International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-4, April 2019

Adaptive Scheduling Mechanism in Cloud

Manisha T. Tapale, R. H. Goudar, Mahantesh N. Birje

Abstract*: Cloud user expects its service requests to be served as early as possible. While cloud service providers (CSP) tries to serve user requests within deadline and obtain more profit. These two factors (deadline and profit) are conflicting - if only deadline is considered profit earned may be lesser; if only profit is considered few low profit jobs may starve for resources. So it is a challenging task for CSP to serve user requests considering these factors. This paper proposes an Adaptive Scheduling Mechanism which prioritizes the jobs based on their profit and deadline, and then orders them in priority queue for execution. On peak loads as new jobs with high profit arrive, some of the existing jobs having lesser profit in a priority queue suffer from starvation. In such context, the proposed mechanism adapts to generate another queue, which holds relatively lesser profit jobs from priority queue, and uses FCFS scheduling approach. Then the scheduler schedules jobs alternatively from multilevel queue - choosing one from priority queue and another from FCFS queue. This scheme returns more profit to CSP and also avoids starvation of jobs. The proposed work is simulated using CloudSim and it is observed that it performs better compared to existing work.

***Index Terms*: Cloud computing, Adaptive Scheduling, Service Provisioning, Deadline, Profit.**

I. INTRODUCTION

Cloud computing represents a huge volume of computing, communication, and storage resources as well as various software resources for providing different services to organizations and public. It has become the most essential technology in business, education, research community and other sectors as many cloud services are coming up day-by-day. Cloud computing provides everything as a service to the users [1] – Software as a service (SaaS), platform as a service (PaaS), and Infrastructure as a service (IaaS). Each service in a cloud computing is based on the pay-per-use of the resources. There are many Cloud Service Providers (CSPs) providing similar services, thus forcing them to provide best services with competitive costs to users. Also policy on SLAs, pricing, Service access control etc. will vary as they belong to different administrative domains. However, each CSP would like to offer services at possible higher price and also optimize the utilization of their resources. On the other hand, cloud users would like to obtain the services at possible lower price; thus leading to conflicting interests of both stakeholders of cloud, the CSP and the user [2, 3]. The CSP uses a scheduling mechanism to allocate the jobs to various resources in its pool. The scheduler has to schedule the jobs considering different requirements of the user and different constraints, such as size of the task, task execution time, availability of resources, and load on the resources. Some of the factors considered for scheduling mechanism are: deadline, profit , priority , FCFS, Max - Min, Min - Min etc [4]. The proposed work considers profit and deadline based priority scheduling. During the peak loads the scheduler adapts to a multilevel queue generation which uses priority and FCFS scheduling. This scheme returns more profit to the CSP as well as avoids starvation of jobs. The next Section presents the "Literature survey" which focuses on the literature study on scheduling mechanisms in cloud . The third section discusses the proposed work for "Adaptive Scheduling Mechanism" in detail and the algorithms for Adaptive Scheduling Scheme, Job Prioritizing, Adaptive Queuing Mechanism. Simulation section discusses the experimental description, parameters considered and tools used in simulation. "Results" section elaborates results obtained during simulation. Conclusion is given in "Conclusion" section.

II. LITERATURE SURVEY

This section provides the review of various works on resource pricing, resource brokering, and resource scheduling. A survey on cloud concepts, challenges and resource provisioning is given in [1] [5], which discusses the current status of resource provisioning methods and future directions. The work in [6] presents a taxonomy of task scheduling in Distributed Cloud Computing. Various works on scheduling and allocation of resources are given in [7] [8] [9] [10]. The work in [9] discusses scheduling of jobs and dispatching strategy. Task scheduling approaches are compared and discussed in the paper [10]. Genetic and evolutionary algorithm based scheduling is discussed in [11] [12]. Papers [13] [14] discuss about scheduling to minimize the makespan of jobs. There exist some priority based scheduling mechanisms [4] [15] [16] [17] [18]. These algorithms apply methods like analytical hierarchy process, Weighted Fair Scheduling, Round Robin Algorithm to prioritize the tasks and schedule. Various works are available in the literature which consider QoS parameters while scheduling [19] [20]. Some of the works focus on optimizing the parameters like energy, bandwidth, cost etc. Papers in [21][22] present works on energy aware task scheduling. The paper [23] presents work on bandwidth aware task scheduling, while [24] focuses on deadline based resource scheduling.

Manuscript published on 30 April 2019.
***** Correspondence Author (s)
Manisha T. Tapale*, Department of Computer Science and Engineering, KLE Dr. MSSCET, Belagavi, India.
R. H. Goudar, Center for Post Graduation Studies, Visvesvaraya Technological University, Belagavi, India.
Mahantesh N. Birje, Center for Post Graduation Studies, Visvesvaraya Technological University, Belagavi, India.

Trust based scheduling is mentioned in [25]. It is observed from the above discussions that though there exist many works on scheduling, there is a scope to schedule jobs adaptively considering deadline and profit.

III. ADAPTIVE SCHEDULING MECHANISM

Cloud users submit jobs to CSP with certain requirements (such as memory, computing power, bandwidth, etc.) and constraints (such as price, deadline, etc.). We consider that user's request is communicated to CSP through the broker for price negotiation. The block diagram of the proposed scheduling mechanism is shown in Fig. 1 .

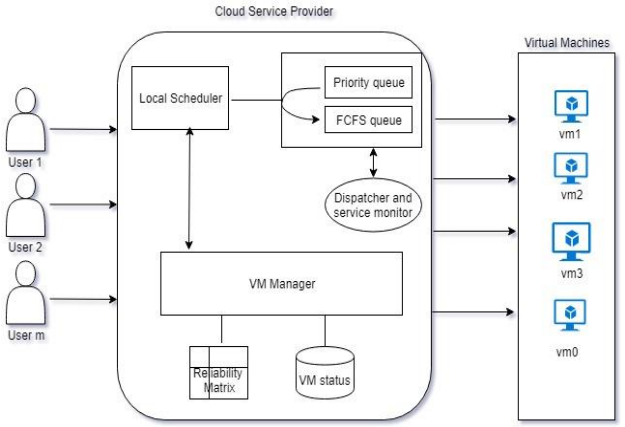

Fig. 1 : Block diagram of scheduling

The CSP has two main components - Local scheduler and a VM Manager. The local scheduler is responsible for scheduling the jobs to appropriate Virtual Machines. The scheduler refers the VM manager to check the reliability status of the various VM's . The VM Manager maintains a reliability matrix of all the VM's. The VM status is updated at appropriate time intervals by the VM Manager.

The local scheduler prioritizes and orders the jobs for execution based on their profit and deadline. For a given job set Algorithm 1 describes prioritizing and ordering of jobs.

Algorithm 1: Job Prioritizing Algorithm

Input: Job set J = { j_1, j_2.....j_n } from different users U ={ u_1, u_2, ...u_m }
Surplus S_{ji}
Deadline dl_{ji}

Output: Ordered (prioritized) job set J

1. for the job set J in a given time interval T={t_1,t_2.....t_p}
2. for each job j_i in the job set
3. Calculate the weightage W = { w_{j1},w_{j2}, w_{j3}.....w_{jn} } as W_{ji} = 0.6 S_{ji} × 0.4 dl_{ji}
4. end for
5. end for
6. order the jobs in an decreasing order of their weightage.

Job prioritizing algorithm takes a job set as an input from different users, along with the profit (*surplus*) S_{ji} for each job, and the *deadline* dl_{ji} in which it should complete the job. The surplus and deadline are given a weightage of 60% and 40% respectively. After all the weightages are calculated for the jobs in a given time interval, the jobs are ordered in decreasing values of their weightage in the priority queue for execution. For newly arriving jobs and/ or during peak loads in a cloud, there is a chance that most of the new jobs may provide a high surplus compared to the ones' already in the priority queue. The existing jobs in a priority queue may starve due to these new jobs. To avoid such starvation, scheduler adapts to generate multilevel queue. Algorithm 2 describes an Adaptive Queuing Mechanism to avoid starvation of jobs.

Algorithm 2: Adaptive Queuing Mechanism for Starvation Avoidance at Scheduler

Input: Ordered job set J
Output: Multilevel queue generation
Initialize: Q_{max} such that $Q_{max} > n$
Create a priority queue using a doubly linked list(DLL) with a single node

1. for each job j_i in the ordered job set J
2. insert job j_i in the priority queue
3. if queue size > th
4. if FCFS does not exists
5. create and insert the last job in priority queue to FCFS queue
6. else insert the job in Priority queue to FCFS queue
7. end for

The maximum size (Q_{max}) of the priority queue depends on the number of jobs (n). A threshold value *th* (80 percent of n) is considered . For every ordered job with a calculated weightage insert the job in the priority queue which is already ready as a part of the local scheduler for scheduling and check if the queue size has reached its threshold *th* . If a threshold is reached we create another level in the queue as an FCFS queue and insert the last job in the priority queue into the FCFS queue. There is a chance that most of the jobs entering the queue are providing high profit compared to the ones' already in the priority queue. At peak loads the queue adaptively generates a multilevel queue to avoid starvation of jobs. The jobs with lesser utility are shifted in the FCFS queue, hence providing fair scheduling of the jobs by giving equal chance of execution to jobs from each queue.

IV. SIMUATION AND RESULTS

Simulation of the proposed work is done using CloudSim tool. Number of VMs considered are 50. Some of the results obtained during simulation are compared with [4] and are described as below.

Profit Earned Vs. Number of Jobs: In [4] *price* is calculated based on the size of a job and cost per instruction (CPI) of a VM, and thus *profit* is calculated as the difference between the cost paid by the user and the actual cost of executing a task on a VM.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

We considered non cooperative bargaining game to decide a price of the resource and profit is calculated as the difference between the negotiated price and the reserved price. Fig. 2 shows the Profit Earned by a CSP versus number of jobs. It is observed that when there are lesser number of jobs the profit earned by [4] and proposed work is almost same. but as the number of jobs increase the profit earned by proposed work is more. This is because the bargaining protocol considers the market dynamics during price negotiation, whereas [4] depends on the size of the task only.

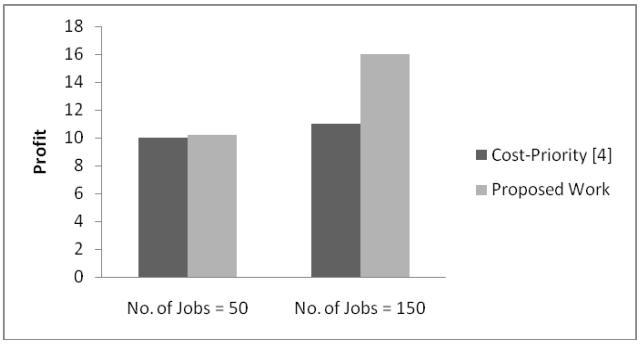

Fig. 2: Profit Earned Vs. Number of Jobs

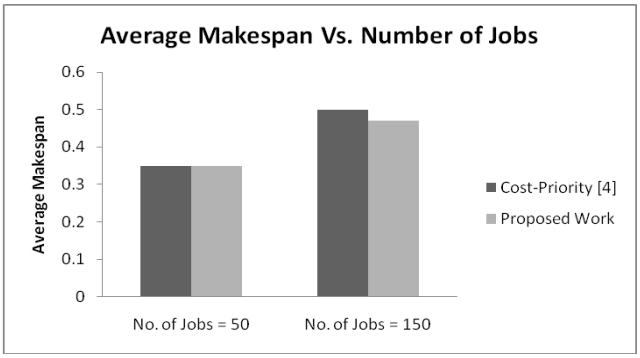

Fig. 3: Average Makespan Vs. Number of Jobs

Average Makespan versus Number of Jobs: Fig. 3 shows average makespan of the jobs considering 50 VMs. It is observed that when there are lesser number of jobs the makespan of both the proposed work and [4] is same. But when there are more number of jobs the makespan of the proposed work is slightly lesser compared to [4]. This is because the proposed scheme adapts to generate a multilevel queue at peak loads with one queue scheduling using priority and second level queue schedules using FCFS. this helps in earning more profit to the CSP, as well as avoids starvation and maintains fairness by switching between the queues, hence giving equal opportunity of scheduling for completing all the jobs within a given deadline. Whereas in [4] the multilevel queue uses aging mechanism to avoid starvation, hence the average makespan is more compared to the proposed work.

Job Execution Rate at Different Scenarios: We also compare our simulation results with [20] considering two scenarios as follows:

Scenario 1: when there are lesser number of jobs (under load) , among which majority are of high priority

Scenario 2: when there are more number of jobs (peak load) , in which equal number of high and low priority jobs exists.

Fig. 4 shows job execution rate for two scenarios as mentioned above. It is observed that in Scenario 1 and 2 the job execution rate of [20] and proposed work are almost same . Hence the successful execution of jobs of the proposed work matched with that of [20].

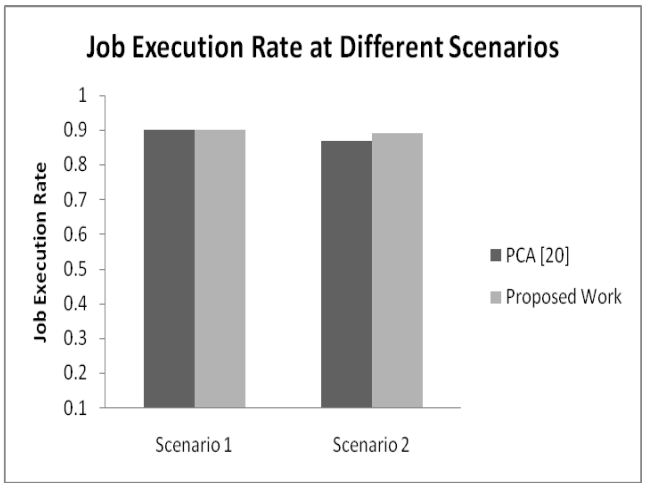

Fig. 4: Job Execution Rate at Different Scenarios

V. CONCLUSION

The proposed work considered profit and deadline based priority scheduling. The proposed scheme adapts to generate a multilevel queue at peak loads with one queue scheduling using priority and second level queue schedules using FCFS.

This helps in earning more profit to the CSP, as well as avoids starvation and maintains fairness by switching between the queues. Hence it gives an equal opportunity of scheduling for completing all jobs within a given deadline. The proposed work performs better compared to the existing works.

REFERENCES

1. Mahantesh N. Birje, Praveen Challagidad, R. H. Goudar, Manisha Tapale, "Cloud Computing Review: Concepts, Technology, Challenges and Security", International Journal of Cloud Computing (IJCC), Inderscience, Vol. 6, No. 1, pp. 32-57, 2017.
2. Goudar, R.H., Tapale. M.T., Birje, M. N., "Price negotiation for cloud resource provisioning", International Conference On Smart Technology for Smart Nation, SmartTechCon 2017.
3. M. N. Birje, S. S. Manvi, Chetan Bulla, Economical Job Scheduling in Wireless Grids, Third Int. conference on Electronics Computer Technology, ICECT 2011, Kanyakumari, India.
4. S.Y.Amdani, S.R.Jadhao, "Novel Hybrid Cost-Priority Based Scheduling in Cloud Environment", IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2016), December 23-25, 2016, Jaipur, India.
5. Singh S, Chana I, "Cloud resource provisioning: survey, status and future research directions", Knowledge Information Systems, 2016, Vol. 49, Issue 3.
6. Casavant, T., Kuhl, J. G, "A Taxonomy of Scheduling in General-purpose Distributed Computing Systems", IEEE Transactions on Software Engineering, Vol.14, No. 2, pp. 141-154, 1988.

7. Ergu D, Kou G, Peng Y, Shi Y, Shi Y, "The analytic hierarchy process: Task Scheduling and Resource Allocation in Cloud Computing Environment", The Journal of Supercomputing. 64(3): 835-848, 2013.
8. Maguluri ST, Srikant R, "Scheduling jobs with unknown duration in clouds", IEEE/ACM Trans Netw (TON) 22(6):1938–1951, 2014.
9. Tai-Lung Chen et al, Scheduling of Job Combination and Dispatching Strategy for Grid and Cloud System, GPC,(2010) 612–621.
10. A. Jain and R. Kumar, "A Comparative Analysis of Task Scheduling Approaches for Cloud Environment," International Conference On Computing for Sustainable Global Development, 2016, pp. 2602-2607.
11. Tsai J-T, Fang J-C, Chou J-H, "Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm", Comput Oper Res 40(12):3045–3055, 2013.
12. Singh S, Kalra M. Scheduling of independent tasks in cloud computing using modified genetic algorithm. In: Computational intelligence and communication networks (CICN), 2014 international conference on .IEEE; 2014. p. 565e9.
13. Bhoi, U., Ramanuj, P. N, "Enhanced Max-Min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol. 2, No. 4, 259-264, 2013.
14. Liu, G., Li, J., Xu, J, "An Improved Min-Min Algorithm in Cloud Computing", International Conference of Modern Computer Science and Applications. Springer, Wuhan, China, 47-52, 2012.
15. Ghanbari, S., Othman, M, "A Priority-based Job Scheduling Algorithm in Cloud Computing", International Conference on Advances Science and Contemporary Engineering (ICASCE). Jakarta, Indonesia, 778-785, 2012.
16. Ghanbari S, Othman M, Bakar MRA, Leong W. J, "Priority-based divisible load scheduling using analytical hierarchy process", Appl Math Inf Sci 9(5):25–41, 2015
17. Li Yang et al, "A new Class of Priority-based Weighted Fair Scheduling Algorithm", Physics Procedia, 33, pp. 942 – 948, 2012 .
18. Deepika Saxena, RK Chauhan, and Ramesh Kait, "Dynamic fair priority optimization task scheduling algorithm in cloud computing: Concepts and implementations", International Journal of Computer Network and Information Security, 8(2): 41, 2016.
19. Wu, X., Deng, M., Zhang, R, Zeng, B., Zhou, S, "A Task Scheduling Algorithm Based on QoS-Driven in Cloud Computing", International Conference on Information Technology and Quantitative Management. Elsevier Procedia, Suzhou, China, 1162-1169, 2013.
20. Tamilselvan L, Anbazhagi, Shakkeera, "Qos based dynamic task scheduling in Iaas cloud", International Conference on Recent Trends in Information Technology (ICRTIT), 2014 International Conference on. IEEE; 2014.
21. Cheng C, Li J, Wang Y, "An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing", Tsinghua Sci Technol 20 (1):28–39, 2015
22. Zhu X, Yang LT, Chen H, Wang J, Yin S, Liu X, "Real-time tasks oriented energy-aware scheduling in virtualized clouds", IEEE Transactions on Cloud Computing 2(2):168–180, 2014.
23. Lin W, Liang C, Wang JZ, Buyya R, "Bandwidth-aware divisible task scheduling for cloud computing", Software: Practice and Experience 44(2):163–174, 2014.
24. Rodriguez MA, Buyya R, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds", IEEE Transactions on Cloud Computing 2(2):222–235, 2014
25. Wei Wang, "Cloud-DLS: Dynamic Trusted Scheduling for Cloud Computing", Expert Systems with Applications 39 pp. 2321–2329, 2012.

AUTHORS PROFILE

Manisha T. Tapale, currently working as an Assistant Professor, Dept of CSE , KLE MSSCET, Belagavi. She has a 13 years of Teaching Experience. She has published papers in International Journals and Conferences. Her subjects of interest are Cloud Computing, Computer Networks, Network Security and Operating Systems.

R H Goudar, currently working as an Associate Professor, Dept. of CNE, Visvesvaraya Technological University, Belagavi.He has 14 years of Teaching Experience at Professional Institutes across India. He worked as a faculty at International Institute of Information Technology, Pune for 4 years and at Indian National Satellite Master Control Facility, Hassan, India. He published over 130 papers in International Journals, Book Chapters and Conferences of High Repute. His Subjects of Interest include Semantic Web, Network Security and Wireless Sensor Networks.

Mahantesh N. Birje received B.E. and M.Tech. degrees in Computer Science and Engineering in 1997 and 2005 respectively. He obtained Ph.D. from Visvesvaraya Technological University (VTU), Belagavi, India in 2012. His current research areas include Cloud Computing, Internet of Things, Data Mining, and Security. He has published many research papers in International refereed journals and Conferences. He is a Reviewer of some International Journals of IEEE, Elsevier, Springer, etc. He has given few invited Lectures and has organized Workshops and Seminars for Faculty and Students. He has executed various academic and administrative responsibilities. Currently he is working as Professor in the Center for Post Graduate Studies, VTU, Belagavi.