International Journal of Engineering & Technology, 7 (1.2) (2018) 117-124

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A fuzzy energy and security aware scheduling in cloud

Sirisati Ranga Swamy [1] *, Sridhar Mandapati [2]

[1] Research Scholar, Dept of Computer Science, Acharya Nagarjuna University, Guntur, Andhra Pradesh, INDIA
[2] Associate Professor, Dept of Computer Applications, RVR & JC College of Engineering, Guntur, Andhra Pradesh, INDIA
**Corresponding author E-mail: sirisatiranga@gmail.com*

Abstract

The cloud computing is the one that deals with the trading of the resources efficiently in accordance to the user's need. A Job scheduling is the choice of an ideal resource for any job to be executed with regard to waiting time, cost or turnaround time. A cloud job scheduling will be an NP-hard problem that contains n jobs and m machines and every job is processed with each of these m machines to minimize the make span. The security here is one of the top most concerns in the cloud. In order to calculate the value of fitness the fuzzy inference system makes use of the membership function for determining the degree up to which the input parameters that belong to every fuzzy set is relevant. Here the fuzzy is used for the purpose of scheduling energy as well as security in the cloud computing.

Keywords: *Cloud Computing; Job Scheduling; Cloud Job Scheduling; Fuzzy Inference System; Energy and. Security.*

1. Introduction

Advancement of technology with the manner in which the organizations run business in the cloud era is a clear evidence in cloud computing which will be the future platform for both business and individual needs of computing. The cloud computing will have gained some massive attention as well as popularity in IT as most of the organizations, individuals and companies will not be in a position to buy resources both hardware and software owing to a high cost that is involved in procuring them [1]. A Cloud computing is now increasingly becoming a platform that is widely preferred for both the individuals and the organizations and it has also been able to bring down the barriers of several high costs that are associated with the required hardware and software.

Cloud computing now has a converted storage, like the 'pay as you go service' that has enabled the organizations and the individuals not having the need to purchase of their resources both hardware and software or even for the purpose of paying for the cost of maintenance. The cloud computing has an impact that is felt in the day to day activities and has also impacted the lives in a wider range of innovations that are completely transformed based on how the services are being produced, priced or even delivered. The organizations and the individuals will drive this on the minimization of cost, the meeting of the demands of customers and the sharing of availability of data. The basic problems with that of a cloud computing environment will be the resource allocation and the cloudlets scheduling.

Security is considered extremely important for the cloud users leading to the relinquishment of the physical possession for the data computation that will pose the risk in terms of security which is towards both confidentiality and the integrity. There is plenty of research that is initiated in the provisioning of hypervisors and there are also certain other problems that have to be addressed for the real time tasks and the security aware tasks that run on the virtual machines. By using the services of security a overhead for the computation time that can violate the deadlines of applications can happen [2].

This type of a distributed nature in the cloud model will involve more such transits over the data and their networks leading to more challenging security risks. The cloud computing will be realized by means of the technologies including virtualization thereby securing the virtual machines that are the key to the securing of the cloud applications. Even though there are major concerns in terms of security that have been associated with that of the virtual machines which are similar but not identical to the ones that are on the physical platforms and the Virtual Machines(VMs) have some potential weaknesses owing to the basic nature of the multi-tenancy support. The virtual machines will permit users to share their resources of that of the host computer and also provide some isolation among the VM and its host. But due to the limitations of architecture the VM and its vendor's approach to isolation or the bugs in the virtualization software that can lead to the ability of compromising the isolation and therefore demands a stringent security check [3].

Another straightforward method for achieving the modelling of such parameters is using fuzzy logic. The scheduling of real-time threads on that of the multi-core processors that have been based on fuzzy inference used for the scheduling of the operating system or the design of the hardware controller has been duly researched. There is a method for the definition of smooth membership functions (MFs) for the deadlines as well as the execution that is proposed by using the mixed cubic-exponential Hermite interpolation parametric curves along with the effect of the changes in the MFs that are parameterized based on the task scheduling and the task priorities that have been reported [4].

Most such researches that concern the real-time system scheduling will assume the scheduling constraints and in most such situations the values of such parameters will suggest the usage of fuzzy logic for deciding as to in what order of the requests will have to be executed for using the system better and therefore bringing down the chances of the request getting delayed or even missed. Therefore, the scheduling parameters will be treated as the fuzzy variables and in this paper there is a complex and a real world scheduling of the jobs that are soft and non-periodic that are being executed within a single processor and its real-time system that makes use of the fuzzy logic algorithm that has been developed. This algorithm has made an attempt to deliver the jobs on time for satisfying the demands of customers.

2. Literature survey

The Scheduling of the user tasks is found to be a process in the environment of cloud computing. The Min-min algorithm and its shortcoming will be the long tasks that are not scheduled. So Liu et al [5] made a proposal of an improved algorithm that has been based on the min-min algorithm along with three other constraints which are the quality of service, the model of dynamic priority and finally the cost of service. By means of making use of the Cloud Sim software is used in simulation for running the experiment and it is compared with the min-min algorithm the results of experiment will show executing of long tasks within a reasonable time ,the increase in the utilization of resources and the requirements hasmeet the users demand.

Konjaang et al [4] made a proposal of another new mechanism known as the Modified Max-Min (MMax-Min) algorithm that is inspired from the Max-Min algorithm. This proposed algorithm has found a cloudlet that has a maximum completion time along with a minimum completion time which either assigns the cloudlets for the execution that is dependent on the specifications for boosting the cloud scheduling process and also for increasing throughput. From the simulation results that use a CloudSim, it has shown that this proposed approach can produce solutions of good quality and some good values of make span and the balancing of the load more effectively compared to a standard Round Robin and a Max-Min algorithm.

The Job scheduling (which is allocation of hardware resources to their tasks) is a very challenging issue in cloud computing and a good policy of scheduling will help in the effective utilization of the Virtual Machines (VMs). This Job scheduling algorithm is one NP-complete problem that will place a very important role in cloud computing. Mohammadi et al [6] further made a survey on the existing type of algorithms of job scheduling in a cloud computing environment and also summarized methods for improving their performance.

Pandey and Singh [7] made a proposal for an algorithm of Job scheduling to ensure the fairness of the allocation of resources in accordance to that of the Quality of Service. This focusses mainly on two different problems one being the selection of virtual machines eligible for executing tasks. One more problem will be the task justification in accordance to the quality of service. This proposed approach has simplified the algorithm’s complexity with a reduced overhead that is associated with the choice of an appropriate as well as a justified virtual machine for any task. This also ensures a fairness in the allocation of resources for all such classified tasks thereby justifying the allocation of the overall system. Furthermore, it uses fuzzy logic for adjusting the expectation of the vector of the task that is based on the allocation and its fairness.

Fahmy [8] made an implementation of a fuzzy logic algorithm which will assume a heavily loaded machine along with a single processor that is shared by the multi users. The jobs are assumed to have various times of processing, wasting time and different priority of exterior. The jobs have also assumed the non-periodic that have the deadlines for execution. The main idea in this is to make use of the algorithm of fuzzy logic for investigating the priority of the job that has to be executed first. The second fuzzy algorithm will be used for adapting the priorities of the other jobs waiting in that case of an arrival of a new job keeping the deadlines in mind. The fuzzy logic load and its scheduling algorithm will be used inside the multi-objective algorithm for bringing down the average delay, the number of jobs that are overdue and throughput times of the job. The Objectives will be to reduce the total time of throughput and improve the satisfaction of the user. The test cases having various job factors that are used for evaluating the algorithms and also demonstrating its strength are used and the jobs that have less processing time will be assigned some newer priorities of processing thereby improving the user satisfaction.

Guo et al., [9] presented a Workflow Task Scheduling algorithm based on the resources' Fuzzy Clustering named FCBWTS. The major objective of scheduling was to be minimized makespan of the precedence constrained applications, which can be modeled as a directed acyclic graph. In FCBWTS, the resource characteristics of cloud computing are considered, a group of characteristics, which described the synthetic performance of processing units in the resource system. With these characteristics and the execution time influence of the ready task in the critical path, processing unit network was pretreated by fuzzy clustering method in order to be realized the reasonable partition of processor network. Therefore, it largely reduced the cost in deciding which processor to execute the current task.

Alla et al., [10] focused on Task Scheduling using a novel architecture with Dynamic Queues based on hybrid algorithm using Fuzzy Logic and Particle Swarm Optimization algorithm (TSDQ-FLPSO) to be optimized makespan and waiting time. The experimental result based on an open source simulator (CloudSim) show that the proposed TSDQ-FLPSO provides an optimal balance results, minimizing the waiting time, reducing the makespan and improving the resource utilization compared to existing scheduling algorithms.

Xiaojun et al., [11] proposed a new cloud computing resources allocation strategy, on the basis of cost model of the parallel task performance, using the improved Fuzzy Clustering Algorithm (CBFCM - C Cost based Fuzzy Clustering Algorithm) to be classified the cloud computing resources set on the service level of user, and the jobs submitted by guest were analyzed using four different scheduling algorithms, and the generated resource scheduling scheme. The experimental results show that job scheduling strategy based on CBFCM resources classification in the cloud computing services have certain advantages. Adaptive First Come Fist Served (AFCFS) job scheduling algorithm in Average Response Time (ART), the Average Waiting Time (AWT), Average Operating Cost (ACP) performance items, has great advantage than the other three algorithm.

Priya&Babu [12] proposed a method called, Moving Average-based Fuzzy Resource Scheduling (MV-FRS) for virtualized cloud environment to be optimized the scheduling of resources through virtual machines. Initially, the MV-FRS method starts by predicting the resource (i.e. bandwidth, memory and processing cycle) requirements. Then a measure of relationships between availability of resources and the requirements of resources are made. Finally, a fuzzy control theory is designed to accomplish system accessibility between user cloud requirements and cloud users resources availability. The simulations results demonstrated that the MV-FRS method was able to be reduced the total waiting time of cloud user resource requirements and also ensure the feasibility and effectiveness of the overall system accessibility in terms of average success rate and resource usage when running in a cloud computing environment.

Singh & Chana [13] presented the fuzzy logic based energy-aware autonomic resource scheduling framework for cloud for energy efficient scheduling of cloud computing resources in data centers. It have been evaluated the proposed framework in CloudSim based simulation environment and real cloud environment. The experimental results show that the proposed framework performs better in terms of resource utilization and energy consumption along with other Quality of Service (QoS) parameters.

Panda & Jana [14] presented the three task scheduling algorithms, called Minimum Completion Cloud (MCC), MEdian MAX (MEMAX) and Cloud Min–Max Normalization (CMMN) for heterogeneous multi-cloud environment, which aim to be minimized the makespan and maximized the average cloud utilization. The proposed MCC algorithm was a single-phase scheduling whereas rests are two-phase scheduling. It performed rigorous experiments on the proposed algorithms using various benchmark as well as synthetic datasets. Their performances are evaluated in terms of makespan and average cloud utilization and experimental results are compared with that of existing single-phase and two-phase scheduling algorithms to be demonstrated the efficacy of the proposed algorithms.

Kumar & Nagarajan [15] provided a secure communication between one cloud parties to another cloud provider. A public key will be generated by key generator in the cloud for every registered user to communicate secretly. The number of cloud users in cloud computing are growing in an exponential rate at every day. Due to this, many users use the same resource in the cloud. In this situation, the server troubles with high load. To reduce the high load problem, we can use Priority Based Scheduling Algorithm (PBSA) and Load balancing algorithm. Exchanging the information from one to another is called data communication. Security is the main problem in data communication. We can solve this problem by using Advanced Encryption Standard (AES) algorithm. In this two algorithms achieve the security and resource allocation in cloud computing. This Concept was achieved by FUZZY logic with the help of MATLAB tool.

3. Methodology

Here in this section, details of the Min-Min, the Max-Min algorithm along with the proposed fuzzy scheduling has been presented.

3.1. Min-min algorithm

The Min-Min algorithm will consider all unassigned task for each such task mapping. The Min-min executes the tasks that are parallel as well as the long tasks will be executed until the entire task set is found to be empty. The Min-min algorithm executes parallely the short tasks that are in parallel and long tasks will follow these short tasks. The Min-min algorithm has a shortcoming that the short tasks are first scheduled before executing the long tasks. The Min-min cause the entire batch tasks that are executed for getting longer and an unbalanced load. Even such long tasks will not be executed. The actual purpose of the Min-Min algorithm [16] will be to ensure that these tasks having a minimum time for completion in parallel and this will further be done to their long jobs once the short task is successfully executed but the main disadvantage is that at time the long tasks remain unattended.

There are two different phases in the Min-Min algorithm. In the first phase it finds the minimum execution time of all tasks. Then in the second phase it chooses the task with the least execution time among all the tasks. The algorithm proceeds by assigning these tasks to their resource which will produce their completion time and this same procedure will be repeated by that of the Min-Min until all tasks which are scheduled.

There are some more limitations of Min-Min algorithm:

```
1)  If task queue T is not Empty, then continue to execute or execute step 13
2)  For (all tasks $t_i$ in task set T)
3)  For (all resources $r_j$ in resource set R)
4)  $C_{ij} = tr_{ij} + tw_{ij}$
5)  End for
6)  End for
7)  For (every elements in C)
8)  Searching tasks $t_i$ with maximum and earliest complete time, relative resource $r_j$
9)  End for
10) Assigning tasks $t_i$ to resource $r_j$
11) Removing task $t_i$ from T
12) Updating $t_w$, return to step 1
13) Exiting loop
```

The resources that have high computational power are utilized by the selected smaller tasks.

- Only a single job can be executed by a single resource at a time.
- The size and the number of resources will be static and known earlier [17].

3.2. Max-min algorithm

The Max-Min [18] will be both a resource allocation and a scheduling algorithm that is used in case of cloud computing for minimizing the make span, cost and also the maximization of profit and the utilization of resource. This will be done by means of choosing the task in a job list having a high completion time based on a resource which may execute this inside a much shorter period. The main concern here is to provide priority to those tasks that have a maximum completion time by means of executing them initially before assigning them to the other tasks having a minimal completion time. But the disadvantage of the Max-Min algorithm will be the execution of such tasks having a maximum completion time that can increase the total time of response for the system making it more efficient.

```
1)  If task queue T is not Empty, then continue to execute or execute step 13
2)  For (all tasks $t_i$ in task set T)
3)  For (all resources $r_j$ in resource set R)
4)  $C_{ij} = tr_{ij} + tw_{ij}$
5)  End for
6)  End for
7)  For (every elements in C)
8)  Searching tasks $t_i$ with minimum and earliest complete time, relative resource $r_j$
9)  End for
10) Assigning tasks $t_i$ to resource $r_j$
11) Removing task $t_i$ from T
12) Updating $t_w$, return to step 1
13) Exiting loop
```

3.3. Proposed fuzzy

There is a dynamic algorithm of scheduling that makes use of the fuzzy logic controller. This controller will take advantage of two different inputs which are the actual number of requests that have been received in the host from that of the virtual machines and also the value of the tasks and their precedence running in each such virtual machine. The actual number of requests is mentioned as the AGE along with the values of the tasks precedence will be named as the PREC for simplicity. The output of such a fuzzy controller will be the virtual machine id that is assigned to that of the host. By using the scheduling scheme, these virtual machines that are fairly assigned to that of the host in accordance to the time slot of the waiting in the queue along with their precedence [19]. In case of simulation, every task will have the precedence value from 0 to 3 and this AGE value will range between 0 and 3 for four different virtual machines. The Design of such fuzzy control routing systems will contain of a series of steps which are:

Step 1: Define both the input and the output variables, and the number of requests received in the host from the virtual machines along with the value of tasks precedence running on all virtual machines and ID of the virtual machine which has to be assigned to its host as their output.

Step2: Each such variable is quantified, for example the AGE are the requests that are sent to a host and set it to 0, 1, 2 and 3 (the four virtual machines), it is quantified as a Zero, a Small, a Medium and also a Big. Each such quantification of variables is assigned one membership function the cost of which are singletons.

Step3: The fuzzy rule base is designed that determines the action that takes place.

Step4: A method of defuzzification is applied to fuzzy control actions for producing crisp costs and this method will be the "centers of gravity" used for a crisp output.

Step5: These virtual machine IDs having a low cost of AGE and the PREC values is chosen for the output in order to access its host.
For any first fuzzy algorithm [20], the membership functions are three in number which are high, medium and low. An inference mechanism in a fuzzy logic algorithm will resemble the reasoning process of humans. In this the fuzzy logic technology will be associated with the artificial intelligence.

3.3.1. Project/task description

The fuzzy system designed consists of four input variables for the scheduling process, one output variable that is used to decide the job selected based on the fuzzy rules. The input variables defined in this system are job length assigned to represent the length of the job, VM memory assigned to represent the available memory in virtual machine, security level to represent the level of security available in the fuzzy based cloud, Energy required to represent the required energy to execute the tasks requested by the user [21]. The output variable is defined as Result that is used to decide the scheduled job assignment selected to a particular task. The parameters and the details for the project used in this system are also shown in the table 1.

Table 1: Project Statistics

Input Variables	4
Output Variables	1
Intermediate Variables	0
Rule Blocks	1
Rules	405
Membership Functions	17

3.3.2. System structure

The structure of this system will identify the fuzzy logic and its inference flow from both the input variables and the output variables. This fuzzification in the input and their interfaces will translate the analog inputs into the fuzzy values and the fuzzy inference will take place in that of the rule blocks that contain the rules of linguistic control. The total output of rule blocks will be the linguistic variables and the defuzzification in these output interfaces will translate these linguistic variables among the analog variables. The aim of improving the utilization of the resources by scheduling is achieved.
The entire structure of the fuzzy system that includes the input variables, input interfaces, the rule blocks and the output interfaces is given in figure 1. Their connecting lines hereby symbolize the data flow. The fuzzy inference system consists of three steps including fuzzification, rule evaluation and defuzzification [22].
The process of mapping the real values to categorical or linguistic variables which represents the characteristics of the input is called as fuzzification. The conversion takes place with the help of membership function. The membership function maps the input value to a value between 0 and 1. Rule evaluation is a decision structure to determine the rules which is in the form “if”- “then” rules. Certain operators are required for determining the rules along with the expert knowledge. The 'if' part will describe the situation based on the domain and the 'then' part will provide the response of their fuzzy system.
Defuzzification transposes the fuzzy outputs to crisp outputs. There are many methods used in defuzzificztion. Some of them are Center of Maximum (COM), MOM, Center of Gravity (COG). The fuzzy outputs are transposed to their membership functions similarly as in fuzzification. Output of the inferencing system is a crisp output that is used to process or control events or element elsewhere [23]

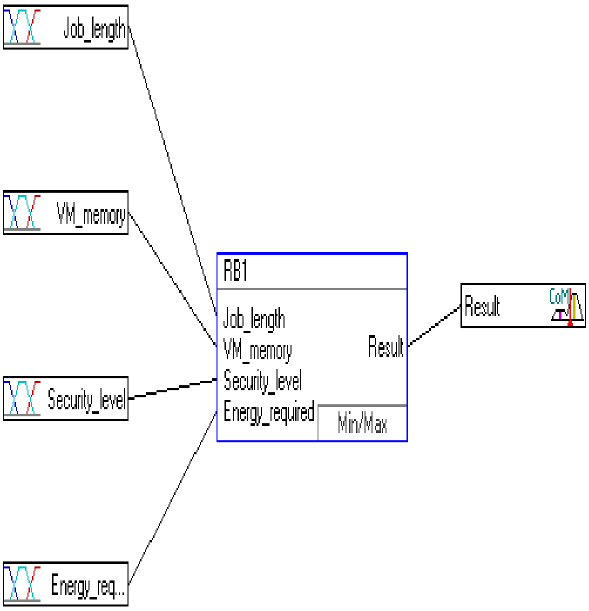

Fig. 1: Structure of the Fuzzy Logic System.

3.3.3. Input variables

The input variables are the requirements of tasks’ to utilize available resource of users’ virtual machines. The fuzzy values of the input variables are described as low, medium and high. The details of the variables job length, VM memory, security level, Energy required are given in table 2.

Table 2: Variables of Group "Inputs"

#	Variable Name	Type	Unit	Min	Max	Default	Term Names
1	Energy_required		Units	0	1	0.5	Low medium high
2	Job_length		Units	0	1	0.5	Low medium high
3	Security_level		Units	0	1	0.5	Low medium high
4	VM_memory		Units	0	1	0.5	Low medium high

3.3.4. Output variables

The output variable is the decision to assign a job to corresponding VM. The fuzzy values of the output variables are described as very_low, low, medium, high and very_high. The details of the variableResult is given in the table 3.

Table 3: Variables of Group "Outputs"

#	Variable Name	Type	Unit	Min	Max	De-fault	Term Names
5	Result	CoM	Units	0	1	0.5	very_low low medium high very_high

The default value of an output variable is used if no rule is firing for this variable. Different methods can be used for the defuzzification, resulting either into the 'most plausible result' or the 'best compromise'.
The best compromise is produced by the methods:
CoM (Center of Maximum)
CoA (Center of Area)

CoA BSUM, a version especially for efficient VLSI implementations

The most plausible result is produced by the methods:

MoM (Mean of Maximum)

MoM BSUM, a version especially for efficient VLSI implementations

3.3.5. Input variable “energy_required”

The membership function maps the crisp values from its specified domain to the seys related to the degree of membership that is designed. Membership functions is represented graphically to represent a fuzzy set. The shape of the membership function is chosen in such a way that it represents the defined variables. The selection of the shape is also dependent on the computing resources that are available.

The triangular membership function is defined by three parameters lower limit (a), upper limit (b) and the peak value (m), where a < m < b.

The Trapezoidal function is defined by four parameters lower limit (a), upper limit (d) and a lower support limit (b), and an upper support limit (c), where a < b < c < d.

Three membership functions for the input variable Energy required is defined for the fuzzy values low, medium and high. The membership function for low and high value is defined as a trapezoidal function. The membership function for medium value is defined as a triangular function and is shown in Figure 2. The definition points of the membership functions are given in the table 4.

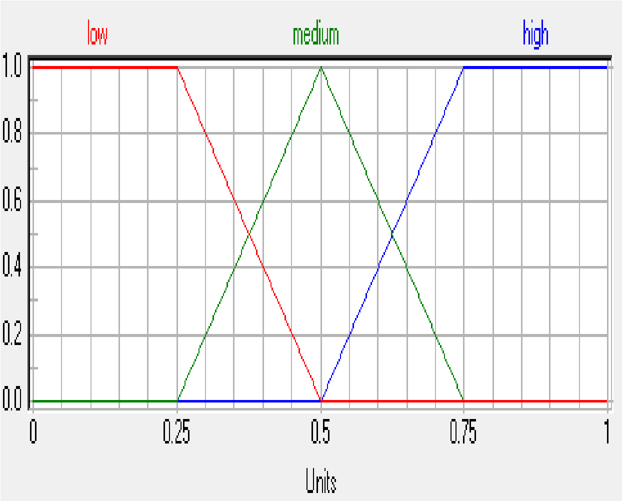

Fig. 2: MBF of "Energy_required".

Table 4: Definition Points of MBF "Energy_required"

Term Name	Shape/Par.	Definition Points (x, y)
Low	Linear	(0, 1) (0.25, 1) (0.5, 0) (1, 0)
Medium	Linear	(0, 0) (0.25, 0) (0.5, 1) (0.75, 0) (1, 0)
High	Linear	(0, 0) (0.5, 0) (0.75, 1) (1, 1)

3.3.6. Input Variable “Job_length”

Three membership functions for the input variable Job_length is defined for the fuzzy values low, medium and high. The membership function for low and high value is defined as a trapezoidal function. The membership function for medium value is defined as a triangular function and is shown in Figure 3. The definition points of the membership functions are given in the table 5.

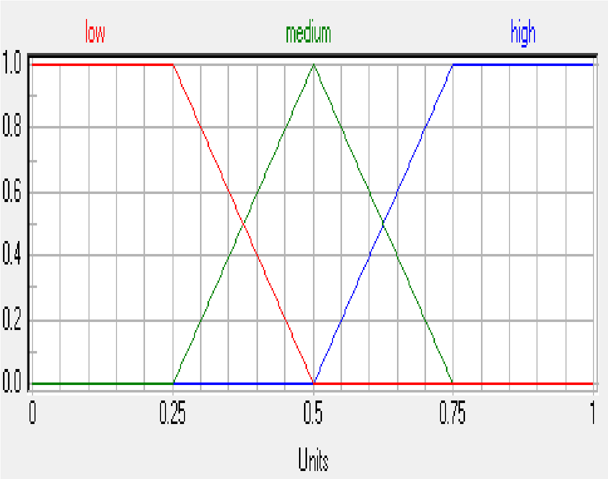

Fig. 3: MBF of "Job_length".

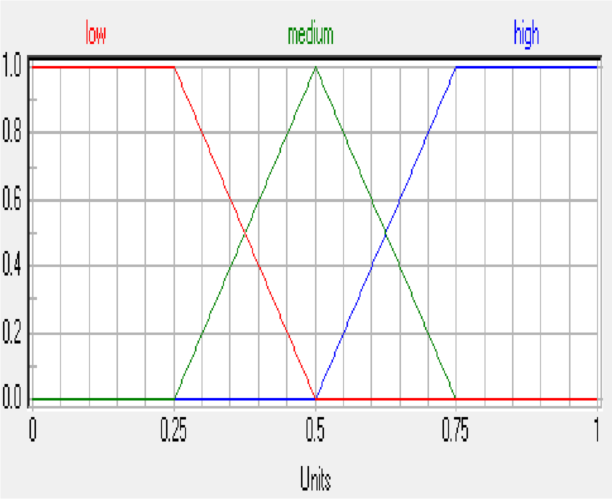

Fig. 4: MBF of "Security_level".

Table 5: Definition Points of MBF "Job_length"

Term Name	Shape/Par.	Definition Points (x, y)
Low	Linear	(0, 1) (0.25, 1) (0.5, 0) (1, 0)
medium	Linear	(0, 0) (0.25, 0) (0.5, 1) (0.75, 0) (1, 0)
High	Linear	(0, 0) (0.5, 0) (0.75, 1) (1, 1)

3.3.7. Input variable “security_level”

Three membership functions for the input variable Security level is defined for the fuzzy values low, medium and high. The membership function for low and high values are defined as a trapezoidal function. The membership function for medium value is defined as a triangular function and is shown in Figure 4. The definition points of the membership functions are given in the table 6.

Table 6: Definition Points of MBF “Security_length”

Term Name	Shape/Par.	Definition Points (x, y)
Low	Linear	(0, 1) (0.25, 1) (0.5, 0) (1, 0)
medium	Linear	(0, 0) (0.25, 0) (0.5, 1) (0.75, 0) (1, 0)
High	Linear	(0, 0) (0.5, 0) (0.75, 1) (1, 1)

3.3.8. Input variable “VM_memory”

Three membership functions for the input variable VM_memory is defined for the fuzzy values low, medium and high. The membership function for low and high values are defined as a trapezoidal function. The membership function for medium value is defined as a triangular function and is shown in Figure 5. The definition points of the membership functions are given in the table 7.

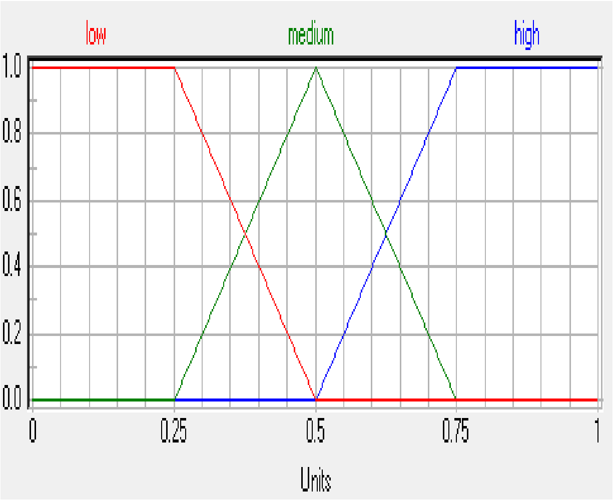

Fig. 5: MBF of "VM_Memory".

Table 7: Definition Points of MBF "VM_Memory"

Term Name	Shape/Par.	Definition Points (x, y)
Low	Linear	(0, 1) (0.25, 1) (0.5, 0) (1, 0)
medium	Linear	(0, 0) (0.25, 0) (0.5, 1) (0.75, 0) (1, 0)
High	Linear	(0, 0) (0.5, 0) (0.75, 1) (1, 1)

3.3.9. Output Variable "Result"

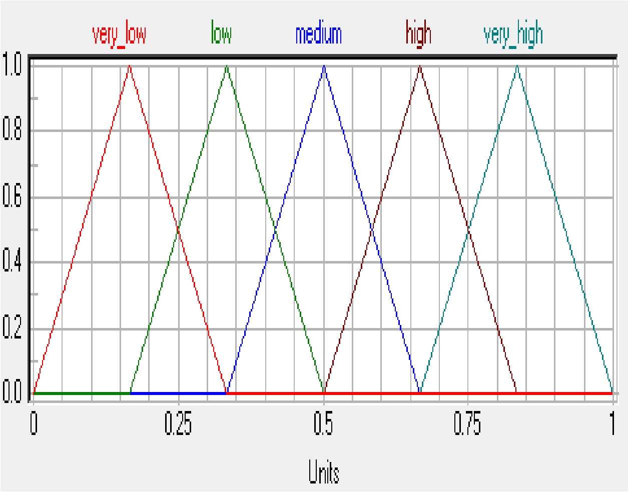

Fig. 6: MBF of "Result".

Five membership functions for the output variable Result is defined for the fuzzy values very_low, low, medium, high and very_high. The membership function for all the fuzzy values are defined as a triangular function and is shown in Figure 6. The defuzzification method used here is the Center of Gravity method or COG by computing the centroids of sets. The output membership functions converts the fuzzy outputs to crisp outputs and are restricted to being singletons to limit the degree of computation.These singleton values of a output is are computed by using weighted average technique. The definition points of the membership functions are given in the table 8.

Table 8: Definition Points of MBF "Results"

Term Name	Shape/Par.	Definition Points (x, y)
very_low	Linear	(0, 0) (0.16666, 1) (0.33334, 0) (1, 0)
Low	Linear	(0, 0) (0.16666, 0) (0.33334, 1) (0.5, 0) (1, 0)
medium	Linear	(0, 0) (0.33334, 1) (0.5, 1) (0.16666, 0) (1, 0)
High	Linear	(0, 0) (0.5, 0) (0.16666, 1) (0.83334, 0) (1, 0)
very_high	Linear	(0, 0) (0.16666, 0) (0.83334, 1) (1, 0)

3.4. Rule Blocks

Rule Blocks:
These rule blocks will contain control strategies for fuzzy logic systems. Each such rule blocks will confine the rules for this context and such a context will be defined by a similar input and output variables of such rules.
These rules of the 'if' part will describe the situation, in which the rules will be designed and the 'then' part will describe the response of their fuzzy system for this situation and the degree of support (DoS) will be used for weighing every rule in accordance to its importance.
These processing of such rules will begin with the 'if' part. The main operator type of this rule block will determine as to which method can be used. The operator type will be the MIN-MAX, MIN-AVG and the GAMMA that are available. The trait of every operator type has been influenced by that of an additional parameter.
For instance:
MIN-MAX, parameter value 0= Minimum Operator (MIN)
MIN-MAX, parameter value 1 = Maximum Operator (MAX)
GAMMA, parameter value 0 = Product Operator (PROD)
The actual minimum operator will be a generalization of this Boolean 'and'; a maximum operator is the generalization of that of the Boolean 'or'.
A fuzzy composition that eventually combines these different rules to that of one single conclusion. In case the BSUM method is being used all the firing rules get evaluated, and in case the MAX method is being used only dominant rules will be evaluated.
Rule Block "RB1":
The rule block that is used for improving the utilization of the resources by scheduling is MINMAX. The rule block consists of 405 rules with four input variables that are nothing but the requirements of tasks' to utilize available resource of users' virtual machines. The fuzzy inference system uses one output variable to take decision to assign a job to corresponding VM.
Parameter
Aggregation: MIN MAX
Parameter: 0.00
Result Aggregation: MAX
Number of Inputs: 4
Number of Outputs: 1
Number of Rules: 405

4. Result and discussion

The Tables 9 to 11 and the figures 7 to 9 will show the makespan, the Degree of Imbalance and the resource utilization for the proposed fuzzy.

- Degree of Imbalance (DI): The Degree of imbalance will be the factor that is calculated by Equation (1):

$$DI = \frac{T_{\max} - T_{\min}}{T_{avg}} \quad (1)$$

in which $T_{\max}$, $T_{\min}$, and T_{avg} denote the maximum, the minimum, and the average total execution times, among all the VMs.

- Makespan: A very popular criterion for optimization is the minimization of its make span that indicates the time when the last job gets over and the small make span values will mean that a scheduler will provide a good as well as an efficient assignments of the jobs to the resources. Generally, the make span will be calculated using equation (2):

$$\min_{S_j \in Schedlist} \left\{ \max_{j \in joblist} Fin_j \right\} \quad (2)$$

In which Fin_j is the time when j the job finalizes, s_j the current schedule list, and Schedule list the set of possible schedules, and the job list the list of jobs being ready

Table 9: Makespan for Proposed Fuzzy

Number of Jobs	Min-Min	Max-Min	Proposed Fuzzy
200	47	46	44
400	102	99	92
600	162	158	147
800	208	204	193
1000	265	258	247

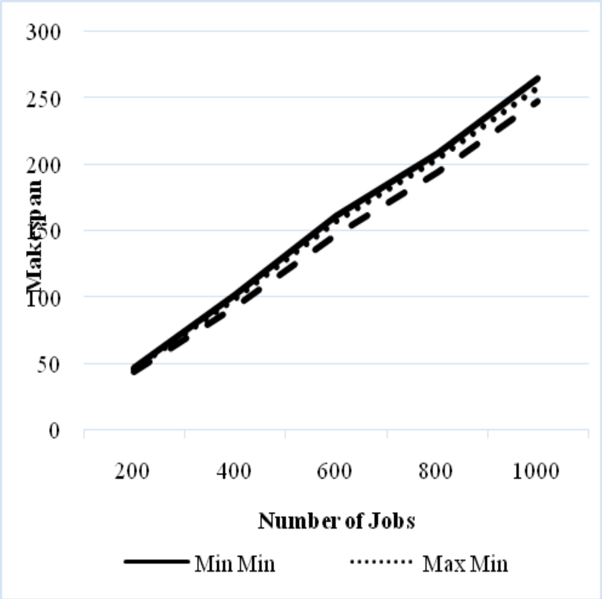

Fig. 7: Makespan for Proposed Fuzzy.

Table9 and figure7 shows that the Makespan for proposed Fuzzy performs better than Min-Min by 6.59%, by 10.31%, by 9.71%, by 7.48% and by 7% at number of jobs 200, 400, 600, 800 and 1000 respectively. Also the Makespan for proposed Fuzzy performs better than Max-Min by 4.44%, by 7.33%, by 7.2%, by 5.54% and by 4.36% at number of jobs 200, 400, 600, 800 and 1000 respectively.

Table 10: Degree of Imbalance for Proposed Fuzzy

Number of Jobs	Min-Min	Max-Min	Proposed Fuzzy
200	2.9	2.7	2.5
400	2.7	2.6	2.4
600	2.4	2.3	2.1
800	2.3	2.1	2
1000	2.2	2	1.9

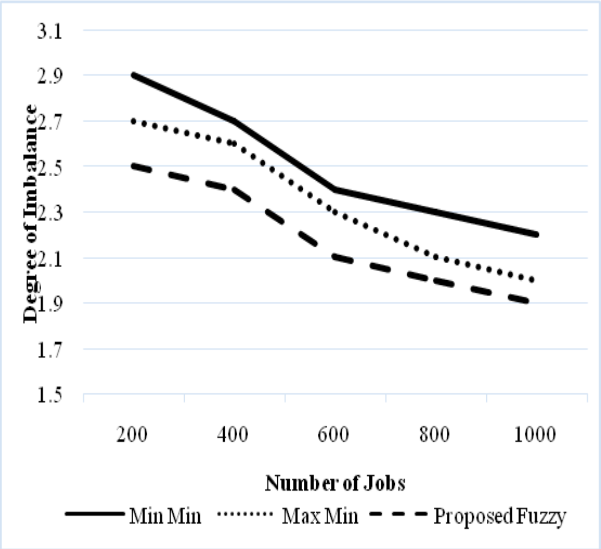

Fig. 8: Degree of Imbalance for Proposed Fuzzy.

Table 10 and figure 8 shows that the degree of imbalance for proposed Fuzzy performs better than Min-Min by 14.8%, by 11.76%, by 13.3%, by 13.95% and by 14.63% at number of jobs 200, 400, 600, 800 and 1000 respectively. Also the degree of imbalance for proposed Fuzzy performs better than Max-Min by 7.69%, by 8%, by 9.1%, by 4.88% and by 5.13% at number of jobs 200, 400, 600, 800 and 1000 respectively.

Table 11: Resource Utilization for Proposed Fuzzy

Number of Jobs	Min-Min	Max-Min	Proposed Fuzzy
200	76	77	79
400	74	75	78
600	77	78	81
800	78	79	82
1000	76	77	79

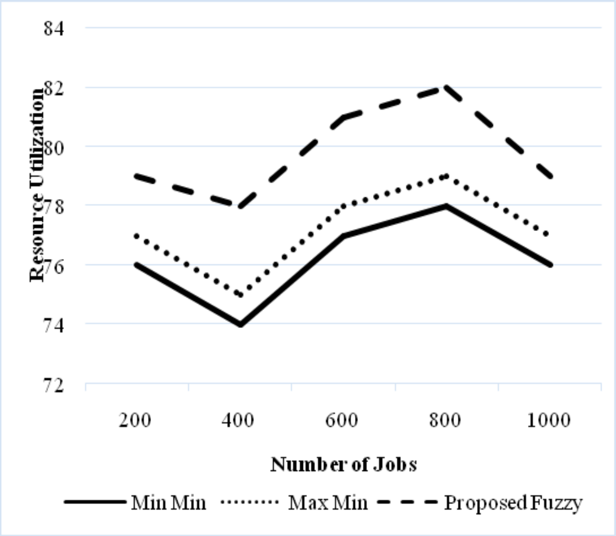

Fig. 9: Resource Utilization for Proposed Fuzzy.

Table 11 and figure 9 shows that the resource utilization for proposed Fuzzy performs better than Min-Min by 3.87%, by 5.26%, by 5.1%, by 5% and by 3.87% at number of jobs 200, 400, 600, 800 and 1000 respectively. Also the resource utilization for proposed Fuzzy performs better than Max-Min by 2.56%, by 3.92%, by 3.8%, by 3.73% and by 2.56% at number of jobs 200, 400, 600, 800 and 1000 respectively.

5. Conclusion

The Scheduling is the manner in which the tasks get assigned to be run on the resources available and the cloud will submit its request with the user request that is formed as tasks. Such tasks are scheduled to their corresponding services and they may need various security services like integrity, confidentiality and its authentication having various levels that are denoted by users. The results have proven that the Makespan for this proposed Fuzzy has performed better than that of the Min-Min by about 6.59%, by about 10.31%, by 9.71%, by about 7.48% and by about 7% at number of the jobs 200, 400, 600, 800 and 1000 respectively. Further this Makespan for the proposed Fuzzy performs much better than that of the Max-Min by 4.44%, by about 7.33%, by 7.2%, by about 5.54% and by 4.36% at the number of jobs 200, 400, 600, 800 and finally 1000 respectively.

References

[1] Konjaang, J., Ayob, F. H., & Muhammed, A. (2017). An Optimized Max-Min Scheduling Algorithm in Cloud Computing. Journal of Theoretical & Applied Information Technology, 95(9).

[2] Chandran, K., Shanmugasudaram, V., & Subramani, K. (2016). Designing a fuzzy-logic based trust and reputation model for secure resource allocation in cloud computing. Int. Arab J. Inf. Technol., 13(1), 30-37.

[3] Mann, Z. Á. (2015). Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms. ACM Computing Surveys (CSUR), 48(1), 11. https://doi.org/10.1145/2797211.

[4] Muhuria PK, Shukla KK. Real-time scheduling of periodic tasks with processing times and deadlines as parametric fuzzy numbers. Appl Soft Comput 2009;9(3):936–46. https://doi.org/10.1016/j.asoc.2008.11.004.

[5] Liu, G., Li, J., & Xu, J. (2013). An improved min-min algorithm in cloud computing. In Proceedings of the 2012 International Conference of Modern Computer Science and Applications (pp. 47-52). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33030-8_8.

[6] FazelMohammadi, D., Jamali, S., &Bekravi, M. (2014). Survey on Job Scheduling algorithms in Cloud Computing. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2278-6856.

[7] Pandey, P., & Singh, S. Fuzzy logic based job scheduling algorithm in cloud environment.

[8] Fahmy, M. M. M. (2010). A fuzzy algorithm for scheduling nonperiodic jobs on soft real-time single processor system. Ain Shams Engineering Journal, 1(1), 31-38. https://doi.org/10.1016/j.asej.2010.09.004.

[9] Guo, F., Yu, L., Tian, S., & Yu, J. (2015). A workflow task scheduling algorithm based on the resources' fuzzy clustering in cloud computing environment. International Journal of Communication Systems, 28(6), 1053-1067. https://doi.org/10.1002/dac.2743.

[10] Alla, H. B., Alla, S. B., Ezzati, A., &Mouhsen, A. (2017). A novel architecture with dynamic queues based on fuzzy logic and particle swarm optimization algorithm for task scheduling in cloud computing. In Advances in Ubiquitous Networking 2 (pp. 205-217). Springer Singapore. https://doi.org/10.1007/978-981-10-1627-1_16.

[11] Xiaojun, W., Yun, W., Zhe, H., & Juan, D. (2015, June). The research on resource scheduling based on fuzzy clustering in cloud computing. In Intelligent Computation Technology and Automation (ICICTA), 2015 8th International Conference on (pp. 1025-1028). IEEE. https://doi.org/10.1109/ICICTA.2015.258.

[12] Priya, V., & Babu, C. N. K. (2017). Moving average fuzzy resource scheduling for virtualized cloud data services. Computer Standards & Interfaces, 50, 251-257. https://doi.org/10.1016/j.csi.2016.10.011.

[13] Singh, S., & Chana, I. (2016). EARTH: Energy-aware autonomic resource scheduling in cloud computing. Journal of Intelligent & Fuzzy Systems, 30(3), 1581-1600. https://doi.org/10.3233/IFS-151866.

[14] Panda, S. K., & Jana, P. K. (2015). Efficient task scheduling algorithms for heterogeneous multi-cloud environment. The Journal of Supercomputing, 71(4), 1505-1533. https://doi.org/10.1007/s11227-014-1376-6.

[15] Kumar, S. K., & Nagarajan, M. K. (2016). Fuzzy Logic Based Secure Data Transfer and Retrival Using PBSA and AES for Resource Allocation in Cloud. Fuzzy Systems, 8(7), 191-196.

[16] Kaur, D., & Singh, S. (2014). An Efficient job scheduling Algorithm using MINMIN and Ant Colony Concept for grid computing. International Journal of Modern Education and Computer Science (IJMECS) ISSN, 2075-0161.

[17] G. Jaspher W. Kathrine and MansoorIlaghi U, “Job Scheduling Algorithms in Grid Computing – Survey”, International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 7, September - 2012 ISSN: 2278-0181.

[18] Brar, S. S., & Rao, S. (2015). Optimizing workflow scheduling using max-min algorithm in cloud environment. International Journal of Computer Applications, 124(4).

[19] Mehranzadeh, A., & Hashemi, S. M. (2013). A novel-scheduling algorithm for cloud computing based on fuzzy logic. International Journal of Applied Information Systems (IJAIS), 5(7). https://doi.org/10.5120/ijais13-450939.

[20] Chen, Z., Zhu, Y., Di, Y., & Feng, S. (2015). A dynamic resource scheduling method based on fuzzy control theory in cloud environment. Journal of Control Science and Engineering, 2015, 34. https://doi.org/10.1155/2015/383209.

[21] Chen, Z., Zhu, Y., Di, Y., & Feng, S. (2015). A dynamic resource scheduling method based on fuzzy control theory in cloud environment. Journal of Control Science and Engineering, 2015, 34. https://doi.org/10.1155/2015/383209.

[22] Zavvar, M., Rezaei, M., Garavand, S., & Ramezani, F. (2016). Fuzzy Logic-Based Algorithm Resource Scheduling for Improving The Reliability of Cloud Computing. Asia-Pacific Journal of Information Technology and Multimedia, 5(1).

[23] Jabarzadeh, A., Rostami, M., Shahin, M., &Shahanaghi, K. (2017). Two-stage fuzzy-stochastic programming for parallel machine scheduling problem with machine deterioration and operator learning effect. Journal of Industrial and Systems Engineering, 10(3), 16-32.