International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

Optimized Job Scheduling Algorithm in Cloud Computing Environment

Shyam Sunder Pabboju, G.Nagi Reddy, P.Satya Shekar Varma, K.Nikhil Kumar

Abstract: There is a strong trend towards cloud technologies in the SME(Small and Medium Enterprises) sector, largely due to cost reduction, capex availability, and sometimes by collaboration features. Now a days so many startup companies are getting emerged and for them buying the resources for one time usage it becomes more expensive so they are preferring to using the cloud services(SLA) to overcome all this problems. In Cloud computing it essential to check weather a resource is available for allocating or not and allocating the jobs for the clients request is the big task. There are many job scheduling algorithm already proposed by the researchers to implement in cloud environment. After studying there algorithm we have came up with the most effective job scheduling algorithm. It is totally depending on Size and Arrival time of the job. By implementing our proposed algorithm we can obtain the better optimal solution to improve the overall performance of the system and gives more effective results compared to other job scheduling algorithms.

Keywords: cloud computing, job scheduling, performance, Quality of Service, virtualization.

I. INTRODUCTION

Cloud computing provides the resources over the internet with pay as you go assessing. Here there is no need of purchasing and maintaining the physical data centres and servers. We can directly access the technology services, such as computing power, storage and database, on an as needed basis from a cloud provider like Amazon Web Services, Microsoft Azure etc. Here we can select the services which is required for our business and only pay to those services and use it for low operating cost, run your infrastructure more efficiently and scale as your business needs change. The services offered in the cloud computing are software as a service (SaaS), Platform as a service (PaaS),
Infrastructure as a service (IaaS). Sometime it is referred as cloud computing stack because they are on the top of the other services like iaas, paas, saas.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
Shyam Sunder Pabboju*, Assistant Professor, CSE, JNTUH, MGIT, Hyderabad, India. shyampabboju@gmail.com
G.Nagi Reddy, Assistant Professor, CSE, JNTUH, MGIT, Hyderabad, India.nagireddymtech@gmail.com
P.Satya Shekar Varma, Assistant Professor CSE, JNTUH, MGIT, Hyderabad, India. satyashekarvarma@gmail.com
K.Nikhil Kumar, student, CSE, JNTUH, MGIT, Hyderabad. kasthurinikhilkumar13@gmail.com

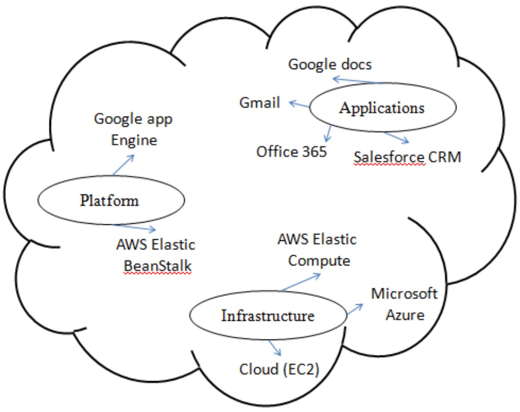

Fig 1.1 Cloud Environment Overview

In cloud computing for every client request for resources the provider provides the virtual machines so that the operator can use the resources from the back of beyond. To develop the job scheduling algorithm we should consider various criteria like resource utilization, cost, make span, user bandwidth, throughput etc. here I this paper we came with effective scheduling algorithm to improve the recital of resource allocation to the user request. All the requests are considered as unit time and the needs are updated automatically for every unit time based on the algorithm we proposed. In the results it shows the effective utilization of resources in unit time.

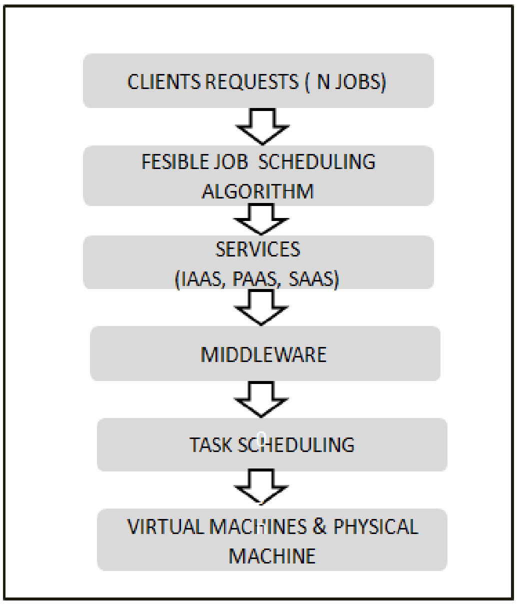

Fig 1.1 Proposed System

II. RELATED WORK

First work [1], They are dealing with the heuristic and ml for virtual machine allocation process and linking in terms of Qos, energy consumption, network. Their result shows the 24% improvement in servers to servers network and execution time. In Second Work [2], they have gone through the different job scheduling algorithm and taken the important aspects from different algorithm and built their machine to improve the runtime environment services. In third Work [3], They are estimating how many virtual machines are required for the client requests with a unit time and how CPU, memory is requires is analysed. In Fourth Work [4], They used the genetic algorithm to ensure the performance and scalability of the system. Every work has advantages and disadvantages with respect to performance and efficiency of the resource allocation.

III. OPTIMIZED JOB SCHEDULING ALGORITHM

We should consider various criteria while design the scheduling algorithm for cloud environment they are cost, resources utilization, make span, user bandwidth throughput.

In cloud for every clients requests per unit time is analysed. Here we are representing the job requests from the client as R1,R2,…,Rn and size of the job as the S1,S2,..,Sn and Arrival time of the job as A1,A2,..,An. Depends on the size of the job and Coming time of the job the job scheduling is taking place. All the requests for the resources in unit time is considered as Rt.

Now based on the size of each job the jobs are arranged in sequence order. If any two or more jobs having the same size then we will consider the arrival time of the job in some cases both jobs can have the same size and arrival time then as usual follows the FCFS process. After all the job are sorted in order the resources are mapped to the their request and each assigned is monitored whether the resource served the job or if it fails to request the SLA a server the job will be rescheduled again and allocate the available resource by the agent.

Algorithm :-

1 DO until all requests is completed Rn.
2 Compare the size of each job and Sort accordingly.
3 IF N[i] job size equals another job size.
4 THEN
        compare the Arrival time of the
    N jobs with same size
5 IF N job with same size
   THEN
      It follows the FCFS method
      All jobs pushed into queue.
6 ELSE
            Based on the time pushed into queue.
7 ELSE
   Pushed based on the job size.
8 Repeat for all N jobs in queue.
9 Get N[i] from queue.
10   Request the scheduler to check for the resources.
11   IF Resource is
      THEN
12   Allocate the Resource.
   ELSE
13 Go to [8] until All jobs
14 End

In the above mentioned algorithm the incoming request from client is taken as the inputs like Rq(x). Firstly we are sorting all the request based on size S(x) if m job having the same size then check the arrival time of the job based on the both constraints the jobs are enqueued into the queue once the queue is ready scheduler searches for the free resources and assign the resources to the job until all the jobs completed in queue.

IV. EXPERIMENTAL SETUP AND RESULTS

Cloud Sim is a framework made on java for designing and simulation of cloud computing infrastructure and services. It has a collection of classes representing the different features of the cloud environment. It models data centres, hosts, vm, cloudlets(Tasks).

There are two main areas in scheduling one is allocation of processors to the vm(vm scheduler) and second one is submission of cloudlets to the vm (cloudlet scheduler). To implement the proposed scenario we have to setup the cloudsim firstly in any of the java supporting IDE's like Intellij, Netbeans, Eclipse.

Then open the scheduler folder and create the new class and implement the base scheduler present in the cloud sim and make changes according to the proposed algorithm and run it first it creates the data centre and in each data centre vm are created with different id's.

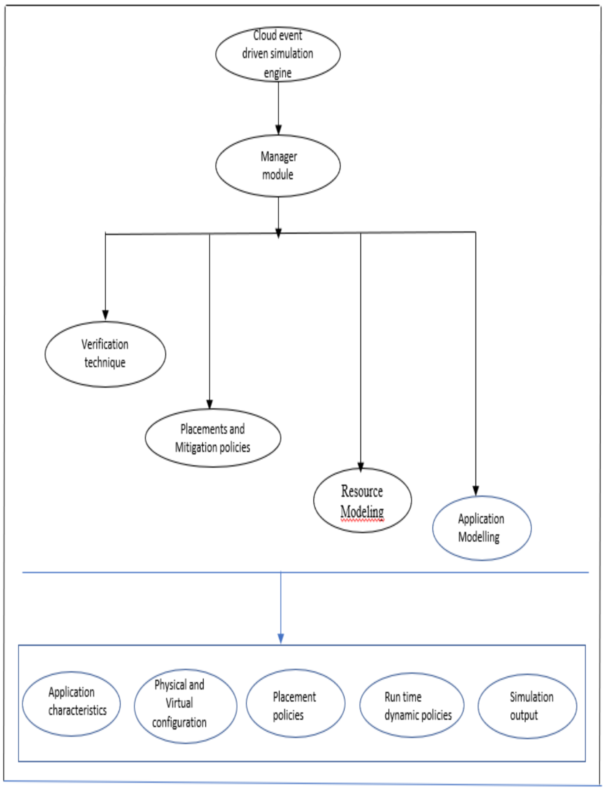

Fig 4.1 CloudSim Overview

Each vm created has it own CPU, RAM, Disk based on their size they comes under small(6 CPUS, RAM 6GB, Memory 650GB), large(15 CPUS, RAM 10GB, Memory 1TB), Extra large(25 CPUS, 20GB, Memory 1.25TB) configuration of setup of vm with starting time and finishing time of the jobs is considered.

Job ID	Task ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Depth
50	Stage-in	SUCCESS	2	0	0.11	0.1	0.21	0
0	1,	SUCCESS	2	7	13.36	0.21	13.57	1
7	8,	SUCCESS	2	6	13.45	0.21	13.66	1
3	4,	SUCCESS	2	5	13.59	0.21	13.8	1
4	5,	SUCCESS	2	4	13.71	0.21	13.92	1
1	2,	SUCCESS	2	3	13.76	0.21	13.97	1
2	3,	SUCCESS	2	2	13.83	0.21	14.04	1
5	6,	SUCCESS	2	0	13.88	0.21	14.09	1
6	7,	SUCCESS	2	1	13.88	0.21	14.09	1
21	22,	SUCCESS	2	9	10.91	13.8	24.72	2
35	36,	SUCCESS	2	6	11.11	13.66	24.78	2
8	9,	SUCCESS	2	7	11.2	13.57	24.78	2
26	27,	SUCCESS	2	11	10.96	13.92	24.89	2
18	19,	SUCCESS	2	0	11.03	14.09	25.12	2
14	15,	SUCCESS	2	13	11.25	13.97	25.22	2
24	25,	SUCCESS	2	12	11.39	13.92	25.31	2
19	20,	SUCCESS	2	1	11.42	14.09	25.51	2
33	34,	SUCCESS	2	8	11.74	13.8	25.54	2
25	26,	SUCCESS	2	10	11.62	13.92	25.54	2
16	17,	SUCCESS	2	17	11.62	13.97	25.59	2
22	23,	SUCCESS	2	4	11.71	13.92	25.63	2
13	14,	SUCCESS	2	16	11.7	13.97	25.67	2
15	16,	SUCCESS	2	19	11.65	14.04	25.69	2
11	12,	SUCCESS	2	5	11.9	13.8	25.7	2
20	21,	SUCCESS	2	15	11.76	13.97	25.73	2
9	10,	SUCCESS	2	14	11.79	13.97	25.76	2
17	18,	SUCCESS	2	3	11.83	13.97	25.8	2
10	11,	SUCCESS	2	2	11.8	14.04	25.84	2
32	33,	SUCCESS	2	18	11.8	14.04	25.84	2
31	32,	SUCCESS	2	9	11.79	24.72	36.51	2
28	29,	SUCCESS	2	7	11.75	24.78	36.52	2
30	31,	SUCCESS	2	6	11.76	24.78	36.53	2
29	30,	SUCCESS	2	11	11.75	24.89	36.63	2
12	13,	SUCCESS	2	0	11.67	25.12	36.79	2
23	24,	SUCCESS	2	13	11.67	25.22	36.89	2
34	35,	SUCCESS	2	12	11.67	25.31	36.98	2
27	28,	SUCCESS	2	1	11.65	25.51	37.16	2
36	37,	SUCCESS	2	0	2.27	37.16	39.43	3
37	38,	SUCCESS	2	0	4.2	39.43	43.63	4

Fig 4.2 Results in Shared Memory

For every job it creates unique job id, task id, status of every job, Data centre, vm id, time to complete the job, starting time of the job, finishing time of the job, depth. At the end it shows the total completion time of the job. If any task has the fail status it will rescheduled by the agent.

The algorithm proposed comes under the non pre-emptive scheduling, since we are not only focusing on the jobs which will be having some important like more value in binding at all. That will be taken for consideration in the future work. By applying our algorithm instead of the existing system, we could observe the improved some efficiency in terms of completion of jobs and execution time. After applying our algorithm the simulation shows the performance improvement in scheduling when compared to the standard algorithms. By seeing both the output we can tell that our algorithm is taking the less finishing time compared to the previously proposed algorithm. By this we can say that my algorithm is working more effectively compared to the another algorithm.

Job ID	Task ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Depth
50	Stage-in	SUCCESS	2	0	0.11	0.1	0.21	0
0	1,	SUCCESS	2	7	13.08	0.21	13.29	1
7	8,	SUCCESS	2	6	13.17	0.21	13.38	1
3	4,	SUCCESS	2	5	13.31	0.21	13.52	1
4	5,	SUCCESS	2	4	13.43	0.21	13.64	1
1	2,	SUCCESS	2	3	13.48	0.21	13.69	1
2	3,	SUCCESS	2	2	13.55	0.21	13.76	1
5	6,	SUCCESS	2	0	13.65	0.21	13.86	1
6	7,	SUCCESS	2	1	13.65	0.21	13.86	1
21	22,	SUCCESS	2	9	10.37	13.52	23.89	2
24	25,	SUCCESS	2	12	10.3	13.64	23.94	2
35	36,	SUCCESS	2	6	10.62	13.38	24	2
8	9,	SUCCESS	2	7	10.71	13.29	24	2
26	27,	SUCCESS	2	11	10.42	13.64	24.06	2
33	34,	SUCCESS	2	8	10.62	13.52	24.14	2
25	26,	SUCCESS	2	10	10.53	13.64	24.17	2
19	20,	SUCCESS	2	1	10.33	13.86	24.19	2
16	17,	SUCCESS	2	17	10.53	13.69	24.22	2
22	23,	SUCCESS	2	4	10.63	13.64	24.27	2
13	14,	SUCCESS	2	16	10.61	13.69	24.3	2
11	12,	SUCCESS	2	5	10.79	13.52	24.31	2
15	16,	SUCCESS	2	19	10.57	13.76	24.33	2
20	21,	SUCCESS	2	15	10.65	13.69	24.34	2
18	19,	SUCCESS	2	0	10.49	13.86	24.35	2
9	10,	SUCCESS	2	14	10.7	13.69	24.39	2
14	15,	SUCCESS	2	13	10.71	13.69	24.4	2
17	18,	SUCCESS	2	3	10.73	13.69	24.42	2
10	11,	SUCCESS	2	2	10.7	13.76	24.46	2
32	33,	SUCCESS	2	18	10.73	13.76	24.49	2
31	32,	SUCCESS	2	9	10.7	23.89	34.59	2
30	31,	SUCCESS	2	12	10.67	23.94	34.61	2
29	30,	SUCCESS	2	7	10.64	24	34.64	2
12	13,	SUCCESS	2	11	10.58	24.06	34.64	2
28	29,	SUCCESS	2	6	10.66	24	34.66	2
23	24,	SUCCESS	2	8	10.57	24.14	34.71	2
27	28,	SUCCESS	2	1	10.56	24.19	34.75	2
34	35,	SUCCESS	2	10	10.6	24.17	34.77	2
36	37,	SUCCESS	2	0	2.24	34.77	37.01	3
37	38,	SUCCESS	2	0	4.2	37.01	41.21	4

Fig 4.3 Results In Local Memory

The above two output simulates the average utilization of resources and the process with different number of client requests. It clearly denoted that our proposed scenario is boosting the performance of the resource utilization so that we can obtained the better performance improvement in over all scheduling process.

V. CONCLUSION AND FUTURE SCOPE

In this paper we are considering two factors one is incoming job size and time of arrival time of the jobs. Based on this considerations some values assigned for each job. Initially smaller size job will be taken first it includes some special cases in such case if two or more jobs having same size then we will consider arrival time .this process will be repeated for n jobs. Based on the values generated the jobs are sorted accordingly & stored into the queue. Jobs in the queue is taken one by one and allocate the resource if it is available. For the jobs same process is repeated until all the jobs are completed in the queue. At end we are achieving the improved performance in the scheduling process.

Future work we have to work on energy consumption in the cloud environment to improve existing mechanism and another one is performance when it comes to performance we are facing so many issues to work on it.

REFERENCES

1. Ali Pahlevan, Xiaoyu Qu, Marina Zapater andDavid Atienza" Integrating Heuristic and Machine-Learning Methods for Efficient Virtual Machine Allocation in Data Centers"IEEE transactions on computer-aided design of integrated circuits and systems.
2. Xiaomin Zhu, Yabing Zha, Ling Liu, and Peng Jiao" General Framework for Task Scheduling and Resource Provisioning in Cloud Computing Systems" IEEE Conferences Computer Software and Applications 2016 .
3. Mohammed Joda Usman , Abdul Samad Ismail1, Hassan Chizari and Ahmed Aliyu "Energy-Efficient Virtual Machine Allocation Technique Using Interior Search Algorithm for Cloud Datacenter" 978-1-5090-6255-3/17/$31.00 ©2017 IEEE.
4. Mugen Peng, Kecheng Zhang, Jiamo Jiang, JiahengWang, and WenboWang, "Energy-Efficient Resource Assignment and Power Allocation in Heterogeneous Cloud Radio Access Networks" IEEE transactions on vehicular technology, vol. 64, no. 11, november 2015.
5. J. Praveen chandar, Dr.A.Tamilaras, "The Feasible Job Scheduling Algorithm for Efficient Resource allocation Process in Cloud Environment" International Conference on Recent Trends in Advanced Computing (ICRTAC-CPS2018).
6. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: a Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software: Practice and Experience,vol. 41, no. 1, pp. 23-50, 2011.
7. M. Abdullah and M. Othman: Simullated Annealing Approach to Costbased Multi-Quality of Service Job Scheduling in Cloud Computing Environment, American Journal of Applied Sciences, Vol. 18, 2014, pp. 872-877.
8. B. Gomathi and K. Karthikeyan: Task Scheduling Algorithm Based on Hybrid Particle Swarm Optimization in Cloud Computing Environment, Journal of Theoretical and Applied Information Technology, Vol. 89, 2013, pp. 33-38.
9. U. Ayesta, M. Erausquin, E. Ferreira, and P. Jacko: Optimal Dynamic Resource Allocation to Prevent Defaults, Operations Research, Vol. 6, 2016, pp. 451-456.
10. Asha, N., &Rao, G. R. (2013, July). A Review on Various Resource Allocation Strategies in Cloud Computing. International Journal of Emerging Technology and Advanced Engineering, 3(7), 177-183. Retrieved March 10, 2014.

AUTHORS PROFILE

Shyam Sunder Pabboju, working as Assistant Professor, CSE , MGIT. He is currently pursuing Ph.D. in Osmania University, Hyderabad. He is life time member of CSI .He published several international and national journals and attended several conferences. He has 14 years of teaching experience and his areas of interest are cloud computing, machine learning, deep learning.

G. Nagi Reddy, working as Assistant Professor in MGIT. .He published several international and national journals and attended several conferences. He has 13 years of teaching experience and his areas of interest are machine learning , deep learning, image processing

Satya Shekar Varma Poranki , working as Assistant Professor, CSE MGIT. He is life time member of ISTE. He published several international and national journals and attended several conferences. He has 13 years of teaching experience and his areas of interest are Big data, image processing.

K.Nikhil Kumar pursuing final year B.Tech, CSE in MGIT..