IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

An Algorithm for Load Balancing in Cloud Computing

Afsana[1], Suman Deswal[2*]

[1,2]CSE Dept., Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonipat-131039

[*]*Corresponding Author: suman_gulia2000@yahoo.co.in,*

Available online at: www.ijcseonline.org

Accepted: 16/Aug/2018, Published: 31/Aug/2018

***Abstract*—** In today's scenario, cloud computing is considered as the most extensively spreading platform to execute tasks. These tasks are executed with the help of virtual machines as processing elements. The scheduling of tasks in a cloud computing environment is an important issue as several tasks run on cloud and users sends continuous request to the cloud. The multiple jobs running in parallel slow down the cloud system. The choice of proper scheduling algorithm decreases the cost of executing independent application on cloud resources and improves the performance. Several scheduling techniques are proposed to maintain performance of cloud environment. This paper presents a new algorithm called Suffrage algorithm for scheduling tasks in a cloud computing environment. The proposed algorithm is compared with the existing FCFS and Min-Min algorithms. The comparative analysis of FCFS, Min-Min and Suffrage shows that the Suffrage algorithm performs better than the existing algorithms in terms of makespan time, deadline and finishing time.

***Keywords*—** Cloud Computing, Load Balancing, FCFS, Min-Min.

I. INTRODUCTION

Cloud computing is the mechanism which provides many types of services to users such as servers, storage, database, software, networking & analytics on the network. The companies who offer such computing services are known as cloud providers. These companies charge for cloud computing services used by users. Load balancing is an integral part of cloud computing. A vital role is played by load balancing in cloud computing in the usage of online services as sending email, editing documents, watching movies, and listening music, playing games & storing pictures. Cloud might be internet or network. They have been utilized in wide area network as well as in local area network. It may be used in virtual private network too. Cloud computing [1] has offered platform independency as there is no compulsion to set software on computer. Figure 1 presents the need of cloud computing in today's scenario. The present business applications have become mobile & collaborative due to cloud computing. It enables IT teams to fast set resources to fulfill changeable non-predictable claim related to business. Cloud computing is Offering online development tools too [2]. Several tasks are running on cloud and users sends continuous request to the cloud. As there are numerous tasks to be executed, therefore there is a requirement of scheduling algorithm in cloud computing in order to improve the performance [3].

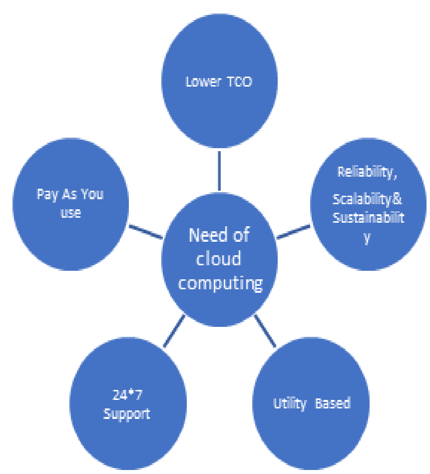

Figure 1 Need of Cloud Computing

Several scheduling techniques have been proposed in order to maintain the performance of cloud environment. These techniques are FCFS [4], Round Robin, Min-Min [5], Min-Max, Genetic algorithm [6]. The paper is organized as follows: The work related to scheduling in cloud computing is presented in section 2, section 3 presents the proposed Suffrage algorithm for scheduling in cloud computing environment, the results and comparative analysis of the existing and proposed algorithms is presented in section 4 and finally section 5 presents the conclusion.

II. RELATED WORK

The author in [7] have made the comparative analysis of task scheduling approaches for cloud environment. They have considered first come first serve and load balancing mechanism for resource management. The author in [8] discussed improved Max-Min algorithm in cloud computing. The influence of RASA (Resource Aware Scheduling Algorithm) algorithm in scheduling tasks has been made in this research. Research depends on expected execution time. But it has considered only Min-Max algorithm and ignored another efficient algorithm. The author in [9] performed the analysis of job scheduling algorithms in cloud computing. It discussed the time complexity, resource allocation, waiting time in case of FCFS, SJF, priority, Round Robin algorithm, and genetic algorithm. But this research ignored the concept of makespan time along with task meet at deadline time. A survey of different scheduling algorithm in cloud computing environments is performed in [10]. It makes comparison of scheduling method, scheduling parameter, scheduling factor and environment in case of RASA (Resource Aware Scheduling Algorithm), RSDC (Reliable Scheduling Distributed in Cloud Computing), Priority based service scheduling, priority-based job scheduling, extended Max-Min, improved cost-based algorithm and gang scheduling algorithm in cloud computing. This research has not provided any practical solutions for problems related to scheduling algorithms. The author in [11] discussed application of Min-Min and Max-Min algorithm for task scheduling in cloud environment in time shared and space shared VM model. Max-min algorithm is found more efficient as compared to scenarios in space shared mode. Min-Min algorithm is providing minimum delays in case of tasks processing. Max-Min is providing better utilization of virtual machine and load balancing. But this paper has not considered the concept of task meet to deadline time. The author in [12] discussed resource allocation with improved Min-Min algorithm. This system has proposed efficient rescheduling dependent task scheduling algorithm known as improved Min-Min algorithm. It is performing scheduling to improve performance of system in distributing system. This research did not consider scheduling mechanism for the cloud computing environment. The author in [13] discussed Min-Max algorithm to solve the resource allocation problem. The proposed work has applied a surrogate relaxation mechanism for getting lower as well as upper bounds. It does not consider the concept of deadline and makespan time. P Yadav, et al [14] worked on blocks. These blocks are sorted and duplicated blocks are identified using Phase Correlation as similarity criterion, but this technique is not able to identify forgery in multiple smooth area. M Jaberi, et al [15] worked on SIFT key points, improving the detection and localization of duplicated regions using more powerful key point-based features, but this technique is failed when the similar methods using key point-based features for matching.

III. METHODOLOGY

III.I SCHEDULING ALGORITHM IN CLOUD COMPUTING

Several scheduling algorithms have been discussed in literature [16]. This section discusses the base scheduling algorithms FCFS and Min-Min for cloud computing. Further a new scheduling algorithm, Suffrage algorithm is proposed.

i. FCFS & Min-Min Scheduling algorithms

This section explains the basic algorithms FCFS & Min-Min.

ii. FCFS Algorithm

First in, first out is known as first come, first served [17]. It is simplest scheduling algorithm. FIFO simply queues processes in order that they arrive in ready queue. In this, the process that comes first will be executed first and next process starts only after the previous gets fully executed.

iii. Min-Min Algorithm

The basic Min-Min procedure in cloud environment [18] chooses task with least size and chooses a cloud resource that has the minimum capacity. After allocation of a task to a virtual machine that task is removed from the queue. There is loop for all tasks ti in MT. where ti represents the task, MT represents minimum time and there is inner loop for all machines mj. The completion time $CT_{ij} = ET_{ij} + R_j$ is calculated, where CT_{ij} represents completion time, ET_{ij} represents execution time and R_j represents ready time of task ti on machine mj. It is be repeated until all tasks in MT are mapped. The calculation is done for each task ti in MT. The task ti with the minimum CT_{ij} is identified and is allocated the resource. CT_{ij} is modified for i.

III.II Suffrage algorithm

The FCFS & Min-Min algorithm suffer from some limitations as given below:

i. In case of high traffic website, the effective use of cloud load balancing is not discussed [19].

ii. The common challenges of finishing time calculation and calculation of task meeting the deadline is not performed.

iii. Load balancing could provide protection against system failures but FCFS, Min-Min scheduling mechanisms failed in load balancing [20].

iv. Some important constraints are not considered like cost, resource utilization.

The Suffrage algorithm considers minimum completion time of each job. First minimum completion time and second minimum completion time of job is calculated. The difference of both times is taken. This difference is the suffrage value. The jobs having maximum difference are set on priority. They are scheduled first for processing and the status of machine is updated after processing of jobs. Figure 2 represents the Suffrage Algorithm.

Algorithm

BEGIN

1. While (J! = Null)
2. For each job $j_i \in J$

    For each machine m_j

    Calculate the completion time

    $C_{ij} = E_{ij} + R_j$ // C_{ij}, E_{ij} and R_j represents completion time, execution time and ready time of job j_i on machine m_j
3. For each job $j_i \in J$

    (a) Find the First minimum completion time (FST_MCT_i) and second minimum completion time (SEC_MCT_i) of job j_i.

    (b) Calculate the suffrage value: $SV_i = SEC\_MCT_i - FST\_MCT_i$

    End For
4. Search the job j_u having maximum suffrage value among all unassigned jobs.
5. Allocate j_u to machine m_v that has resulted in obtaining minimum completion time of j_u.
6. Delete job j_u from the job set J: $J = J - j_u$
7. Update the ready time of machine m_v as: $R_v = Cuv$

    End While

END

Figure 2 Suffrage Algorithm

Figure 3 represents the flow chart of Suffrage algorithm. It provides opportunity for resource selection. All jobs are submitted to the machine which executes the jobs one by one. There is loop of jobs to find both minimum completion times. If there is any job in list then loop continues working otherwise it stops. Calculation of the completion time of jobs is made by following equation $C_{ij} = E_{ij} + R_j$ were C_{ij} represents completion time, E_{ij} represents execution time, R_j represents ready time of job j_i on machine m_j. The difference of first minimum completion time and second minimum completion time is the suffrage value: SVi = SEC_MCTi - FST_MCTi. Jobs that are having maximum suffrage value among all unassigned jobs are scheduled first. The job j_u is allocated to the machine m_v that has resulted in obtaining minimum completion time of j_u. After that, job j_u is deleted from the job set $J= J-J_u$. At the end the ready time of machine is updated by comparing both minimum completion times. The ready time of machine mv is set to R_v = Cuv. R_v

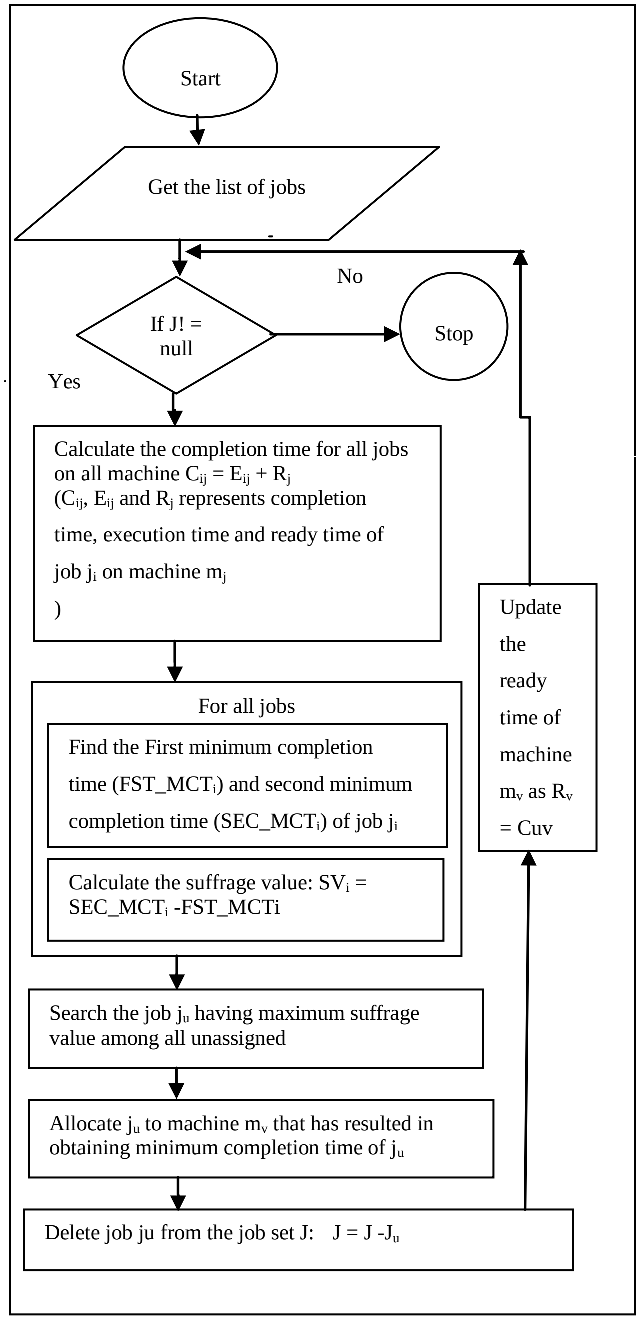

Figure 3 Flow Chart of Suffrage Algorithm

R is the ready time for machine after removal of j_u job, u and v are counter with respect to Machine and Resources.

IV. Results and Discussion

This section presents the comparison of FCFS Min-Min and Suffrage algorithms. The algorithms are simulated on cloudSim. The virtual machines are created and cloudlets are allotted to them for execution [21].

The finishing time of the job has been noted. The comparison charts are created in order to compare finish time, task meet to deadline and makespan time in case of three algorithms.

IV.I Makespan Time

Makespan is the duration from start of work to end. The makespan time of FCFS, Min-Min and Suffrage is considered. The makespan time in case of FCFS, Min-Min and Suffrage is computed to make comparative analysis. The readings for makespan time are considered as per number of tasks i.e. 5, 15, 30, 45, 60, 75, 90, 100 tasks. Figure 4 is showing makespan time in case of FCFS, Min-Min and Suffrage algorithm. This time is calculated in milliseconds.

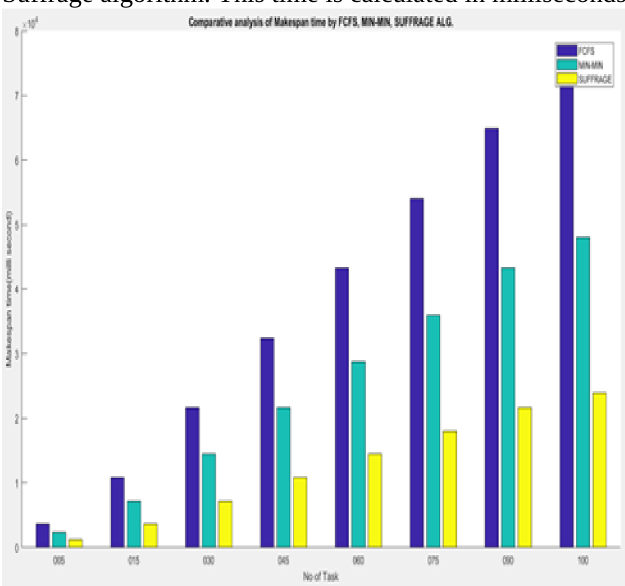

Figure 4 Comparative Analysis of Makespan Time

Results represent that the performance of Suffrage is better as compared to FCFS and Min-Min. It has scheduled the tasks in the organized manner over the cloud system as compared to Min-Min and FCFS. Suffrage is 66.66% better than FCFS and 50% better than Min-Min in case of makespan time, when average calculated.

IV.II Finishing Time

The finishing time is the job completion time, after waiting and processing it.

Finishing time = (Total processing time + total waiting time)/Number of jobs

Simulation has been made by performing comparative analysis of finishing time in FCFS, Min-Min and Suffrage algorithms. Finishing time reading have been considered according to increasing number of jobs i.e. 5, 15, 30, 45, 60, 75, 90, 100 jobs. Finishing time of FCFS, Min-Min and Suffrage are computed in seconds.

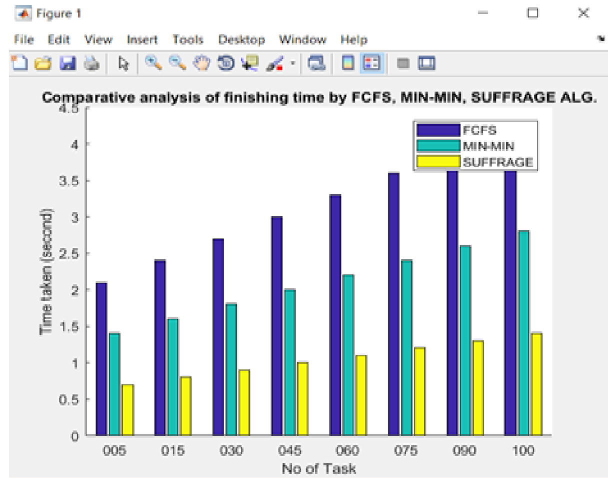

Figure 5 Comparative Analysis of Finishing Time.

Figure 5 presents the finishing time of FCFS, Min-Min and Suffrage algorithm. Suffrage performs better than FCFS and Min-Min algorithms. When average is calculated, Suffrage is found 66.66% better as compared to FCFS and 50% better as compared to Min-Min.

IV.III Deadline Time

Deadline time is the time within which should be completed in a dynamic priority scheduling algorithm. Comparison of deadline time in case of FCFS, Min-Min and suffrage has been calculated then the comparative analysis of tasks meeting the deadline according to schedule as per number of intervals has been made. The number of tasks meeting the deadline time is found according to number of intervals 1, 2, 3, 4, 5, 6, 7, 8. The intervals time taken by FCFS, Min-Min, and Suffrage has been considered in seconds. Figure 6 is representing deadline time in case of FCFS, Min-Min and Suffrage algorithm. The performance of Suffrage is better as compare to FCFS and Min-Min. Suffrage is 62.39% better as compare to FCFS and Proposed work performs 50.42% better as compare to Min-Min in case of deadline time.

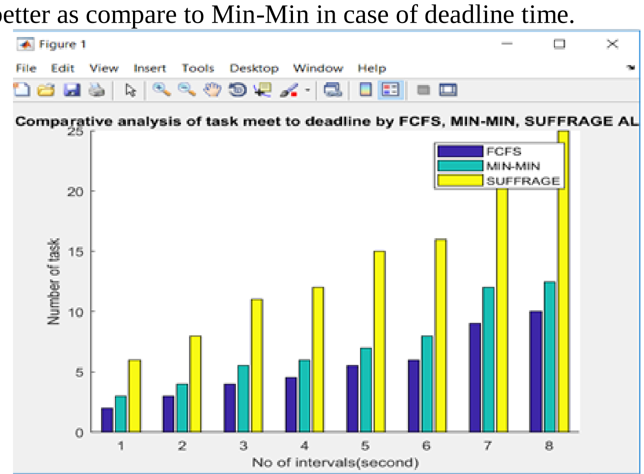

Figure 6 Comparative Analysis of Task Meet to Deadline

V. CONCLUSION AND FUTURE SCOPE

The common challenges in case of scheduling were analysis of finishing time, calculation of task meet to deadline. These scheduling parameters are proven important for load balancing. This research focusses on makespan time issues with traditional FCFS and Min-Min algorithm. The simulation for comparative analysis of finishing time, task meet to deadline, makespan time in case of FCFS, Min-Min and Suffrage algorithm has been done using simulation tool CloudSim. The simulation concludes that the performance of Suffrage algorithm is better than that of Min-Min and FCFS algorithm. It scheduled jobs in the better way on cloud environment as compared to Min-Min and FCFS. Suffrage is 66.66% better than FCFS and 50% better than Min-Min in case of makespan time. Suffrage is 66.66% better than FCFS and 50% better than Min-Min in case of finishing time. Suffrage is 62.39% better than FCFS and 50.42% better than Min-Min in case of deadline time.

Future Scope

There would be always need of efficient scheduling in cloud computing in order to execute parallel tasks. Tasks would be properly scheduled and executed on virtual machines as processing elements. Therefore, implementing high performance computing with efficient scheduling with help of cloud computing would be considered a powerful approach. Enhanced task scheduling algorithm on the cloud computing environment would reduce the makespan time, as well as, decrease the price of executing the independent tasks on the cloud resources.

REFERENCES

[1] Mohsin Nazir, “Cloud Computing: Overview & Current Research Challenges” ISOR journal of computer engineering, ISSN: 2278-0661, ISBN: 2278-8727, Vol.8, Issue1, Nov.-Dec 2012, PP 14-22

[2] K. Vijesh, P. Santhadevi, “Cloud Computing: A Beginners Primer”, International Journal for Research in Applied Science and Engineering Technology (IJRASET), Volume 2 Issue vii, July 2014, Page No: 42-52

[3] Jing Wang, Gongqing Wu, Bin Zhang and Xuegang Hu, “A heuristic algorithm for scheduling on grid computing environment”, In Seventh China Grid Annual Conference, IEEE,2012,13.

[4] Yoshiaki Inoue, Hiroyuki Masuyama, Tetsuya Takine, Toshiyuki Tanaka, “The stationary distribution of the age of information in FCFS single-server queues” 2017 IEEE International Symposium on Information Theory (ISIT)

[5] Kobra Etminani, “A Min-Min Max-Min Selective Algorithm for Grid Task Scheduling” ieee 1-4442-1007-x/07/2007(2007)

[6] Shyam Singh Rajput, Virendra Singh Kushwah, “A Genetic Based Improved Load Balanced Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing” 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)

[7] A Jain, R. Kumar, “A comparative analysis of task scheduling approaches for cloud environment”, In Computing for Sustainable Global Development (INDIA Com), 2016 3rd International Conference on 2016 Mar 16 (pp. 1787-1792). IEEE

[8] O.M Elzeki, M. Z Reshad, M.A Elsoud, “Improved Max-Min Algorithm in Cloud Computing”, International Journal of Computer Applications 50(12):22-27, July 2012.

[9] Rajveer Kaur, Supriya Kinger, “Analysis of Job Scheduling Algorithms in Cloud Computing”, International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 7 – Mar 2014

[10] Pinal Salot, “A survey of various scheduling algorithm in cloud computing environment,” IJRET: International Journal of Research in Engineering and Technology ISSN: 2319-1163

[11] Sanjay Kumar, Atul Mishra, “Application of Min-Min and Max-Min Algorithm for Task Scheduling in Cloud Environment Under Time Shared and Space Shared VM Model”, International Journal of Computing Academic Research (IJCAR) ISSN 2305-9184, Volume 4, Number 6 (December 2015), pp.182-190

[12] Rajwinder Kaur, “Resource Allocation with improved Min-Min Algorithm”, International Journal of Computer Applications (0975 – 8887) Volume 76– No.15, August 2013

[13] Selcuk Karabati, Panagiotis Kouvelis, “A Min-Max-Sum Resource Allocation Problem and Its Applications”, The Institute for Operations Research and the Management Sciences, 1 Dec 2001, Vol. 49, No. 6

[14] Yadav, Preeti, Yogesh Rathore, "Detection of copy-move forgery of images using discrete wavelet transform." International Journal on Computer Science and Engineering4.4 (2012): 565.

[15] Jaberi, Maryam, et al., "Improving the detection and localization of duplicated regions in copy-move image forgery." Digital Signal Processing (DSP), 2013 18th International Conference on. IEEE, 2013.

[16] Shah Mihir , Asst. Prof. Yask Patel, “A Survey of Task Scheduling Algorithm in Cloud Computing”, In International Journal of Application or Innovation in Engineering & Management (IJAIEM) January 2015.

[17] Manish Kumar Mishra, “An Improved FCFS (IFCFS) Disk Scheduling Algorithm” International Journal of Computer Applications (0975 – 888) Volume 47– No.13, June 2012

[18] Davinder Kaur, “An Efficient Job Scheduling Algorithm using Min-Min and Ant Colony Concept for Grid Computing”, International Journal of Engineering and Computer Science ISSN:2319-7242 Volume 03 Issue 07 July, 2014 Page No. 6943-6949

[19] Khiyaita A, El Bakkali H, Zbakh M, El Kettanid, “Load balancing in cloud computing: state of art”, In Network Security and Systems (JNS2), 2012 National Days of 2012 Apr 20 (pp. 106-109). IEEE.

[20] Tarun Kumar Ghosh, Rajmohan Goswami, Sumit Bera, Subhabrata Barman, “Load Balanced Static Grid Scheduling Using Max-Min Heuristic”, In 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012.

[21] Kathiravelu, Pradeeban, Veiga, Luís,“Concurrent and Distributed CloudSim Simulations (PDF)” IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems Paris. pp. 490–493. Archived from the original (pdf) on 9 September 2014. Retrieved 2 January 2016.

Authors Profile

Suman Deswal, is pursuing her Ph.D from DCR University of Science & Technology, Murthal, India. She has completed her M.Tech (Computer Science & Engg.) from Kurukshetra University, Kurukshetra, India and B.Tech (Computer Science & Engg.) from CR State College of Engg., Murthal, India in 2009 and 1998 respectively. She possesses 16 years of teaching experience and is presently working as Assistant Professor in the department of Computer Science and Engg. at DCR University of Science and Technology, Murthal, India. She has published more than 30 papers in various international journals.teaching experience and 4 years of Research Experience.

Afsana , is pursuing her M.Tech (Computer Science & Engg) from DCR University of Science & Technology, Murthal, India. She has completed her B.Tech (Information Technology) from BPSMV University (Bhagat Phool Singh Mahila Vishwavidyalaya) Khanpur Kalan, Sonepat, India in 2016.

.

