the dead line period of the job-request is considered to be three times of the total execution time on both the virtual machines i.e, $3(ET_{1i}+ET_{2i})$.

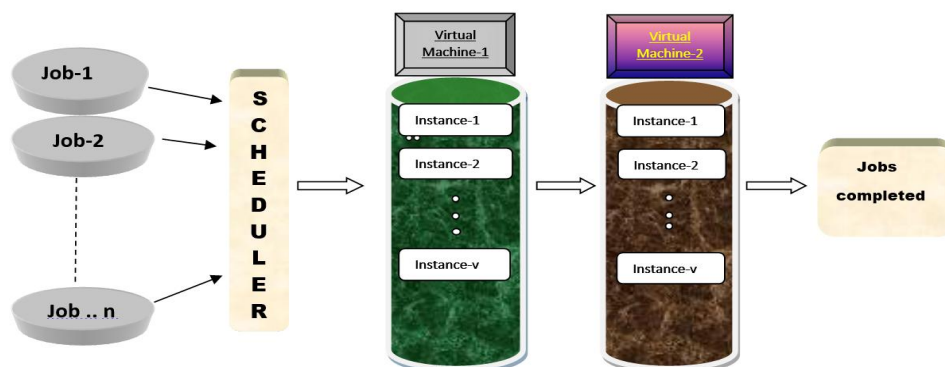


Fig.3.1: Cloud scheduling model with load balancing

Optimal dynamic scheduler is one type of Event-driven scheduler. Event-driven schedulers are invariably used in all moderate and large-sized applications having many jobs.

In this paper is proposed a novel Optimal dynamic scheduling algorithm which accepts 'n' number of job-requests, where 'n' is in multiples of 2, and divides them into 'v' sub scheduling sequences. The algorithm arranges each sub scheduling sequence in the ascending order with priority,

based on the shortest Deadline period, among the unprocessed jobs. The Optimal dynamic Scheduling algorithm creates ‘v’ number of virtual machine instances for each virtual machine type on respective physical resources. The solution vector is an array, which stores the scheduling sequence of job-requests and can be processed with one virtual machine instance of each virtual machine type in sequence. In this model a scheduling sequence $S = \{ J_1, J_2, J_3, \dots, J_n \}$ can be divided into two sub sequences as $S_1 = \{ J_i, \text{ for all } i \leq t \}$, $S_2 = \{ J_i, \text{ for all } i > t \}$ where $t = (n \% v)$, and each sub sequence has to be scheduled for each instance of virtual machine type. For example the scheduling sequence which consists of 8 job requests $\{ J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8 \}$ can be divided into 2 sub sequences as follows. $S_1 = \{ J_1, J_2, J_3, J_4 \}$, $S_2 = \{ J_5, J_6, J_7, J_8 \}$. Every job J_i in each sub sequence is assigned with a priority based on the dead line period. In the sequence of the job requests, the job with earliest period of deadline is assigned with highest priority and so on. Calculate then the turnaround time and waiting time.

Waiting Time of a job request is the time elapsed between the arrival time of job request and when the job request starts its work on Virtual Machine of type-1, plus the time elapsed between the time it completes its work on Virtual Machine of type-1 and starts its work on Virtual Machine of type-2. Turnaround Time of job request is the time elapsed between the completion time of the job request and the arrival time of the job request. The performance metrics can be computed by the following computations for a given scheduling sequence. Average Waiting Time (AWT), Average Turnaround Time (ATT), of all requests can be computed as

$$AWT = \frac{\sum_{i=1}^n ((C_i - ET_{i2}) - (S_i + ET_{i1}))}{n} \quad (3.1)$$

$$ATT = \frac{\sum_{i=1}^n C_i}{n}$$

Where

C_i : Completion Time of job request J_i in Virtual Machine of type-2.

S_i : Start Time of job request J_i in Virtual Machine of type-1.

Algorithm: Optimal dynamic Scheduling
($ET_{[1..n]1}$, $ET_{[1..n]2}$, No of instances, v)

1. Begin
2. Find total number of Sub-Sequences, given no. of instances and total no. of jobs
3. Sub_seq_count=v ;
4. Calculate the total execution time of each job on both VM's together i.e $TET_{[1..n]}$
5. For i=1 to n do
6. Begin
7. $TET[i] = ET[0][i] + ET[1][i]$
8. End
9. For i=1 to n do
10. Begin
11. Sub_index = n%v;
12. Append Optimal_seq[i] to subsequence $S[\text{sub_index}]$;
13. End
14. Take deadline with respect to the execution time.

15. Assign priority to each task based on earliest deadline first.
16. For t = 1 to sub_seq_count do
17. Begin
18. Sort $S[t]$ based $TET_{[1..n]}$
19. End
20. For i=1 to n do
21. Begin
22. Calculate average waiting time (AWT), average Turnaround time (ATT) for each subsequence $S[i]$;
23. End;
24. End ;

4. PERFORMANCE MEASURE OF THE CLOUD SCHEDULING ALGORITHM

To evaluate the performance of the current model, the author has considered 8 job requests and their corresponding execution times on virtual machine1 and virtual machine2 and is given in the table 4.1.

Table 4.1: Job requests and time required to process on VM1 and VM2

Job request, J_i	Execution time on VM -1	Execution time on VM -2
1	23	56
2	39	10
3	65	51
4	21	35
5	43	23
6	25	8
7	75	45
8	38	9

4.1 Calculation of the deadline period:

In this paper the author also observed deadline period for the evaluation of performance. The dead line period of a job-request is considered to be three times of the total execution time on both the virtual machines i.e. $3(ET_{i1} + ET_{i2})$, as shown in the table 4.2 below.

Table 4.2: Job requests and time required to process on VM1 and VM2, Deadline period (DP).

J_i	ET on VM -1	ET on VM -2	TEP	DP
1	23	56	79	237
2	39	10	49	147
3	65	51	116	348
4	21	35	76	168
5	43	23	66	198
6	25	8	33	99
7	75	45	120	360
8	38	9	47	141

The given job requests are divided into two sub sequences and each sub sequence is processed with an instance of each virtual machine type. The above sequence of eight job requests, with two virtual machine types, is divided into two sub sequences S1 = { 1, 2, 3, 4 } and S2 = { 5, 6, 7, 8 }. Priority is assigned to each Job-request Ji, depending on the period of the deadline, for each sub sequence separately and the job with earliest deadline is assigned with top priority. The Turnaround Time (TAT) and Waiting Time (WT) are then calculated for each subsequence S1 and S2 as mentioned above. The results are presented in the table 4.3 and table 4.4 below, for both the sub sequences S1, S2 respectively.

Table 4.3: Optimal dynamic scheduling for sub sequence S1

J _i	ET 1	ET 2	DP	P	TAT	WT
1	23	56	237	3	151	74
2	39	10	147	1	49	0
3	65	51	348	4	202	86
4	21	35	168	2	95	39

Average Turnaround Time = 497/4 = 124.25

Average Waiting Time = 199/4 = 49.75

Gantt chart for Optimal dynamic scheduling for sub sequence S1

	0	39	60	83	148	3.1
VM-1	J ₂	J ₄	J ₁	J ₃	3.1.1	
VM-2	3.1.2	J ₂	3.1	J ₄	J ₁	J ₃
	3.1.4	49	3.2	95	151	202

Table 4.4: Optimal dynamic scheduling for sub sequence S2

J _i	ET 1	ET 2	DP	P	TAT	WT
5	43	23	198	3	129	63
6	25	8	99	1	33	0
7	75	45	360	4	226	106
8	38	9	141	2	72	25

Average Turnaround Time = 460/4 = 115.00

Average Waiting Time = 194/4 = 48.50

Gantt chart for Optimal dynamic scheduling for sub sequence S2

	0	25	63	106	181	3.3		
VM-1	J ₆	J ₈	J ₅	J ₇	3.3.1			
VM-2	3.3.2	J ₆	3.3	J ₈	3.3	J ₅	3.3.3	J ₇
	3.3.6	3.3.7	33	72	129	226		

Table 4.5: Calculation of Total Average Turnaround Time and Total Average Waiting Time

Performance evaluation metrics in Optimal dynamic scheduling with two virtual machine instances for each virtual machine type			
	S ₁	S ₂	Avg. (S ₁ ,S ₂)
ATT	124.25	115.00	119.625
AWT	49.75	48.50	49.125

From the equations 3.1, Table 4.4 and 4.5 average turnaround times (ATT) for sequence S1 is 124.25, sequence S2 is 115.00 and total average Turnaround Time of both S1 and S2 is 119.625. Average Waiting Time (AWT) for S1 is 49.5, S2 is 48.50 and total average waiting time is 49.125.

5. COMPARATIVE STUDY

A custom simulation environment has developed in JAVA to analyze the First Come First Server (FCFS) Scheduling, Two-stage Scheduling and Optimal Scheduling Algorithm as follows.

5.1 FCFS's scheduling, the given job requests are divided into two sequences based on first come first serve and each sub sequence is processed with an instance of each virtual machine type. The above sequence of eight job requests is divided into two sub sequences S1 = { 1, 2, 3, 4 } and S2 = { 5, 6, 7, 8 }. Initially, one can process J1 on first instance of virtual machine type-1, J5 on second instance of virtual machine type-1 followed by remaining jobs sequentially for each sub sequence. Later each job request processes its operation on an instance of virtual machine type-2. Start Time, Turnaround Time and Waiting Time of the given job request is shown in the table5.1

Table5.1: First Come First Service scheduling for sub sequence S1

Job Request, J _i	Start time of J _i	Turnaround time of J _i	Waiting time of J _i
1	0	79	0
2	23	89	40
3	62	178	62
4	127	213	157

Average Turnaround Time = 559/4 = 139.75

Average Waiting Time = 259/4 = 64.5

Table5.2 First Come First Service scheduling for sub sequence S₂

Job Request, J _i	Start time of J _i	Turnaround time of J _i	Waiting time of J _i
5	0	66	0
6	43	76	43
7	68	188	68
8	143	197	150

Average Turnaround Time = 537/4 = 134.5

Average Waiting Time = 261/4 = 65.2

Table5.3 Calculation of Total Average Turnaround Time and Total Average Waiting Time

Performance evaluation metrics in FCFS scheduling with two virtual machine instances for each virtual machine type			
4.	S ₁	S ₂	Avg (S ₁ ,S ₂)
ATT	139.75	134.5	137.125
AWT	64.5	65.2	64.88

5.2 Two-stage Scheduling, the given job requests are initially arranged in to a sequence by following Johnson’s rule, resulting in a scheduling sequence of {4, 1, 3, 7, 5, 2, 8, 6}, for the problem under consideration. Later the scheduling sequence is divided into two sub sequences by allotting job requests alternatively. The above given scheduling sequence is divided into sub sequence as S₁ = {4, 3, 5, 8} and S₂={1, 7, 2, 6}.Table 5.4 and 5.5 shows the start time, turnaround time and waiting time of sub sequence S₁ and S₂ respectively.

Table 5.4: Two stage scheduling for sub sequence S₁

Job Request, J _i	Start Time of J _i	Turnaround Time of J _i	Waiting Time of J _i
4	0	56	0
3	21	137	21
5	86	160	94
8	129	176	129

Average turnaround time = 529/4 = 132.5

Average waiting time = 244/4 = 61

Table 5.5: Two stage scheduling for sub sequence S₂

Job Request, J _i	Start Time of J _i	Turnaround Time of J _i	Waiting Time of J _i
1	0	179	0
7	23	143	23
2	98	153	104
6	137	170	137

Average turnaround time = 545/4 = 136.25

Average waiting time = 264/4 = 66

Table 5.6: Calculation of Total Average Turnaround Time and Total Average Waiting Time

Performance evaluation metrics in Two Stage scheduling with two virtual machine instances for each virtual machine type			
5.	S ₁	S ₂	Avg(S ₁ ,S ₂)
ATT	132.25	136.25	134.25
AWT	61	66	63.5

Table 5.7: Comparison table for FCFS, Two Stage and Optimal Dynamic scheduling algorithms

Comparison of performance evaluation metrics in FCFS, Two-stage and Optimal dynamic scheduling with two virtual machine instances for each virtual machine type			
6.	FCFS	Two-Stage	Optimal dynamic
ATT	137.125	134.25	119.625
AWT	64.88	63.5	49.125

It is evident from the table 5.7, the average turnaround time and average waiting time was reduced for optimal dynamic scheduling algorithm in comparison to FCFS and Two Stage scheduling algorithms. The average turnaround time, average waiting time for optimal dynamic scheduling with two virtual machines was slashed by 10.89%, 22.63% in comparison to two stage and about 12.76%, 32% in comparison to FCFS algorithms. The above results are shown in the fig.5.1.

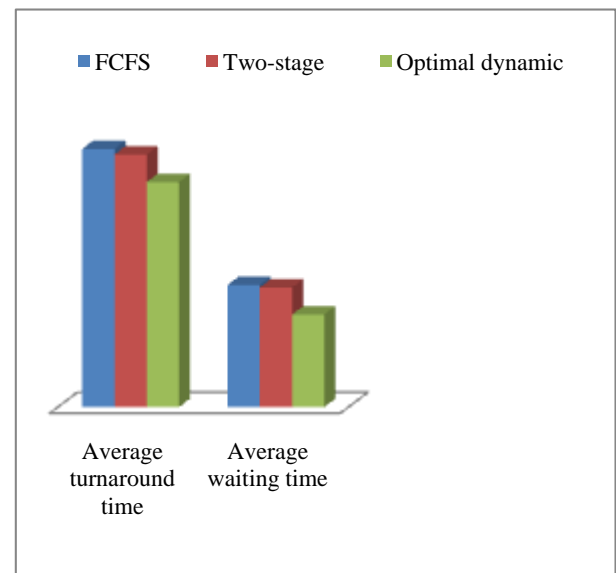


Fig.5.1 Comparison of performance evaluation metrics in different scheduling strategies (n=8)

Similarly performance evaluation metrics is compared for the fore said algorithms by considering a job set containing

sixteen (n=16) job requests, Table 5.8 below shows the simulation results. Fig.5.2 depicts the comparison of FCFS, Two stage scheduling algorithm with Optimal dynamic scheduling algorithm multiple VM instances with respect to the given two valuation metrics.

Table 5.8: Job Requests (n=16) and Time required to process on VM1 and VM2

Job request, J_i	ET_{i1}	ET_{i2}
1	10	34
2	30	23
3	25	37
4	18	9
5	82	15
6	30	40
7	50	49
8	21	32
9	19	6
10	43	25
11	54	70
12	65	27
13	43	28
14	7	9
15	3	4
16	2	1

Table 5.9: Comparison table for FCFS, Two Stage and OD scheduling algorithms

Comparison of performance evaluation metrics in FCFS, Two-stage and Optimal dynamic scheduling with two virtual machine instances for each virtual machine type			
	FCFS	Two-Stage	Optimal dynamic
ATT	180.685	155.375	105.75
AWT	137.437	98.75	67.0625

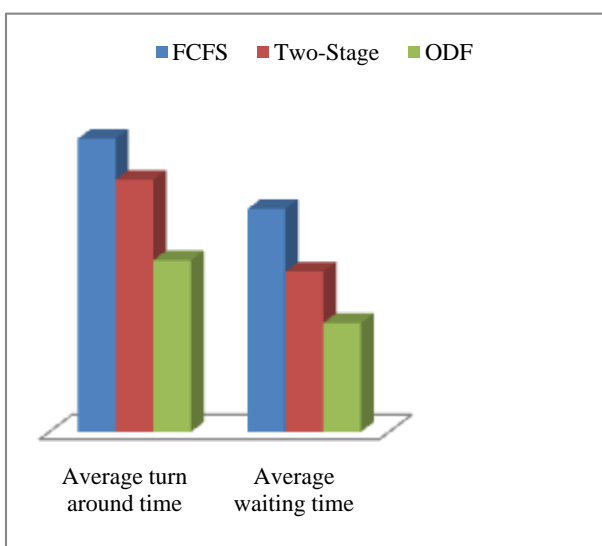


Fig.5.2: The performance evaluation metrics in different scheduling strategies (n=16)

With Optimal dynamic scheduling algorithm, the average turnaround time, average waiting time are reduced by 41.47%, 51.20% respectively in comparison to FCFS and about 32% both compared to Two-Stage scheduling algorithm. Hence the Optimal dynamic scheduling algorithm gives better performance metrics when compared with FCFS and two-stage scheduling. It is also observed that the performance of the optimal scheduling algorithm is improved as the number in the job set is increased.

6. CONCLUSION & FUTURE SCOPE

Cloud computing is internet based computing in which resources are provided to users on demand. Scheduling is one in all foremost vital tasks in cloud computing atmosphere. In this paper the author designed and developed an optimal dynamic cloud scheduling algorithm based on load balancing analytics. This method shows better performance evaluation metrics when compared with FCFS and Two-Stage scheduling algorithms. The total turnaround time and average waiting time are considerably decreased in Optimal Dynamic scheduling when compared to FCFS and Two-Stage scheduling algorithm.

7. FUTURE SCOPE

This method can be extended further for various jobs using Service Level Agreement for allocation of virtual machines in cloud computing.

8. REFERENCES

- [1] Peter Mell, and Timothy Grance, 2011, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, U.S. Department of Commerce, NIST Special Publication.
- [2] P. Krishnam Raju Indukuri, P. Suresh Varma, M. V. Rama Sundari, 2015, "Performance Evaluation of Two Stage Scheduling Algorithm in Cloud Computing", British Journal of Mathematics & Computer Science Vol.6, No.3, pp.247-256.
- [3] P. Krishnam Raju Indukuri, P. Suresh Varma, Moses. G.J, 2012, "Performance measure of multi stage scheduling algorithm in cloud computing", International Conference Cloud Computing Technologies, Applications and Management (ICCCTAM), ISBN:978-1-4673-4415-9
- [4] Paul, M., Sanyal G., 2011, "Survey and analysis of optimal scheduling strategies in cloud environment", IEEE, World Congress on Information and Communication Technologies (WICT), pp. 789-792.
- [5] Jeyarani R, Ram R, Vasanth, Nagaveni N, 2009, "Design and Implementation of an Efficient Two-Level Scheduler for Cloud Computing Environment", IEEE, International conference on Advances in Recent Technologies in Communication and Computing, pp.884-886
- [6] Huang Qi-yi, Huang Ting-lei, 2010, "An optimistic job scheduling strategy based on QoS for Cloud Computing", IEEE, International Conference on Intelligent Computing and Integrating Systems (ICISS), pp. 673-675.
- [7] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, 2009, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing", IEEE, International Symposium on Parallel and Distributed Processing with Application, pp. 629-634.

- [8] Hao Li, Huixi Li, 2011, "A Research of Resource Scheduling Strategy for Cloud Computing Based on Pareto Optimality M×N Production Model", IEEE, International Conference on Management and Service Science(MASS), pp.1-5.
- [9] Kuan-Rong Lee, Meng-Hsuan Fu, Yau-Hwang Kuo, 2011, "A hierarchical scheduling strategy for the composition services architecture based on cloud computing", IEEE, The 2nd International Conference on Next Generation Information Technology (ICNIT), pp.163-169.
- [10] Gunho Leey, Byung-Gon Chunz, Randy H. Katzy,2011, "Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud", Proceedings of the 3rd USENIX Conference o Hot Topics in Cloud Computing, pp.4-4
- [11] Shu-Ching Wang, Kuo-Qin Yan, Shun-Sheng Wang, Ching-Wei Chen, 2011, "A Three-Phases Scheduling in a Hierarchical Cloud Computing Network", IEEE, Third International Conference on Communications and Mobile Computing(CMC), pp.114-117.
- [12] Peixoto M.L.M., Santana M.J., Estrella J.C., Tavares T.C., Kuehne B.T., Santana R.H.C.,2010, "A Meta scheduler architecture to provide QoS in the cloud computing", IEEE, 17th International Conference on Telecommunications(ICT), pp.650-657.
- [13] Yogita Chawla and Mansi Bhonsle, 2012"A study on scheduling methods in Cloud Computing", International Journal of Emerging Trends & Technology in Computer Science, Vol.1, Issue.3, pp.12-17.
- [14] Lipsa Tripathy, Rasmi Ranjan Patra, 2014, "Scheduling in Cloud Computing", International Journal on Cloud Computing Services and Architecture (IJCCSA), Vol. 4, No. 5, pp.21-27..
- [15] Bin Lin, P A Dinda and D.Lu, 2004,"User driven scheduling of interactive virtual machines", GRID'04 Proceedings of the 5th IEEE/ACM International Work shop on Grid Computing. Pages 380-387.
- [16] Arabi E.Keshk, 2014, "Cloud Computing Online Scheduling", Egypt, IOSR Journal of Engineering (IOSRJEN), Vol. 04, Issue 03pp.7-17.
- [17] Abdulrahman Alahmadi, Dunren che, Mustafa khaleel,, Michelle M.Zhu, Parsia Ghodous, 2015, "An innovative Energy aware cloud task Scheduling framework, proceedings of the IEEE 8th International conference on cloud computing", pp.493-500.
- [18] Shridhar G. Damanal and G. Ram Mahana Reddy, 2014, "Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines", IEEE, Sixth International Conference on Communication Systems and Networks (COMSNETS), pp.1-4.