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

QoS Aware Resource Scheduling and Performance Assessment of Heuristics for Processing Jobs on Cloud

Krunal Vaghela, Amit Lathigara, Paresh Tanna

Abstract*: *Cloud computing or in other words, shared computing is a unique way of sharing resources via the Internet. It combines and extends features of parallel processing, grid computing, and distributed computing. Cloud Computing environments provide a competent way to schedule and process various jobs on remote machines. Rather than relying on local machines, Cloud users access services remotely via high-speed networks. Various users submitting jobs to be processed to Cloud would expect Quality of Service (QoS). So, currently, many researchers are proposing various heuristics that provide QoS to cloud users. The job scheduler is responsible for scheduling various jobs to its best-matched resource to achieve desired QoS. There are Service Level Agreements (SLAs) between Cloud Service Providers (CSPs) and Cloud users, which need to be followed by both the parties. Benefits would be affected in case of not complying with SLAs. In this paper various SLAs like Hard SLA, Best Effort SLA and Soft SLA are proposed. Jobs with required QoS parameters like Reliability, Execution Time and Priority are submitted to the scheduler. QoS of resources is determined by parameters like Reliability, Job Completion Time and the Cost of the resource. Schedulers then assign the Job to the best-matched resource according to specified SLA. Simulation is performed for First Fit and Best Fit heuristic approaches. Performances of both the heuristic approaches are evaluated with performance parameters like Average Resource Utilization (ARU), Success Rate of Jobs (SR) and Total Completion Time (TCT). This research work is useful for various organizations that provide various Cloud services to users who seek different levels of QoS for various applications.

Keywords: Average Resource Utilization (ARU), Best Fit (BF), First Fit (FF), Success Rate of Jobs (SR), Total Completion Time (TCT)

I. INTRODUCTION

Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers.

Cloud provides an efficient way to store, process and analyze huge amounts of data. Apart from that Cloud provides various services like Infrastructure Services (IAAS), Software Services (SAAS) and Platform Services (PAAS) [16]. By using the cloud, users can have the tremendous computing power and huge storage capacity online. Cloud users can make use of any platform, software, hardware, operating system, processing power, etc. without purchasing it [16]. In short, you can have resources on rent. By doing these, unutilized infrastructure and other IT resources can be utilized by paying a minimal cost. Cloud Computing introduces much diversity as various resources are owned by different service providers who are also located at different remote locations. Apart from this, Cloud users are also located at different locations. Cloud users and Cloud service providers are connected by high-speed networks. In such a scenario, providing Quality of Service (QoS) is a big research issue [17]. In such a diversified scenario, where expectations of Cloud users and Cloud Service Providers are varying, achieving the desired QoS would be crucial. In the case of static scheduling, jobs would get their resources to get executed before the execution. In the case of dynamic scheduling, the scheduler keeps allocating the resources to the job as a job keeps arriving. In batch processing, generally static scheduling is preferred and in real-time applications generally, dynamic scheduling is preferable. Apart from these various applications also demand varying services. So, with virtualized cloud resources, it becomes very challenging to satisfy the requirements of every application as well [17]. The scheduler will play a very crucial role here. The performance of each scheduling heuristic is directly proportional to effective scheduling. In this paper, we have proposed various scheduling approaches for job processing in the Cloud Computing Environment which satisfies various criteria of QoS. We have also evaluated scheduling heuristics concerning various performance parameters related to QoS.

II. RELATED WORK

Many researchers have proposed QoS guided scheduling heuristics with different QoS parameters. The summary for the same has been presented in Table I below.

Revised Manuscript Received on April 30, 2020.
* Correspondence Author
Krunal Vaghela*, Research Scholar, Faculty of Technology, RK University, Rajkot, India.& Assistant Professor, Marwadi University, Rajkot, India. Email: krunal.vaghela910@gmail.com
Dr. Amit Lathigara, Computer Engineering, Marwadi University, Rajkot, India. Email: amit.lathigara@marwadieducation.edu.in
Dr. Paresh Tanna, MCA, RK University, Rajkot, India. Email: paresh.tanna@rku.ac.in

Table I. Summary of Various Scheduling Heuristics

Sr. No	Job Scheduling Heuristics	Scheduling (QoS) Parameters	Findings
1	A Particle Swarm Optimization-based Heuristic for Scheduling [5]	Execution Time, Resource Utilization	Better Resource Utilization. Fair workload distribution among resources.
2	Improved Cost-Based Algorithm [3]	Latency, makespan	Better makespan with more numbers of VMs. Overall improved latency.
3	Innovative transaction-intensive cost-constraint scheduling algorithm [6]	Execution Time, Cost	Lesser cost for few situations. Permits the negotiations of completing cost and time.
4	SHEFT workflow scheduling algorithm [7]	Execution Time, Scalability	Optimized Execution Time. Runtime Scaling of Resources.
5	Job Scheduling based on Berger Model [8]	Job Completion Time, Fairness	Meeting the user's expectations. Better Job Execution Time.
6	Multiple QoS Constrained Scheduling Strategy of Multi-Workflows [4]	Makespan, the success rate of jobs, execution time	Improved job execution time. Dynamic scheduling of jobs.
7	Task scheduling algorithm based on QoS-driven in cloud computing [9]	Total completion time, Priority	Fair Scheduling for balancing the load. Better performance as far as the total completion task is concerned.
8	PAPRIKA [2]	Complication Time, Utility	Enhanced resource utility. Taking less time for task allocation to resources.
9	Ant Colony Optimization (ACO) [10]	Execution Time, Cost	Improved execution time. Enhanced cost as far as resource utilization is concerned
10	Multi QoS Scheduling Algorithm [11]	Execution Time	Improved efficiency up to some extent.

III. SIMULATION AND PROPOSED METHODOLOGY

Definition of Job QoS Parameters

Each job that is submitted by the user is having QoS requirements in terms of the following parameters.

Reliability (RE (j)): Minimum probability of successful completion of the job in the first attempt.

Execution Time (ET (j)): The time difference between the submission and completion of a job.

Priority (PR (j)): Relative importance of the task specified by the user submitting the job.

Definition of Resource QoS Parameters

Each resource owned by the service provider and is having QoS parameters by which the quality of the resource is determined.

Reliability (RE (r)): Ratio of successful no of jobs executed at the first attempt to total no of jobs.

Completion Time (CT (r)): Execution time of the job.

Cost (CO (r)): Cost of execution of the job. Directly proportional to the priority of the job.

Service Functions

To determine the requirements of the job in terms of required QoS parameters, we have proposed a simple service function of a job which is SF (j) given below. Equation (1) and (2) represents that the service function of job SF (j) is a total of all 3 QoS parameters of the job which are described above. The total of all 3 parameters should be 1.

$$SF(j) = f(RE(j), ET(j), PR(j)) \quad (1)$$

$$f(RE(j), ET(j), PR(j)) = (RE(j) + ET(j) + PR(j)) \quad (2)$$

Where:
$0 <= RE(j) <= 1$
$0 <= ET(j) <= 1$
$0 <= PR(j) <= 1$
$RE(j) + ET(j) + PR(j) = 1$

To determine services provided by the resource in terms of QoS parameters, we have proposed simple service functions of the resource which is SF(r) given below. Equation (3) and (4) represents that the service function of resource SF (r) is a total of all 3 QoS parameters of the resource which are described above. The total of all 3 parameters should be 1.

$$SF(r) = f(RE(r), CT(r), CO(r)). \quad (3)$$

$$f(RE(r), ET(r), CO(r)) = (RE(r) + CT(r) + CO(r)) \quad (4)$$

Where:
$0 <= RE(r) <= 1$
$0 <= CT(r) <= 1$
$0 <= CO(r) <= 1$
$RE(r) + CT(r) + CO(r) = 1$

Service Level Agreements (SLAs)

To determine how much the job requirements should be matched with the resource capabilities, we have defined three Service Level Agreements (SLAs) given below.

Retrieval Number: F4638049620/2020©BEIESP
DOI: 10.35940/ijitee.F4638.049620
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Hard-SLA: QoS description with this level has the strongest restraint. The description of the task must be met, otherwise the scheduling is invalid. It means SF (J) <= SF(R) then the only job is scheduled otherwise it will be rejected and kept in the unallocated job queue. As per equation (5) below, if the reliability of resource is greater than or equal to reliability of job, and execution time for that particular job on that resource is less than or equal to the completion time of that particular job, and cost of the job is greater than or equal to the priority of resource then the only job is scheduled. Otherwise, the job is rejected.

$$SF(r) = SF(j) \text{ if } \{(RE(r) >= RE(j)) \cap (ET(r) <= CT(j)) \cap (PR(r) >= CO(j))\} \quad (5)$$

Best Effort SLA: QoS description doesn't need a very strong constraint. The job will obtain the biggest benefit if such descriptions are met. Otherwise, the scheduling is not considered invalid, just benefit may be affected. As per equation (6), if the total of Reliability, Completion time and Cost of the resource is greater than Reliability, Execution Time and Priority of the job then only scheduling is valid otherwise the job is rejected.

$$SF(r) \geq SF(j) \text{ if } \{(RE(r) + CT(r) + CO(r)) >= (RE(j) + ET(j) + PR(j))\} \quad (6)$$

Soft-SLA: If the requirement of the job is partially met then also scheduling is valid. Otherwise scheduling is invalid, and the job will be kept in the unallocated queue. Here jobs will do maximum compromise for its requirements. As per equation (7) below, if the reliability of resource is greater than or equal to half of the reliability of job, and execution time for that particular job on that resource is less than or equal to half of the completion time of that particular job, and cost of the job is greater than or equal to half of the priority of resource then the only job is scheduled. Otherwise, the job is rejected.

$$SF(r) \geq SF(j)/2 \text{ if } \{(RE(r) \geq (RE(j)/2) \cap (ET(r) \leq CT(j)/2) \cap (PR(r) \geq CO(j)/2)\} \quad (7)$$

Proposed Scheduling Heuristics

We have proposed two approaches for the allocation of jobs to resources for execution.

First Fit (FF): We have studied the traditional First Fit approach [15] and modified it according to the requirement of our approach. As per the below algorithm, step 2 and step 3 will initialize all performance parameters of all jobs and all available resources and set it to 0. Step 4 will initialize the flag variable to 0. In the steps 5 to 8, the scheduler will pick a job from the queue and match the QoS requirements of the job with the capacity of the first available resource as per defined SLA. As per step 9 to 11, if resource which can satisfy QoS requirements of the job is found then the resource will be allocated to that job for execution and resource will be removed from the list of available resources. As per step 12, the Flag variable will set to 1 which indicates the job is scheduled. If the first picked resource cannot fulfill QoS requirements of the job then the scheduler will check the next available resource and so on. As per steps 16 to 18, if the scheduler cannot find any resource which can satisfy QoS requirements of the job then the job will be kept in the unallocated job queue. The scheduler will repeat this process for all jobs waiting in the queue. After finishing all waiting jobs in the queue, the scheduler will process all unallocated jobs again to find available resources for execution. Below is the algorithm of First Fit heuristic.

1. **Begin**
2. Initialize the list of user jobs as J = {j1, j2, j3, …., jn}
3. Initialize the list of VM resources as R = {r1, r2, r3, …., rm}.
4. flag = 0
5. **For** i = 1 to n do
6.   **Calculate** SF (ji)
7.   **For** j = 1 to m do
8.   **Calculate** SF (rj)
9.   **if** SF (ri) = SF (ji)
10.   **Allocate** (rj, ji)
11.   Remove RJ from list
12.   flag = 1.
13.   **Break Loop**
14.   **end if**
15.   **end For**
16. **if flag =** 0
17.   Put the job in the unallocated queue
18. **end if**
19. **end For**

If the queue is empty, then repeat the allocation process for the unallocated queue.

Best Fit (BF): We have studied the traditional Best Fit approach [15] and modified it according to the requirement of our approach. Our approach is almost the same as our modified First Fit. As per steps 5 to 7, the requirements of all the jobs are calculated. Then as per steps 8 to 10, the capacity of all the resources is calculated. Now, here in this approach as per steps 11 to 15, the scheduler will scan all the available resources first before allocation. The scheduler will select resources with the best matching parameters and then it will allocate the best resource to the job. It means resources with the nearest value of SF(r) to SF (j) will be selected. Best Fit heuristic works as per the below steps.

1. **Begin**
2. Initialize the list of user jobs as J = {j1, j2, j3, …., jn}
3. Initialize the list of VM resources as R = {r1, r2, r3, …., rm}.
4. flag = 0
5. **For** i = 1 to n do
6.   **Calculate** SF (ji)
7. **end for**
8. **For** j = 1 to m do
9.   **Calculate** SF (rj)
10. **end for**
11. **For** i = 1 to n do
12.   **Compare** SF (ji) with SF (r) of each resource
13.   **Allocate** ji to best match resource
14.   **Remove** that resource from the list
15. flag = 1
16. **end for**
17. **if flag =** 0
18.   Put the job in the unallocated queue
19. **end if**

If the queue is empty then repeat the allocation process for the unallocated queue.

Here in both the heuristics, n is the Total number of Jobs, m is the Total number of Virtual Machines (Resources), J is set of tasks j1, j2, j3… jn arrived at any time interval and R is set of available VM resource r1, r2, r3,….., rm.

Definition of scheduling heuristic performance parameters

Average Resource Utilization (ARU): This is the ratio of running time of processor of resource to total time. Total time also includes an idle time of the processor.

Success Rate of Jobs (SR): It is a ratio of total successfully executed jobs in the first attempt to total no of jobs.

Total Completion Time (TCT): It is the total completion time of all successfully scheduled jobs. It also includes waiting time in the queue.

Simulation Environment

Table II Simulation Environment

Sr. No.	Item Name	Value
1	Hardware	Core i5 2.3 GHz, 8 GB RAM
2	Operating System	Windows 8
3	Simulator	CloudSim 3.0.3
4	Platform	Eclipse
5	Simulation Environment 1	Tasks: 5000 Resources: 2000
6	Simulation Environment 2	Tasks: 5000 Resources: 1500
7	Simulation Environment 3	Task : 5000 Resources: 1000

IV. RESULT DISCUSSION

Table III SLA wise performance evaluation for Best Fit Algorithm in Simulation Environment 1 to 3

Simulation Environment – 1 – Best Fit (BF)			
	ARU	SR	TCT
Hard - SLA	0.52	0.63	0.85
Best Efforts- SLA	0.59	0.71	0.82
Soft -SLA	0.72	0.85	0.77
Simulation Environment – 2 – Best Fit (BF)			
	ARU	SR	TCT
Hard - SLA	0.46	0.58	0.89
Best Efforts- SLA	0.54	0.68	0.86
Soft -SLA	0.65	0.82	0.83
Simulation Environment – 3 – Best Fit (BF)			
	ARU	SR	TCT
Hard - SLA	0.41	0.53	0.93
Best Efforts- SLA	0.49	0.59	0.92
Soft -SLA	0.54	0.75	0.95

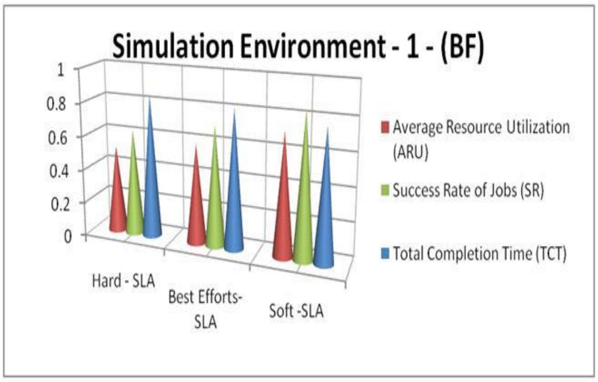

Figure 1. SLA wise performance evaluation for Best Fit Algorithm in Simulation Environment 1

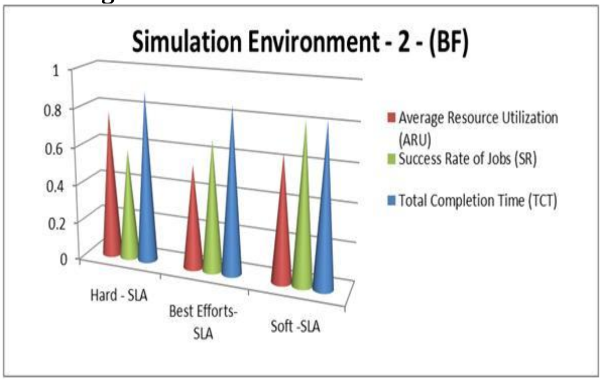

Figure 2. SLA wise performance evaluation for Best Fit Algorithm in Simulation Environment 2

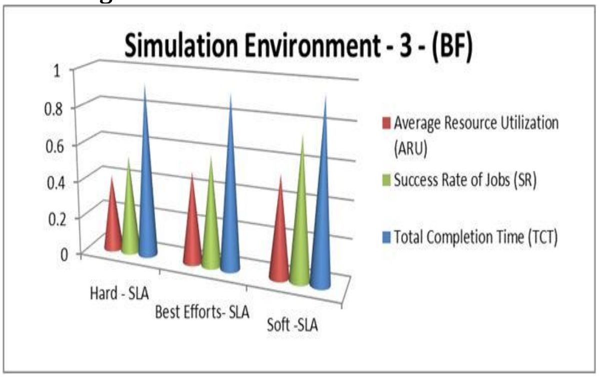

Figure 3. SLA wise performance evaluation for Best Fit Algorithm in Simulation Environment 3

Table III and Figure 1 to Figure 3 are showing SLA wise performance evaluation of Best Fit heuristic concerning all three performance parameters in all three Simulation Environments. For all three simulation environments, Average Resource Utilization (ARU) is increasing while moving from Hard –SLA to Best Effort – SLA to Soft – SLA because more number of jobs will get scheduled due to weaker restrain of SLA which will keep resources more utilized. In the same context, the Success Rate of Jobs (SR) will also improve due to the weaker restrain of SLA and more jobs will get the chance to be scheduled. Total Completion Time (TCT) is decreased as jobs need not wait in the waiting queue and get its turn easily as SLAs getting weaker. In the Simulation environment wise comparison, if we move from Simulation Environment 1 to Simulation Environment 3 then available resources are less which will harm all three performance parameters.

Retrieval Number: F4638049620/2020©BEIESP
DOI: 10.35940/ijitee.F4638.049620
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table IV SLA wise performance evaluation for First Fit Algorithm in Simulation Environment 1 to 3

Simulation Environment – 1 – First Fit (FF)			
	ARU	SR	TCT
Hard - SLA	0.50	0.60	0.81
Best Efforts- SLA	0.56	0.69	0.76
Soft -SLA	0.70	0.81	0.70
Simulation Environment – 2 – First Fit (FF)			
	ARU	SR	TCT
Hard - SLA	0.45	0.54	0.85
Best Efforts- SLA	0.51	0.62	0.81
Soft -SLA	0.60	0.77	0.77
Simulation Environment – 3 – First Fit (FF)			
	ARU	SR	TCT
Hard - SLA	0.39	0.50	0.91
Best Efforts- SLA	0.44	0.56	0.89
Soft -SLA	0.50	0.70	0.85

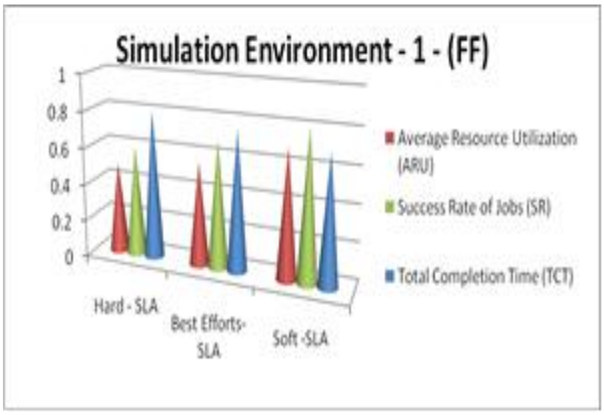

Figure 4. SLA wise performance evaluation for the First Fit Algorithm in Simulation Environment 1

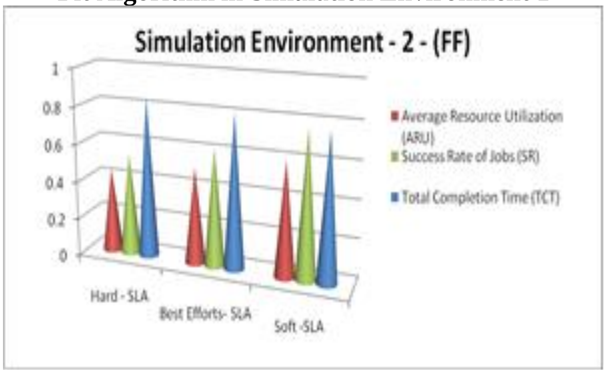

Figure 5. SLA wise performance evaluation for the First Fit Algorithm in Simulation Environment 2

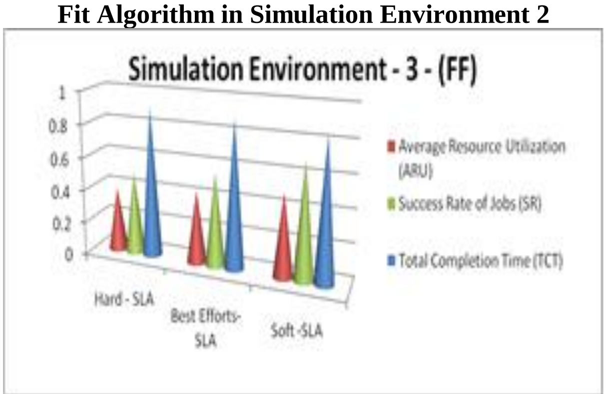

Figure 6. SLA wise performance evaluation for the First Fit Algorithm in Simulation Environment 3

Table IV and Figure 4 to Figure 6 are showing SLA wise performance evaluation of First Fit heuristic concerning all three performance parameters in all three Simulation Environments. The effects of the SLAs and Simulation Environments on the results are identical to Best Fit heuristic that we have just discussed. Now, we will compare both First Fit and Best Fit heuristics in all simulation environments. In all three simulation environments, Average Resource Utilization (ARU) in Best Fit heuristic is slightly better than First Fit because the job will find the best resource which can satisfy the requirements of the job. Success Rate of Job (SR) is also slightly better in Best Fit heuristic than the First Fit because the job will find the most suitable resource to get executed. But Total completion Time (TCT) is better in the First Fit that the Best Fit in all three Simulation Environments. Best Fit heuristic will scan the entire pool of available resources for each job, which will cause a delay in the scheduling of the job. So the Best Fit gives better results for ARU and SR but First Fit will perform better in case of TCT of the job.

V. CONCLUSION

In this paper, we have simulated First Fit and Best Fit heuristics with various SLAs like Hard SLA, Best Effort SLA, and Soft SLA. Jobs with required QoS parameters like Reliability, Execution Time and Priority are assigned to the best-matched resource according to heuristic and SLA. Quality of resource is determined by parameters like Reliability, Job Completion Time and the Cost of the resource. Performances of both the heuristic approaches are evaluated with performance parameters like Average Resource Utilization (ARU), Success Rate of Jobs (SR) and Total Completion Time (TCT). Best Fit heuristic will perform slightly better than the First Fit for ARU and SR. But First Fit will perform better for TCT in all simulation environments due to the early allocation of resources.

The highest number of jobs will get scheduled in Soft SLA due to the weakest restrain and Lowest in the case of Hard SLA due to the hardest restrain which will cause an increase in ARU, SR while the decrease in TCT. So, Soft SLA will perform better for both heuristics and in all simulation environments but Job will have to compromise the most in Soft SLA while Hard SLA will not perform that much good compare to Soft SLA as far as the number of jobs which will get scheduled is concerned. But in Hard SLA jobs will have to do the lowest compromise for their QoS requirements. Performance of Best Effort SLA is in between Hard SLA and Soft SLA for both the heuristics.

REFERENCES

1. Huifang Li, Siyuan Ge, Lu Zhang "A QoS-based Scheduling Algorithm for Instance-intensive Workflows in Cloud Environment"26th Chinese Control and Decision Conference (CCDC) 978-1-4799-3708-0/14 IEEE 2014.
2. Hilda Lawrance, Dr. Salaja Silas, Efficient QoS Based Resource Scheduling Using PAPRIKA Method for Cloud Computing. International Journal of Engineering Science and Technology (IJEST) vol. 5(3) (2013).

3. Cui Lin, Shiyong Lu," Scheduling Scientific Workflows Elastically for Cloud Computing" in IEEE 4th International Conference on Cloud Computing, (2011).
4. Y. Yang, K. Liu, J. Chen, X. Liu, D. Yuan, and H. Jin, An Algorithm in SwinDeW-C for Scheduling Transaction-Intensive Cost-Constrained Cloud Workflows, Proc. of 4th IEEE International Conference on e-Science, 374-375, Indianapolis, USA, December 2008.
5. C. Lin, S.Lu, "Scheduling Scientific Workflow Elasticity for Cloud Computing", IEEE 4th International Conference on Cloud Computing, pp. 246-247, (2011).
6. Hongbo Yu, Yihua Lan*, Xingang Zhang, Zhidu Liu, Chao Yin, Lindong Li" Job Scheduling Algorithm In Cloud Environment" International Conference on Computational and Information Sciences IEEE, (2013).
7. Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou. A task scheduling algorithm based on QoS-driven in Cloud Computing. Information Technology and Quantitative Management (ITQM2013). pp.1162-1169 (2013).
8. Kapil Kumar, Abhinav Hans, Ashish Sharma, Navdeep Singh, Towards the Various Cloud Computing Scheduling Concerns: A Review, International Conference on Innovative Applications of Computational Intelligence on Power, Energy, and Controls with their Impact on Humanity (CIPECH14) 28 & 29 November (2014).
9. Wenjuan Li, Qifei Zhang, Jiyi Wu1, Jing Li, Haili Zhao Trust-based and QoS Demand Clustering Analysis Customizable Cloud Workflow Scheduling Strategies IEEE International Conference on Cluster Computing Workshops, (2012).
10. Mr. Manjunatha S, Mr. Bhanu Prakash, and Mr. Balakrishna H M. A Detailed Survey on various Cloud computing Simulators. International Journal of Engineering Research, vol. 5(4), p.790-791 (2016).
11. Praveen Kumar, Anjandeep Kaur Rai. An Overview and Survey of Various Cloud Simulation Tools. Journal of Global Research in Computer Science, vol. 5(1), (2014).
12. S. Namasudra, P. Roy and B. Balusamy, "Cloud Computing: Fundamentals and Research Issues," 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), Tindivanam, 2017, pp. 7-12.
13. Y. T. H. Hlaing and T. T. Yee, "Static Independent Task Scheduling on Virtualized Servers in Cloud Computing Environment," 2019 International Conference on Advanced Information Technologies (ICAIT), Yangon, Myanmar, 2019, pp. 55-59.
14. K. Haghshenas, S. Taheri, M. Goudarzi and S. Mohammadi, "Infrastructure Aware Heterogeneous-Workloads Scheduling for Data Center Energy Cost Minimization," in IEEE Transactions on Cloud Computing, 2020.
15. Kumaraswamy S and Mydhili K Nair, "Bin packing algorithms for virtual machine placement in cloud computing: a review", 2018, pp. 512-524.
16. C. Miyachi, "What is "Cloud"? It is time to update the NIST definition?" in IEEE Cloud computing, vol. 5, no. 3, pp. 6-11, May. /Jun. 2018.
17. Y. Wang, J. Zhou, J. Liu, G. Xu and X. Song, "Research on QoS Optimization Method of Cloud Service Based on Utility Game between Users and Service Providers in the Cloud Market," 2017 IEEE International Conference on Services Computing (SCC), Honolulu, HI, 2017, pp. 297-304.

AUTHORS PROFILE

Mr. Krunal Vaghela received a B.E. degree in Computer Engineering from Saurashtra University, Rajkot, in 2004 and a Master's Degree from NITTTR Chandigarh in 2014. He is a research scholar at the Faculty of Technology, RK University, Rajkot, India. After completion of B.E., he worked for many companies as Project Engineer. Since 2009, he is working in Academics and currently, he is Assistant Professor in the Department of Computer Engineering, Faculty of Engineering at Marwadi Education Foundation Group of Institutions, Rajkot, India. His areas of interest are Grid Computing, Cloud Computing, Computer Networks, Information Security and Mobile Computing.

Dr. Paresh Tanna received the M.C.A Degree from IGNOU University with First Class in 2006 and a PhD degree in computer science from RK University in 2015. He is currently an Associate Professor in the School of Engineering, RK University, Rajkot. His research interests include Data Mining Algorithms, Big Data Analytics, Data Structure Algorithms, Business Intelligence etc.

Dr. Amit Lathigara is working as Associate Professor at Marwadi University, Rajkot, India and has extensive teaching experience of more than 15 years. He has completed his master's from Anna University, Coimbatore and Ph.D. from RK University. He has written more than 20 research papers published in reputed journals and conference proceedings. His preliminary research area focuses on routing in Mobile ad-hoc network and resource and job scheduling under the Cloud environment.