iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: I Month of publication: January 2018

DOI: http://doi.org/10.22214/ijraset.2018.1393

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Cloud System Task Distribution with VM Status and Configuration Management using Fuzzy Logic

R.K. Shrivastava[1], Vikash Goswami[2]
[1,2] S.M.S Govt. Science Collage, Jiwaji University, S.M.S Govt. Science Collage, Jiwaji University

Abstract*: Task distribution and server management are the key requirements for proper operation of any serving system. The proper distribution of task over a number of VMs, configuring the VM depending upon the requirements and maintaining the states of VM in a cloud environment can greatly improve the QoS and power requirements of the cloud. This paper presents a fuzzy logic based system for above mentioned task, where a number of fuzzy decision making systems are used in combination to select the best VM or creates the new VM for the given task on the basis of current operating conditions (like load, available resources, task requirements etc.), and to maintain the states (dissolve, sleep, keep alive) of VM to save the power and reclaim the resources. Besides the providing better QoS and power saving, the proposed technique also reduces the active physical resources requirements. The effectiveness of the proposed technique it is validated by comparing it against standard algorithms and the simulation results verify that the technique outperforms standard techniques in terms of QoS, resource management, and power savings

***Keywords:* Cloud Computing, Cloud VM Management, Fuzzy Logic Controller**

I. INTRODUCTION

The Internet services or applications implemented in cloud, runs over the Virtual Machines (VMs) formed inside the cloud system. The VMs are formed by virtually (through software configurations instead of direct physical connections) allocating and configuring the physical infrastructure is managed by a virtualization platform. The cloud service provider has to maintain the minimum level Quality-of-Service (QoS), which also known as Service Level Objective (SLO).One possible approach for achieving the SLO is to increase the infrastructure, but this require huge investment and will also increase the maintenance and running cost of the cloud. This increase in cost may not be acceptable particularly in the present competitive business environment. Hence the only way to achieve the SLO without additional investment is to efficiently utilize the available resources.

In order to efficiently utilize the available resources the cloud manager requires adaptive management of VM configuration, VM status and task distribution, depending upon the current workload demands, power saving policies, task queue length etc. In this paper, A Fuzzy logic based system which can manage the balance among above stated things by dynamically configuring them. The organization of rest of the paper is as follows. The Section II provides a brief literature review. The Section III gives an overview of fuzzy logic controller. The Section IV explains the system architecture with simulation configurations, The Section Vex plains the proposed system. The simulation results are presented in Section VI and finally, the Section VII, presents the conclusion and discusses the possibilities of future works

II. LITERATURE REVIEW

In a cloud environment the task distribution is a process used to assigns the execution of tasks on distributed resources [6]. The most common way of doing that is using the optimization techniques which finds the optimal task to VM pair for given objective function and constrains. A number of techniques with different optimization techniques, objective functions and constrains have been proposed in [3, 4, 5, 8, 15, 16, and 17]. However all the techniques having optimization algorithm in common a number of difference in their utilization can be seen. For instance in [3] the Honey-Bee behaviour (HBB) based optimization technique is used to achieve balancing of tasks over available VMs, the proposed algorithm manages the task execution priorities. The [5] and [15] both uses the particle swarm optimization (PSO), although the [5] adopted for deadline constrained task scheduling using self-adaptive learning, while [15] constrained task execution time and data transfer cost. In [8] a multi-objective genetic algorithm used. The multi-objective optimization provides extra facilities to achieve more than one objective simultaneously, hence it is able to achieve four different objectives, namely minimizing task transfer time, task execution cost, power consumption, and task queue length. In [16]the ant colony optimization is used to balance the entire system load at minimum make span of a given tasks set. The improved differential evolution algorithm (IDEA) for the purpose in presented in [17], which combines the Taguchi method and a differential evolution algorithm(DEA).Their multi-objective optimization approach uses the processing and receiving cost as the first objective and receiving, processing, and waiting time as second objective. Another approach for the task scheduling is using the fuzzy logic. The fuzzy logic is a method for solving complex decision making problem where the variables show some degree of overlapping. The fuzzy logic system has been successfully implemented for many control and decision making systems some of its application can be seen on washing machines, refrigerators and other household goods. In [12] the fuzzy logic is used for prediction of the virtual machine to assign the upcoming job, considering that the requirements of memory, bandwidth and disk space are imprecise. In [9] the task scheduling model for virtual data centers with uncertain workload and uncertain nodes availability is presented. The presented solution deals the problem as a two-objective optimization as a trade-off between availability and the average response time of VDC (virtual data center). Since the optimization requires the availability of VDC and workload values in advance the type-I and type-II fuzzy based predictor for VDC availability and Load-Balance is proposed. The hybrid job scheduling algorithm which involves genetic algorithm and fuzzy logic is presented in [1, 14]. The fuzzy logic is used here to reduce the number of iterations required by genetic algorithm to converse.

III. FUZZY LOGIC SYSTEM

Fuzzy logic is an approach where a variable simultaneously belongs to more than one class with certain degree and the degree of membership is defined by membership function. The fuzzy logic approximate decision making using natural language terms. It is especially useful in modeling of systems where information cannot be defined precisely, but some broad definitions can be formed. Because of its simplicity and effectiveness, Fuzzy-logic technology has gained many applications in scientific and industrial applications.

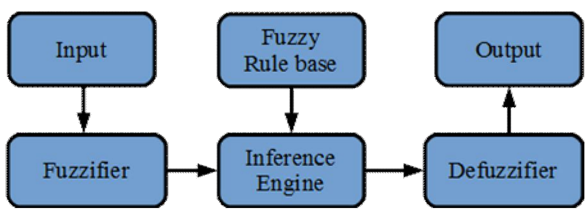


Figure 1: Typical architecture of Fuzzy Logic Controller (FLC) system.

A typical architecture of FLC is shown in fig. 1, which comprises of four principal comprises: a Fuzzifier, a Fuzzy rule base, inference engine, and a de-Fuzzifier. Fuzzification: The Fuzzification is the process of converting crisp values in terms of degree of membership with different classes. The degree of membership is calculating using membership functions. The Fuzzification enables variable association with linguistic term.

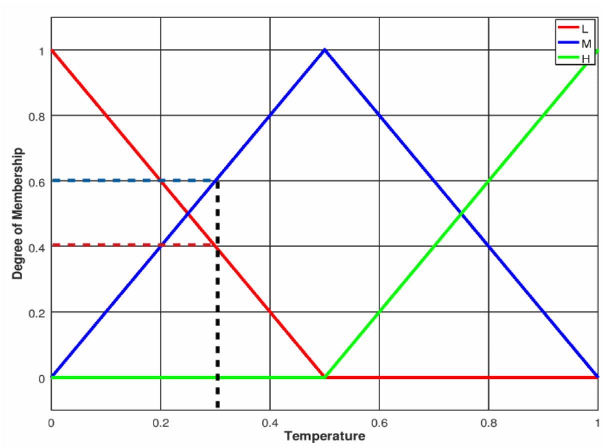


Figure 2: showing Fuzzification process with triangular membership function

The fig. 2, shows the mapping the crisp value of temperature with L, M and H categories, the figure shows that for 0.3 task priority (black dashed line) it has the membership of 0.4 (red dashed line), 0.6 (blue dashed line) and 0 for categories L, M and H respectively. In fig. 2 the membership function is of triangular however it can be, trapezoidal, Gaussian, bell-shaped, sigmoidal etc. The exact type depends on the actual application.

Fuzzy Rule Base: this contains the rule which relates the fuzzy inputs and outputs in linguistic terms. It contains a sequence of the form IF-THEN. For example in a heating system it can be defined as
IF temperature is L THEN set heater power to H
IF temperature is M THEN set heater power to M
IF temperature is H THEN set heater power to L
Defining these rules are requires expertise in related field of application.
Inference Engine: It executes all the fuzzy rules available in the fuzzy rule base for the available inputs. This produces a number of fuzzy outputs one for each rule.

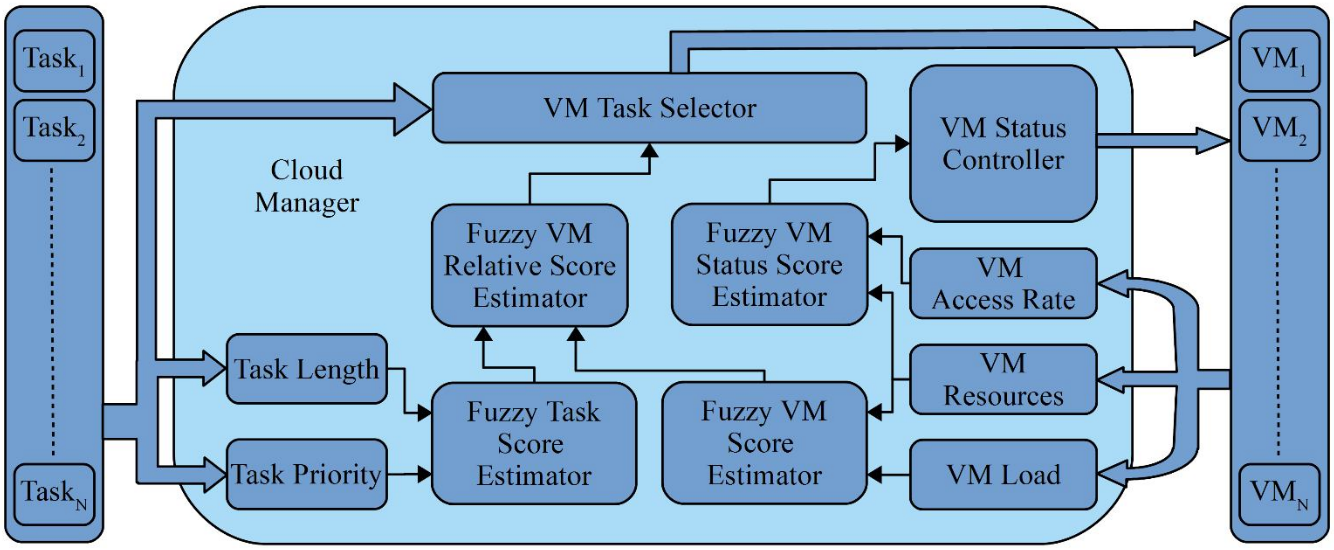


Figure 3: The proposed system architecture showing the different functional blocks and their interconnections.

De-Fuzzification: The output from inference engine is still fuzzy which must be converted into crisp value before it can be used with any non-fuzzy system. This conversion of fuzzy outputs to crisp value is done by De-Fuzzification. The De-Fuzzification can be performed by many ways such as using Centroid, Max-Membership, Weighted Average and Mean-Max Membership etc.

IV. PROPOSED ALGORITHM

A. *System Architecture*

The architecture of the proposed system is presented in fig.3 the system contains four fuzzy logic decision making blocks, two VM controlling blocks and five information extraction blocks. The working details of each block is as follows:

1) Information extraction blocks: these blocks are used to extract useful information from task queue and VMs.
2) Task Length: the length of current task in MIPS
3) Task Priority: Execution priority of current task
4) VM Access Rate: how many times the particular VM has accessed during predefined time interval
5) VM Resources: the resources used by VM
6) VM Load: current load on VM
7) Fuzzy Logic Decision Making Blocks: these blocks are used to make specific decisions based on provided inputs using fuzzy logic
8) Fuzzy Task Score Estimator: this block estimates the task requirements on the basis of task length and priority
9) Fuzzy
10) VM Score Estimator: this block estimates the VM capability to handle tasks on the basic of VM configuration and loadFuzzy VM Relative Score Estimator: this block estimates fitness between VM capability and task requirements.

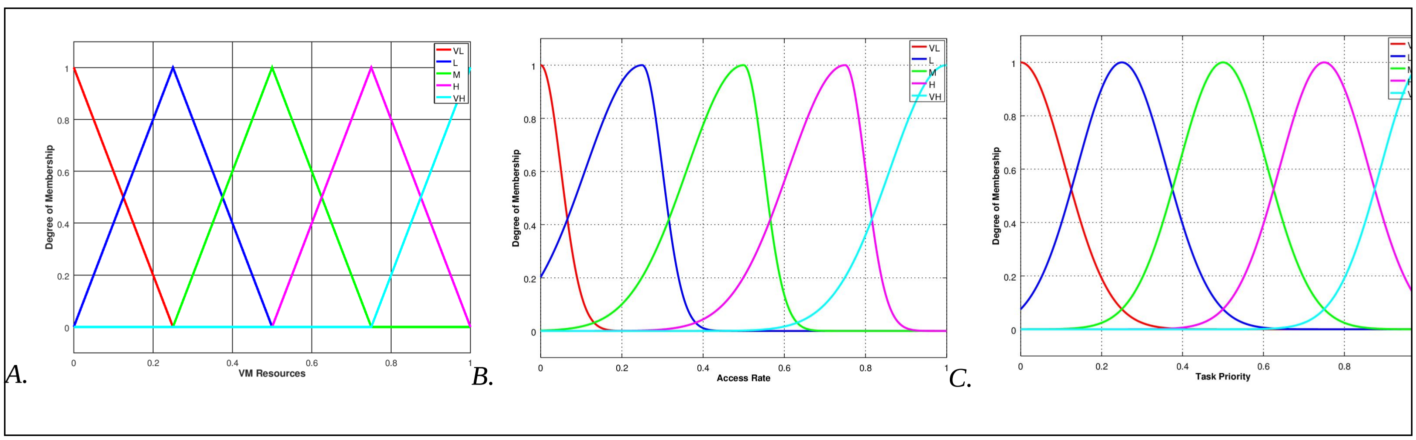
VM Status Score Estimator: this block is used to decide the operational status of VM on the basis of resource utilized by VM and access rate of VM.

VM Task Selector: this block is used to select the best VM for the current task, it takes the input from Fuzzy Relative Score Estimator for all the VMs and selects the VM having the highest relative score.

VM status Controller: it maintains the states of all VM on the basis of VM status score (VM_{score}). This blocks uses two different thresholdst_{sleep} and $t_{dissolve}$ where $0 < t_{sleep} < t_{dissolve} < 1$ which are compared against VM status score to decide the VM status as follows:

$$VM_{status} = \begin{cases} Keep\ Running, & if\ VM_{score} < t_{sleep} \\ Shutdown, & elseif\ t_{sleep} \geq VM_{score} > t_{dissolve} \\ Dissolve, & else\ VM_{score} \geq t_{dissolve} \end{cases}$$

Fuzzy Membership Function Selection: the selection of member function in most difficult and important task for any fuzzy system. Because these membership function defines the fuzziness and the way variable changes their memberships to different classes, the improper selection of these function can drastically degrade the performance of fuzzy system. In the proposed work the triangular membership-ship function (as shown in fig. 4(a)) is used for the variables Task Length, VM Load and VM resources, the triangular membership function is selected for these variable because these parameters are consider to have the linear transition in degree of membership from one to another class. The two sided Gaussian membership function (as shown in fig. 4(b)) is used for the Access Rate, because we want to increase its importance much quickly then linear rate. The two sided Gaussian membership function is designed as its transition from present class to higher class is much faster and it also leaves it present class much sharply. The Gaussian membership function (as shown in fig. 4(c)) is used for variable Task Priority, as the priority is considered to have a continuous symmetric transition.



D. (a) E. (b) F. (c)

Figure 4: the membership functions used with fuzzy estimators, (a) triangular, (b) two sided Gaussian, (c) Gaussian.

B. *Model Terminology*

Before going to description this section explains the terminology used in the algorithm.

Task Length(TL_i) = the length of the i^{th} task.

Execution Capacity of VM(C^i_{VM}) = rate of task execution for i^{th} VM.

Current Load on VM(L^i_c) = the remaining task length of currently executing task on i^{th} VM.

Access Rate of VM(R^i) = number of different tasks assigned to i^{th} VM per unit time.

Task Waiting Time(T^i_w) = describes the time only after that the i^{th} VM can start execution of the requested task. It can also be described as the time required to finish the currently executing tasks on the i^{th} VM.

$$T^i_w = L^i_c / C^i_{VM}$$

Task Execution Time($T_{e,j}^{i}$) = the time required by the i^{th} VM to execute the j^{th} task.

$$T_{e,j}^{i} = TL_j / C_{VM}^{i}$$

Total Task Completion Time($T_{t,j}^{i}$) = it is the sum of Task Waiting Time and Task Execution Time.

$$T_{t,j}^{i} = T_w^i + T_{e,j}^{i}$$

Task Priority (TP_i) = is the inverse of required maximum Total Task Completion Time for the i^{th} task.

Fuzzy Task Score Estimator =$F_{TS}(TL_i, TP_i)$.

Fuzzy Task Score(S_{TS}^{i}) = task score of i^{th} task calculated by Fuzzy Task Score Estimator$F_{TS}(\ )$.

Fuzzy VM Score Estimator=$F_{VM}(L_c^i, C_{VM}^i)$.

Fuzzy VM Score(S_{VM}^{i}) = VM score of i^{th} VM calculated by Fuzzy VM Score Estimator$F_{VM}(\ )$.

Fuzzy VM Relative Score Estimator=$F_{VMR}(S_{TS}^i, S_{VM}^i)$.

Fuzzy VM Relative Score (S_j^i) = relative VM score of i^{th} VM for j^{th} task calculated by Fuzzy VM Relative Score Estimator$F_{VMR}(\ )$.

Fuzzy VM Status Score Estimator=$F_{SS}(R_c^i, C_{VM}^i)$.

Fuzzy VM Status Score(S_{SS}^{i}) = VM status score of i^{th} VM calculated by Fuzzy VM Status Score Estimator$F_{VM}(\ )$.

C. Proposed Algorithm

Let the task length and priority of newly arrived task be TL_{new} and TP_{new} respectively.

N = *total numbers of VMs currently running in VM*.

```
Start Main Routine
        for i = 1 to N
T_w^i = TL_new / C_VM^i
T_e,j^i = TL_j / C_VM^i
T_t,j^i = T_w^i + T_e,j^i
    endfor
    VM_available = 0
    for i = 1 to N
            if (T_t,j^i > 1/TP_new)
VM_available = 1
        break;
endif
endfor
if (VM_available == 0)
```

$CreateNewVM(\ );$

$\boldsymbol{else}$

$AssignTask(\ );$

$\boldsymbol{endif}$

$\boldsymbol{End\ Main\ Routine}$

$\boldsymbol{Start\ Sub - Routine\ CreateNewVM}$

$C_{VM}^{new} = TL_{new} \times TP_{new};$

$create\ new\ VM_{new}\ with\ configuration\ C_{VM}^{new};$

$assign\ Task\ to\ VM_{new}.$

$\boldsymbol{End\ Sub - Routine\ CreateNewVM}$

$\boldsymbol{Start\ Sub - Routine\ AssignTask}$

$S_D = 0;$

$VM_{selected} = 0;$

$\boldsymbol{for}\ i = 1\ to\ N$

$S_{VM}^i = F_{VM}(L_c^i, C_{VM}^i);$

$S_{TS}^i = F_{TS}(TL_i, TP_i);$

$S_c^i = F_{VMR}(S_{TS}^i, S_{VM}^i);$

$\boldsymbol{if}(S_c^i > S_D)$

$S_D = S_c^i;$

$VM_{selected} = i;$

$\boldsymbol{endif}$

$\boldsymbol{endfor}$

$assign\ Task\ to\ VM_{selected}.$

$\boldsymbol{End\ Sub - Routine\ AssignTask}$

$\boldsymbol{Start\ Sub - Routine\ VMStatusControl}$

$\boldsymbol{for}\ i = 1\ to\ N$

$\boldsymbol{if}(isIdle(VM_i))$

$S_{SS}^i = F_{SS}(R_c^i, C_{VM}^i);$

$\boldsymbol{if}(S_{SS}^i > TH_{dissolve})$

Dissolve the VM_i;

elseif$(S^i_{ss} > TH_{sleep})$

Set VM to Sleep

else

do nothing

endif

endif

endfor

End Sub – Routine VMStatus Control

Table 1: The Fuzzy Rules and Surf plots for all fuzzy controllers.

D. Proposed Algorithm Explanations

As shown in fig. 4 the cloud manager waits for the arrival of the new task and as soon as it receives the new task from the task queue it extracts the task related parameters like task length and task priority. Once it founds these values it checks all the VMs for condition $T^i_{t_j} > \frac{1}{TP_{new}}$(as shown in algorithm), if it did not find any VM then it move to create a new VM according to the task requirements (as shown in algorithm sub-routine Assign Task). Otherwise if it find then calculate the relative VM score for all such VMs using fuzzy rules defined in table 1(c). to estimate VM relative score it send task length and priority values to fuzzy task score estimator (as shown in fig. 3). This fuzzy estimator calculates the score according to the values and the rules defined in table 1(a). The fuzzy task score works as one input for the fuzzy relative score estimator for the second input the cloud manager scans all the running VMs for their execution capacity and current load. The above two values in then applied to fuzzy VM score

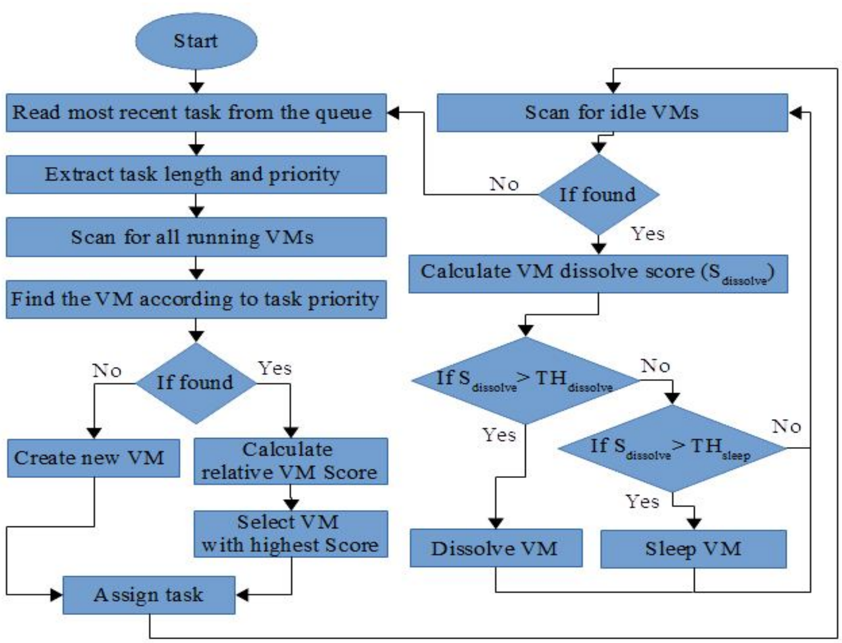


Figure 5: Flow chart of the proposed algorithm.

Estimator block which estimates the VM score according to the rules defined in table 1(b). The VM score works as second input for the VM relative score estimator. The procedure is repeated for all the running VMs and the relative scores are stored. Now the VM with highest relative score is selected for assignment of input task.

To manage the status of VMs the cloud manager scans each VM for their execution capacity and accessing rate then these values are used to estimate their status score by applying fuzzy rules defined in table 1(d). The calculated status score is compared against $TH_{dissolve}$ and if it finds VM score greater then it dissolves the VM and reclaim its resources. Otherwise it check the score against TH_{sleep} to check if it can be set into sleep state or should keep running

V. SIMULATION RESULTS

A. Numerical Model Considerations for Cloud System

Following consideration are taken into account when the numerical models is developed.

1) It is assumed that the load balancer knows the configuration (like processing capacity, memory etc.) of each virtual machine (VM) in the cloud.
2) The load balancer can get the operational state of each VM with zero time delay.
3) The load balancer takes no time in selecting and assigning the tasks to VM's.
4) The load balancer selects the VM for the input tasks on the basis of selected algorithm.
5) Each VM has zero booting time hence start executing assigned task immediately.
6) The incoming tasks size is considered in MI (million instructions) units.
7) The VM's capacities are also considered in MIPS (million instructions per second) units.

The evaluation of the proposed algorithm is performed using OCTAVE/ MATLAB numerical computing software. During the simulation the tasks arrive as a Poisson process at a rate of λ. The random length tasks within the provided minimum and maximum task length limits are generated using a uniform discrete distribution. The similar way is used for the generation of task priorities and defining the VM execution capacities.

B. Definition of Evaluation Terms

The following measures are used to evaluate the performance of the algorithm.

1) *SLA Failure:* is defined as failure of the cloud in serving the task within given time bound (inverse of priority).
2) *SLA Failure Task Length:* defines the length of the *SLAFailure*task.
3) *VM Reboots:* is the booting of VMs from sleep mode, this operation is required when the already running VMs cannot serve the current task.
4) *VM Reforms:* is the formation of new VM form the available unused resources when the current task cannot be handled by the already formed (running or sleeping) VMs.
5) *Resource Utilization Efficiency:* is presents that how efficiently the cloud resources are utilized to serve the tasks, and it is calculated as follows:

$$Resource\ Utilization\ Efficiency = \frac{\sum_{i=1}^{N} TL_i}{\sum_{i=1}^{N}\left(\sum_{j=1}^{A_i} c_{VM}^{j}\right)} \times 100$$

Where TL_i: is the load in cloud at time i.

A_i: is the number of VMs active and running at time i.

c_{VM}^{j}: Execution capacity of the j^{th} VM.

N: is the total simulation time (discrete events of task arrival).

C. Simulation Environment Configurations

To simulate the algorithm properly some important parameters are required to configure these parameters and their values are listed in table 2.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Table 2: The simulation parameters and their values.

G. Configuration Parameter	H. Parameter Value
I. Total Execution Capacity Available	J. 100 MIPS
K. Minimum Task Length	L. 100 MI
M. Minimum Task Execution Time	N. 1 Seconds
O. Maximum Task Execution Time	P. 10 Seconds
Q. Threshold Sleep	R. 0.5
S. Threshold Dissolve	T. 0.7
U. Total Simulation Time	V. 100 Seconds

D. Simulation Outcomes

The outcomes of the simulation is presented in graphical forms. The outcomes of the proposed algorithm is also compared with the two standard task scheduling algorithm names Round Robin and Random Selection

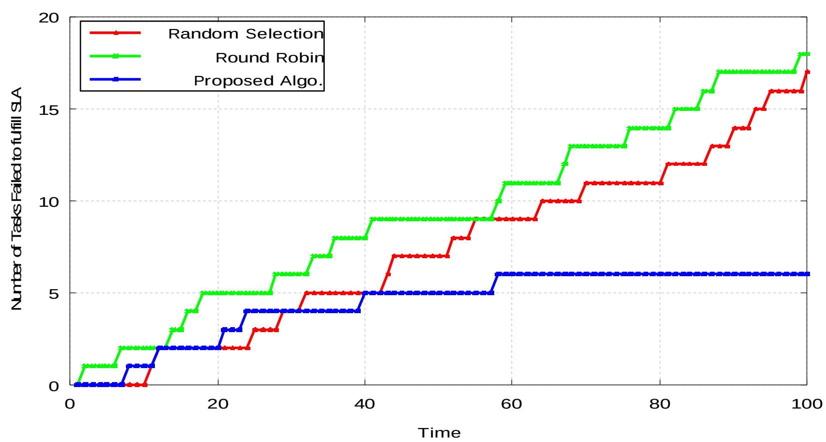


Figure 6: Plot for number of tasks failed to receive the requested SLA by cloud due shortage of resources with respect to simulation time.

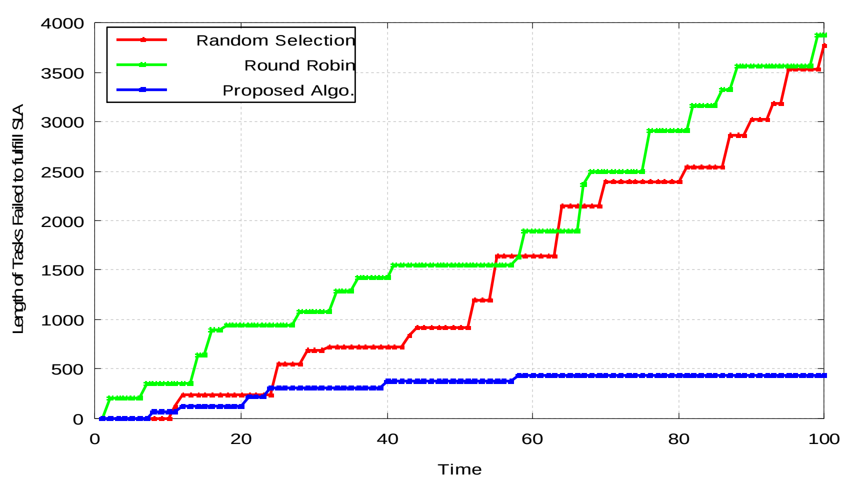


Figure 7: Plot for total length of the tasks which failed to receive the requested SLA by cloud due shortage of resources with respect to simulation time.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

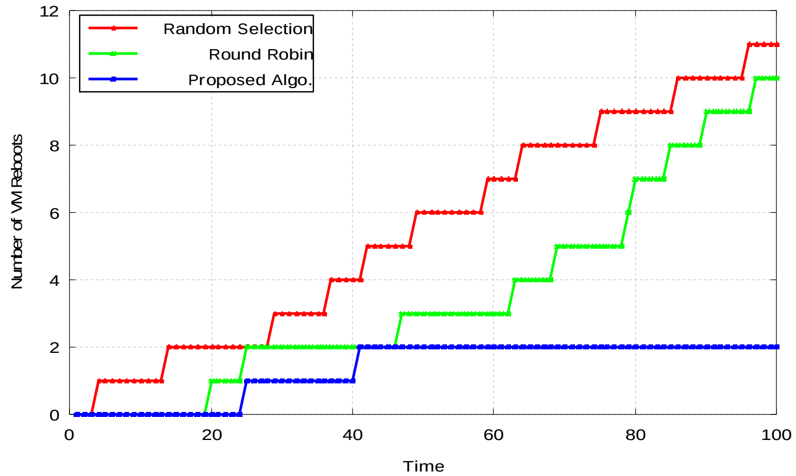


Figure 8: Plot for number of time the VM is rebooted from the sleep mode for assignment of task with respect to simulation time.

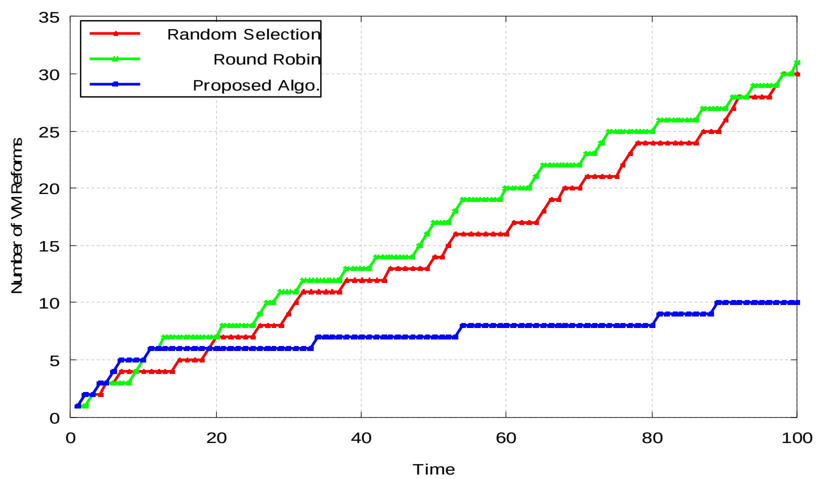


Figure 9: Plot for number of time the VM is reformed from the available resources for assignment of task with respect to simulation time.

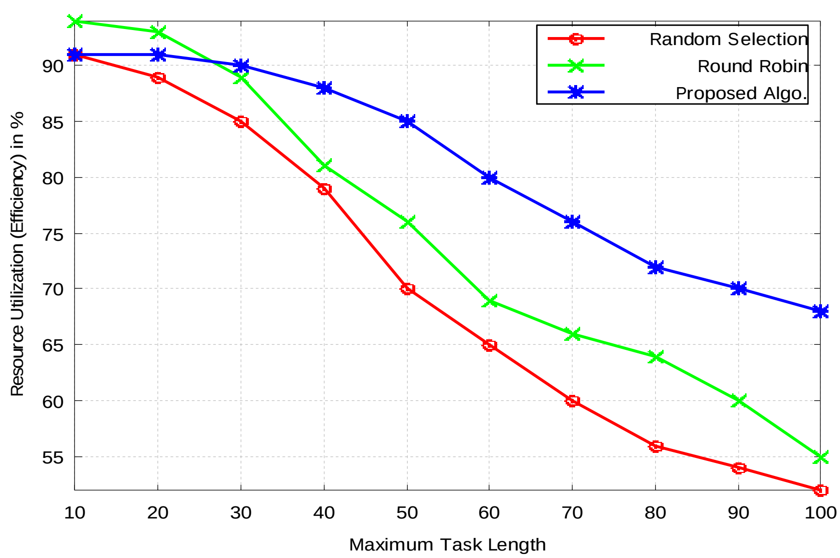


Figure 10: Plot showing the variation of the cloud resource utilization efficiency with respect to maximum task length.

VI. CONCLUSION

In this paper, we presented the fuzzy logic based VM Status and task assignment scheme for Cloud systems. The simulation results shows that the proposed algorithm reduces the number of tasks, cloud failed to deliver the guaranteed SLA by a factor of 2.5 (fig. 6), also the total tasks length failed to get the guaranteed SLA is reduced by a factor of 8.0 (fig. 7) which is very much higher than the 2.5. The huge difference in these factors shows that the algorithm is able to manage the cloud in such a way that it can minimize the loss due to SLA disagreements.

The proposed algorithm manages the number of reboots of VMs which causes delay in the response by a factor of 5.0 (fig. 8) and the number of VM reforms is reduced by the factor of 3.0 (fig. 9). The number of reboots shows the intelligent management of the VMs states so that the most common VM configuration is kept running. While reduction in number of VM reforms shows that the VMs are configured in such a way that most of the time the cloud can get the active VMs for the tasks SLA requirement it also reduce the response time.

At last the efficiency of the algorithms are compared which shows that the proposed algorithm maintains a relative margin of 50% (fig. 10) over others,this margin further increases with the increase in load, this achievement reflects that the proposed algorithm manages the resources of the cloud much efficiently. The overall analysis of these results depicts that the proposed algorithm has the capability of efficiently utilizing the resources in such a way that (1) it can mostly fulfill the task SLA requirements, (2) it reduces the cloud response time and (3) it increases the resource utilization efficiency which reduces the power requirements andresource requirements.

Although the presented algorithm provide better results than the conventional algorithms compared in this paper. The proposed algorithm can be further improved by fine tuning the membership functions and rule base however these modifications are leaved for the future work.

REFERENCES

[1] M. Shojafar, S. Javanmardi, S. Abolfazli, and N. Cordeschi, “Erratum to: FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method,” Cluster Computing, vol. 18, no. 2, pp. 845–845, 2015.

[2] H. Chen, F. Wang, N. Helian, and G. Akanmu, “User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing,” 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), 2013.

[3] D. B. L.d. and P. V. Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” Applied Soft Computing, vol. 13, no. 5, pp. 2292–2303, 2013.

[4] C.-W. Tsai and J. J. P. C. Rodrigues, “Metaheuristic Scheduling for Cloud: A Survey,” IEEE Systems Journal, vol. 8, no. 1, pp. 279–291, 2014.

[5] X. Zuo, G. Zhang, and W. Tan, “Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud,” IEEE Transactions on Automation Science and Engineering, vol. 11, no. 2, pp. 564–573, 2014.

[6] S. Adabi, A. Movaghar, and A. M. Rahmani, “Bi-level fuzzy based advanced reservation of Cloud workflow applications on distributed Grid resources,” The Journal of Supercomputing, vol. 67, no. 1, pp. 175–218, 2013.

[7] D. Poola, S. K. Garg, R. Buyya, Y. Yang, and K. Ramamohanarao, “Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds,” 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, 2014.

[8] F. Ramezani, J. Lu, J. Taheri, and F. K. Hussain, “Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments,” World Wide Web, vol. 18, no. 6, pp. 1737–1757, Oct. 2015.

[9] X. Kong, C. Lin, Y. Jiang, W. Yan, and X. Chu, “Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction,” Journal of Network and Computer Applications, vol. 34, no. 4, pp. 1068–1077, 2011.

[10] W. Lin, C. Liang, J. Z. Wang, and R. Buyya, “Bandwidth-aware divisible task scheduling for cloud computing,” Software: Practice and Experience, vol. 44, no. 2, pp. 163–174, 2012.

[11] S. K. Garg, A. N. Toosi, S. K. Gopalaiyengar, and R. Buyya, “SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter,” Journal of Network and Computer Applications, vol. 45, pp. 108–120, 2014.

[12] M. S. Q. Z. Nine, M. A. K. Azad, S. Abdullah, and R. M. Rahman, “Fuzzy logic based dynamic load balancing in virtualized data centers,” 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2013.

[13] A. N. Toosi and R. Buyya, "A Fuzzy Logic-Based Controller for Cost and Energy Efficient Load Balancing in Geo-distributed Data Centers," 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), Limassol, 2015, pp. 186-194.

[14] Javanmardi S., Shojafar M., Amendola D., Cordeschi N., Liu H., Abraham A. “Hybrid Job Scheduling Algorithm for Cloud Computing Environment,” IBICA 2014. Advances in Intelligent Systems and Computing, vol 303. Springer, Cham.

[15] Pandey, L. Wu, S. M. Guru, and R. Buyya, “A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments,” 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

[16] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, “Cloud Task Scheduling Based on Load Balancing Ant Colony

[17] J.-T. Tsai, J.-C. Fang, and J.-H. Chou, “Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm,” Computers & Operations Research, vol. 40, no. 12, pp. 3045–3055, 2013.

I. PAGE LAYOUT

An easy way to comply with IJRASET paper formatting requirements is to use this document as a template and simply type your text into it.

A. Page Layout

Your paper must use a page size corresponding to A4 which is 210mm (8.27") wide and 297mm (11.69") long. The margins must be set as follows:

1) Top = Bottom= 19mm (0.75")

2) Left = Right = 14.32mm (0.56")

II. PAGE STYLE

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

A. Text Font of Entire Document

The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

Recommended font sizes are shown in Table 1.

B. Title and Author Details

Title must be in 24 pt Regular font. Author name must be in 11 pt Regular font. Author affiliation must be in 10 pt Italic. Email address must be in 9 pt Courier Regular font.

TABLE I

FONT SIZES FOR PAPERS

Font Size	Appearance (in Time New Roman or Times)		
	Regular	Bold	Italic
8	table caption (in Small Caps), figure caption, reference item		reference item (partial)
9	author email address (in Courier), cell in a table	abstract body	abstract heading (also in Bold)
10	level-1 heading (in Small Caps), paragraph		level-2 heading, level-3 heading, author affiliation
11	author name		
24	title		

All title and author details must be in single-column format and must be centered.

Every word in a title must be capitalized except for short minor words such as “a”, “an”, “and”, “as”, “at”, “by”, “for”, “from”, “if”, “in”, “into”, “on”, “or”, “of”, “the”, “to”, “with”.

Author details must not show any professional title (e.g. Managing Director), any academic title (e.g. Dr.) or any membership of any professional organization (e.g. Senior Member IEEE).

To avoid confusion, the family name must be written as the last part of each author name (e.g. John A.K. Smith).

Each affiliation must include, at the very least, the name of the company and the name of the country where the author is based (e.g. Causal Productions Pty Ltd, Australia).

Email address is compulsory for the corresponding author.

C. Section Headings

No more than 3 levels of headings should be used. All headings must be in 10pt font. Every word in a heading must be capitalized except for short minor words as listed in Section III-B.

1) Level-1 Heading: A level-1 heading must be in Small Caps, centered and numbered using uppercase Roman numerals. For example, see heading “III. Page Style” of this document. The two level-1 headings which must not be numbered are “Acknowledgment” and “References”.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

2) *Level-2 Heading:* A level-2 heading must be in Italic, left-justified and numbered using an uppercase alphabetic letter followed by a period. For example, see heading "C. Section Headings" above.
3) *Level-3 Heading:* A level-3 heading must be indented, in Italic and numbered with an Arabic numeral followed by a right parenthesis. The level-3 heading must end with a colon. The body of the level-3 section immediately follows the level-3 heading in the same paragraph. For example, this paragraph begins with a level-3 heading.

D. Figures and Tables

Figures and tables must be centered in the column. Large figures and tables may span across both columns. Any table or figure that takes up more than 1 column width must be positioned either at the top or at the bottom of the page.

Graphics may be full color. All colors will be retained on the CDROM. Graphics must not use stipple fill patterns because they may not be reproduced properly. Please use only *SOLID FILL* colors which contrast well both on screen and on a black-and-white hardcopy, as shown in Fig. 1.

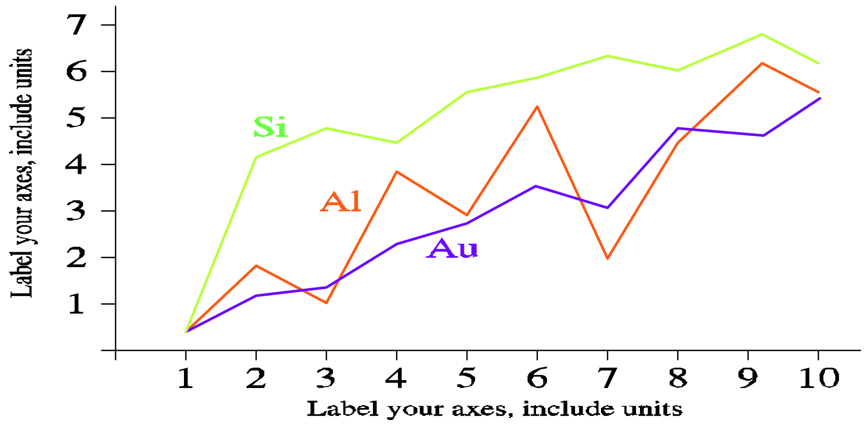


Fig. 1 A sample line graph using colors which contrast well both on screen and on a black-and-white hardcopy

Fig. 2 shows an example of a low-resolution image which would not be acceptable, whereas Fig. 3 shows an example of an image with adequate resolution. Check that the resolution is adequate to reveal the important detail in the figure.

Please check all figures in your paper both on screen and on a black-and-white hardcopy. When you check your paper on a black-and-white hardcopy, please ensure that:

1) the colors used in each figure contrast well,
2) the image used in each figure is clear,
3) All text labels in each figure are legible.

E. Figure Captions

Figures must be numbered using Arabic numerals. Figure captions must be in 8 pt Regular font. Captions of a single line (e.g. Fig. 2) must be centered whereas multi-line captions must be justified (e.g. Fig. 1). Captions with figure numbers must be placed after their associated figures, as shown in Fig. 1.

Fig. 2 Example of an unacceptable low-resolution image

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Fig. 3 Example of an image with acceptable resolution

F. Table Captions

Tables must be numbered using uppercase Roman numerals. Table captions must be centred and in 8 pt Regular font with Small Caps. Every word in a table caption must be capitalized except for short minor words as listed in Section III-B. Captions with table numbers must be placed before their associated tables, as shown in Table 1.

E. Page Numbers, Headers and Footers

Page numbers, headers and footers must not be used.

F. Links and Bookmarks

All hypertext links and section bookmarks will be removed from papers during the processing of papers for publication. If you need to refer to an Internet email address or URL in your paper, you must type out the address or URL fully in Regular font.

G. References

The heading of the References section must not be numbered. All reference items must be in 8 pt font. Please use Regular and Italic styles to distinguish different fields as shown in the References section. Number the reference items consecutively in square brackets (e.g. [1]).

When referring to a reference item, please simply use the reference number, as in [2]. Do not use “Ref. [3]” or “Reference [3]” except at the beginning of a sentence, e.g. “Reference [3] shows …”. Multiple references are each numbered with separate brackets (e.g. [2], [3], [4]–[6]).

Examples of reference items of different categories shown in the References section include:

1) example of a book in [1]
2) example of a book in a series in [2]
3) example of a journal article in [3]
4) example of a conference paper in [4]
5) example of a patent in [5]
6) example of a website in [6]
7) example of a web page in [7]
8) example of a databook as a manual in [8]
9) example of a datasheet in [9]
10) example of a master’s thesis in [10]
11) example of a technical report in [11]
12) example of a standard in [12]

III.CONCLUSIONS

The version of this template is V2. Most of the formatting instructions in this document have been compiled by Causal Productions from the IEEE LaTeX style files. Causal Productions offers both A4 templates and US Letter templates for LaTeX and Microsoft

Word. The LaTeX templates depend on the official IEEEtran.cls and IEEEtran.bst files, whereas the Microsoft Word templates are self-contained. Causal Productions has used its best efforts to ensure that the templates have the same appearance.

Causal Productions permits the distribution and revision of these templates on the condition that Causal Productions is credited in the revised template as follows: "original version of this template was provided by courtesy of Causal Productions (www.causalproductions.com)".

IV. ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered.

Causal Productions wishes to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files which have been used in the preparation of this template. To see the list of contributors, please refer to the top of file IEEETran.cls in the IEEE LaTeX distribution.

REFERENCES

[1] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[4] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.

[5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[6] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

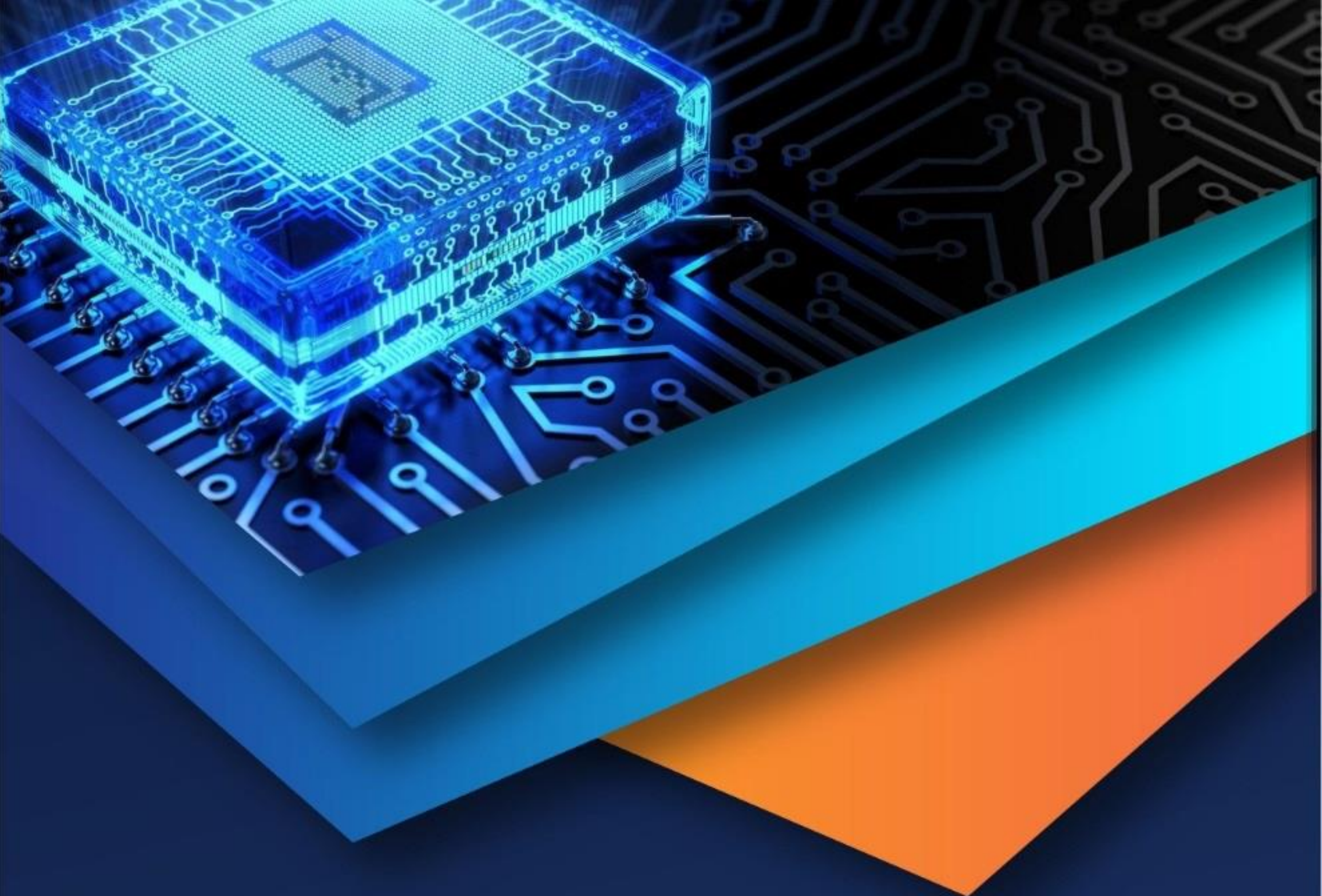
[8] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.

[9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[12] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY