


# Efficient VM Selection Strategies in Cloud Datacenter Using Fuzzy Soft Set

Nithiya Baskaran, National Institute of Technology, Tiruchirappalli, India

 <https://orcid.org/0000-0002-3346-7753>

Eswari R., National Institute of Technology, Tiruchirappalli, India

## ABSTRACT

A cloud data center is established to meet the storage demand due to the rate of growth of data. The inefficient use of resources causes an enormous amount of power consumption in data centers. In this paper, a fuzzy soft set-based virtual machine (FSS\_VM) consolidation algorithm is proposed to address this problem. The algorithm uses four thresholds to detect overloaded hosts and applies fuzzy soft set approach to select appropriate VM for migration. It considers all factors: CPU utilization, memory usage, RAM usage, and correlation values. The algorithm is experimentally tested for 11 different combinations of choice parameters where each combination is considered as fuzzy soft set and compared with existing algorithms for various metrics. The experimental results show that proposed FSS\_VM algorithm achieves significant improvement in optimizing the objectives such as power consumption, service level agreement violation rate, and VM migrations compared to all existing algorithms. Moreover, performance comparison among the fuzzy soft set-based VM selection methods are made, and Pareto-optimal fuzzy soft sets are identified. The results show that the Pareto-based VM selection improves the QoS. The time complexity of the proposed algorithm increases when it finds best VM for migration. The future work will reduce the time complexity and will concentrate on developing an efficient VM placement strategy for VM migration since it has the greater impact on improving QoS in VM placement.

## KEYWORDS

Cloud Computing, Energy Consumption, Energy Efficiency, Fuzzy Soft Set, SLA Violation Rate, VM Consolidation, VM Selection

## 1. INTRODUCTION

Information Technology (IT) infrastructures have continued to grow rapidly over the past few years to meet the demand of computational power of modern computer-intensive business and scientific applications. These computing infrastructures consume large amount of electricity which results in operating costs that exceed infrastructure costs. Large electricity-consumption reduce system reliability and durability due to the overheating problem except for overwhelming operating costs. Similarly, the problem of substantial CO<sub>2</sub> emissions which help out to the greenhouse effect (Brown, R. 2008). In 2010 the overall power consumption of the data centers in the world was calculated at

DOI: 10.4018/JOEUC.20210901.0a8

This article, published as an Open Access article on May 28, 2021 in the gold Open Access journal, Journal of Organizational and End User Computing (converted to gold Open Access January 1, 2021), is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

about 1.5%, which rose to 3% in 2016. In 2013, 260 million watts of electricity consumed by Google data centers which is quite enough to steadily power of 200,000 homes. On the other hand, 30% of cloud data centers average resource usage capacity is 10-15%.

Virtualization technology is one way of reducing a data center's power consumption. This technology uses consolidation technique for servers and Virtual Machines (VMs). The consolidation allows several VMs into one host and reducing the amount of physical hardware usage and minimizing the resource wastage. Cloud computing patterning recently developed advantage is virtualization and provides on-demand resources on a pay-as-you-go basis over the internet. So, the IT companies to drop using their computing environment's maintenance costs and the computing needs could be outsourced to the cloud. For the customers who negotiate in terms of Service Level Agreements (SLA), e.g., throughput, response time, it is great significance for cloud providers to offer Quality of Services (QoS) (Buyya, R., *et al.*, 2008). Cloud providers (e.g., Amazon EC2) must deal with power-performance trade-offs to ensure efficient resource management and provide higher resource utilization, as aggressive consolidation of VMs can lead to loss of performance.

Consolidation refers to live migration, which is the process of moving a running VM from one physical server to another. The main goal of live VM migration is to do migration without down-time and move the VMs to minimum number of hosts and switch off the idle servers to power-saving mode. VM consolidation experiences a lot of issues due to performance deprivation and this could be avoided by optimum usage of the resources. Different parameters such as data centers VM Selection, VM placement, host CPU and memory, SLA, and power consumption must be considered when it comes to consolidating and optimizing VM. There are two types of VM consolidation. The first type is known as static VM consolidation where VM size is set up in a single deed using the peak load demand of the workload. VMs are placed in the same host during their entire lifetime. Setting the VM size for the peak load demand confirms that the VM will not be overloaded. However, since the workloads can present variable demand patterns, it can lead to the idleness of the host. The second type is known as dynamic consolidation, where periodical changes in the workload demand are carried out in each VM and based on that the required configuration changes are performed.

Dynamic consolidation is performed in two necessary steps. One is to migrate VMs from underutilized hosts and put them into sleep mode to minimize the number of active hosts. Another step is migrating VMs from overloaded hosts to avoid performance degradation, which may lead to SLA violation of the quality of service requirements. Furthermore, live migration is the way to achieve energy efficiency. The main advantage of live migration is the ability to transfer VM between the hosts with a near to zero downtime. In the real world, the computation demand is very dynamic, and that is why the decision depends on several criteria.

This paper focuses on efficient selection of VM for migration. A fuzzy soft set based VM selection algorithm is proposed to achieve minimum energy consumption and to maintain the SLA violation at a minimal level. The algorithm first identifies the overloaded hosts classifying the hosts using four-thresholds. It finds the 11 combination of selection policies, fuzzy soft sets, using fuzzy soft set theory such as FSS\_P1, FSS\_P2, FSS\_P3, FSS\_P4, FSS\_P5, FSS\_P6, FSS\_P7, FSS\_P8, FSS\_P9, FSS\_P10, and FSS\_P11. The proposed fuzzy soft set algorithm is applied to all the combination of parameters (Selection policies). The experimental results showed that all combinations produce better performance compared to all the existing algorithms. A performance analysis is made between the 11 fuzzy soft sets each having a different combination of choice parameters. Some fuzzy soft sets optimize energy consumption whereas others optimize SLAV rate and VM migrations. Hence, the problem is viewed as multi-objective problem and the objectives are conflicting in nature, with multiple objectives; decrease in energy consumption increases SLA violation rate and VM migrations. In MOPs, no single solution is the best. So, multiple non-dominated fuzzy soft sets (Pareto-optimal sets) that are good in all the objectives are determined. Every solution in these sets is an acceptable solution. Pareto-optimal fuzzy soft sets are identified and VMs can be selected based on the combination of parameters in these sets for migration.

The remainder of the paper is structured as follows: Section 2 presents the related works. Section 3 describes the proposed fuzzy soft set based VM selection algorithm. In Section 4, the experimental setup and evaluation metrics are discussed. The observations made from the proposed algorithm is also presented in section 4. In Section 5, the experimental results of existing and proposed methods are discussed. Finally, Section 6 concludes the work with future extension.

## 2. RELATED WORKS

(Beloglazov, A., *et al.*, 2012) recommend a consolidation method that fixes upper and lower thresholds based on CPU utilization. If the use of the CPU utilization crosses the upper threshold or falls below the lower threshold then their method selects some VMs and moves them to other hosts. Their model of power consumption only considers CPU utilization. Since the authors used two thresholds and that are fixed values, the consolidation method is static. This reduces the scalability in various workloads of this approach. But our proposed mechanism tries to solve these issues. Static thresholds are not suitable for dynamic workloads. (Beloglazov, A., *et al.*, 2012) proposed a mechanism for dynamic VM consolidation, which takes the history of resources utilization data to determine the adaptive thresholds of each server. The authors proposed Minimum Migration Time (MMT) and Maximum Correlation (MC) VM selection policies to reduce both energy consumption and SLA violation of cloud data centers. In their mechanism, a VM will be selected based on either one of the selection policies. In this paper, the proposed method takes both the factors at a time for VM selection. Verma *et al.* (Verma, A., *et al.*, 2008) designed a power-aware placement controller, pMapper, for virtualized heterogeneous server clusters. The pMapper uses the following power management strategies: DVFS, server power switching and VM consolidation. The authors used a decreasing first fit and a minimum power packing as VM placement algorithms. But, according to (Beloglazov, A., *et al.*, 2012) DVFS never considers threshold values, and VM migration. Moreover, the decreasing best fit is better than decreasing first fit used by pMapper.

(Mhedheb, Y., *et al.*, 2013) proposed a cloud data centers VM consolidation using load and thermal-aware algorithms. They introduced Thermal-aware Scheduler (ThaS) in order to reduce the high temperature of the system and the energy consumption by servers. ThaS uses DVFS as power management technique which schedules VMs concerning the temperature of CPUs and sends VMs to the hosts that have least temperature and CPU usage. But the authors focused only power consumption. They didn't point out SLA violation. (Taheri, M., *et al.*, 2011) introduced a two-phase VM consolidation mechanism to address the issue of incomplete migrations. In the first phase, VMs from overutilized hosts migrate to other hosts, and in the second phase, VMs from underutilized hosts are sent to other hosts. The Bewilder VMs are the VMs to consolidate, but if there is no space in the destination hosts, the system terminates the migration and brings the VMs to the previous position. This problem results in a waste of CPU capacity, power and increases the overhead of the network. (Lim, M. Y., *et al.*, 2009) proposed a power-aware (PADD) selection for live VM migrations. To reduce energy consumption and SLA violation, PADD uses two buffering levels: a local buffer refers to 10% of the CPU capacity reserved for each server and a global buffer refers to reserved pool of CPU capacity across all servers. PADD reduces the number of VM migrations by using these buffers and reduces SLA violations by migrating the VMs that utilize more than the reserved CPU capacity in local buffer. The authors tried to reduced VM migrations and power consumption, but SLA violation is still high.

(Wang, X., *et al.*, 2013) proposed a distributed live VM migration mechanism in cloud data centers. This mechanism uses workload vectors, which collect workload data from all servers. They used a double-threshold technique to make the decision for VM migrations in terms of CPU utilization. The VMs selected for migration may use a vast amount of CPU capacity which will overload the destination server; this issue is ignored by the proposed mechanism. Their mechanism is however static because the thresholds are considered as fixed. This issue makes this mechanism unsuitable for

different types of workloads. (Pineiro, E., *et al.*, 2001), and (Chase, J.S., *et al.*, 2001), suggested a method for minimizing energy consumption in a heterogeneous computing node cluster. It is done by assigning workloads on minimum physical nodes and by switching to idle nodes. Their method deals with the power/performance compromise, since workload consolidation may result in a deterioration of application performance. SLAs define the performance and execution requirements for applications to ensure reliable QoS. (Monil M.A.H., *et al.*, 2015) proposed a fuzzy VM selection algorithm for overloaded host detection, Mean, Median, Standard Deviation (MMSD), but the MMSD algorithm consumes more power than the threshold-based algorithm. (Monil M.A.H., *et al.*, 2016), and (Monil M.A.H., *et al.*, 2017), suggested five fuzzy variations of the VM selection algorithm. Compared to the threshold-based algorithm, all their variations did not achieve energy efficiency.

The concept of intuitionistic fuzzy soft set is also proposed which is a combination of intuitionistic fuzzy set and soft set. Intuitionistic Fuzzy Soft Set theory (IFSS) (Maji, P.K., *et al.*, 2001) may be more applicable to tackle uncertainty and imprecision. The simultaneous consideration of positive and negative information that is the central idea of intuitionistic fuzzy soft sets in any real-life human discourse and action. The current research is carried out using fuzzy soft set since we didn't find any positive and negative information related to our problem.

Mostly researchers have considered any one of the factors such as either RAM or CPU or Memory or Correlation values for selecting VM for migration. (Zhou, Z., *et al.*, 2018) considered both CPU and memory factors at a time for VM selection. In this paper, a new VM selection strategy is proposed using fuzzy soft set. The proposed strategy takes into consideration all the four factors at the same time, and a combination of the four factors is also implemented. The proposed algorithm accurately finds which VM has to migrate from the overloaded hosts in the cloud data center. The following selection policies and fuzzy soft set are adopted in the proposed algorithm.

## 2.1 Selection Policies

Maximum Ratio of CPU Utilization to Memory Utilization (MRCU) is proposed by (Zhou, Z., *et al.*, 2018). It considers both CPU and memory factors. MRCU chooses VMs from the host that is overloaded by CPU-intensive tasks for migration. In this case, out of the total energy consumption, most of the energy consumed by CPU than other components such as memory. It applies the following formula for VM selection.

Minimum the Product of CPU Utilization and Memory Utilization (MPCU) is also proposed by (Zhou, Z., *et al.*, 2018) for VM selection. When I/O-intensive tasks overload a host, it may increase SLA violation. In this case, both CPU and memory consume large amounts of energy. It equally considers both CPU factor and memory factor. The following condition is applied to choose a VM  $v$  for migration.

Minimum Migration Time (MMT) policy migrates a VM from overloaded host to less loaded host when the VM takes minimum migration time (Beloglazov, A., *et al.*, 2012). The migration time is evaluated as size of RAM used by the VM divided by available network bandwidth of the  $j^{th}$  host.

Maximum Correlation (MC) is proposed by (Beloglazov, A., *et al.*, 2012). The strategy behind this policy is that higher the usage of resources in the server by the applications, greater is the chance of the server getting overloaded. So, the VMs that have higher correlation of the CPU utilization compared with other VMs is selected for migration. The multiple correlation coefficient is applied to evaluate the correlation between CPU utilization by VMs. Multiple correlation coefficient harmonizes with the squared correlation between the predicted and the actual values of the dependent variable.

## 2.2 Fuzzy Soft Set

### 2.2.1. Soft Set

Molodstov initiated the concept of soft theory as a new mathematical tool for dealing with uncertainties. He pointed out the difficulties in theory of fuzzy set, theory of intuitionistic fuzzy set

etc. He (Molodtsov, D., 1999), and (Aktaş, H., *et al.*, 2007) presented various applications of the soft set theory such as Game Theory, Operational Research, Riemann integration, Perron integration, Probability, Measurement Theory, etc.

**Definition 1:** Let  $X$  be a non-empty set. A mapping  $M$  from  $X$  into the unit interval  $[0,1]$  is called a fuzzy subset of  $X$ .

**Definition 2:** Let  $U$  be a universal set,  $E$  be a set of parameters, and  $P(U)$  is the power set of  $U$  and  $A \subset E$ .

**Definition 3:** The pair  $(F, A)$  is a soft set over  $U$  where  $F$  is a mapping, given by:

$$F: A \rightarrow P(U) \tag{1}$$

**Definition 4:** Consider the two soft sets  $(F, A)$  and  $(G, B)$  over the universal set  $U$ , then  $(G, B)$  is called a soft subset of  $(F, A)$  if:

$$B \subseteq A$$

$G(\beta)$  is a subset of  $F(\beta)$  for all  $\beta \in B$

### 2.2.2 Fuzzy Soft Set

**Definition 5:** Let  $U$  be an initial universe set and  $E$  be a set of parameters. Let  $P(U)$  denote the set of all fuzzy sets of  $U$ . Let  $A \subset E$ . A pair  $(F, A)$  is called a fuzzy soft set over  $U$ , where  $F$  is a mapping given by  $F: A \rightarrow P(U)$ .

**Definition 6:** A soft set is represented in the form of a table. In this table the entries are  $h_{ij}$  corresponding to  $h_i \in U$  and parameter  $e_j \in E$  where  $h_{ij}$  = membership value of  $h_i$  in  $F(e_j)$ .

**Definition 7:** Comparison table of a fuzzy soft set  $(F)$  is a square table in which number of rows and number of columns are equal. The rows and columns (Table 2) are labeled by the names  $h_1, h_2, \dots, h_n$  of universe. The entries in the table are  $C_{ij}$ , for all  $i, j = 1, 2, \dots, l$ , given by  $C_{ij}$  = the number of parameters for which the membership value  $h_i$  exceeds or equal to the membership value  $h_j$ . Here,  $C_{ij}$  indicates a numerical value,  $0 \leq C_{ij} \leq k$ , and  $C_{ij} = k$ , where  $k$  is the number of parameters in  $P$ .

$$Ci = \sum_{j=1}^P h_{ij} \tag{2}$$

Table 1. Calculation of row-sum and column-sum

	Formula	Explanation
Row-Sum	$r_i = \sum_{j=1}^n C_{ij}$	$r_i$ indicates the total number of parameters in which $h_i$ dominates all the members of universe $U$ .
Column-Sum	$t_j = \sum_{i=1}^n C_{ij}$	$t_j$ indicates the total number of parameters in which $h_i$ dominates all the members of universe $U$ .
Score	$Si = r_i - t_j$	The score value of $i^{th}$ ovm is indicated $S_i$

### 3. PROPOSED WORK

In this paper, an efficient fuzzy soft set based VM selection algorithm and combinations of choice parameters are proposed. This section deliberates the proposed algorithm. The nomenclature used in this paper are listed in Table 2.

#### 3.1 VM Consolidation

VM Consolidation is used to maintain the balance between energy and QoS. Virtual machine consolidation comprises two basic processes:

1. Migration of few VMs from overloaded hosts to maintain the quality of service in terms of service level agreement (SLA).
2. Migration of all VMs from the idle hosts and littleloaded hosts to minimize energy consumption.

A crucial decision must be made in both situations. An efficient VM consolidation method should minimize energy consumption, SLA violation rate, and maximize energy efficiency. It should also have efficient VM migration, and minimum number of active hosts at a given time. It considers the following four steps:

1. **Host Classification:** All hosts in the data centers are clustered into five groups using K-Means Inter Quartile Range clustering algorithm: overloaded hosts, normally loaded hosts, little loaded hosts, less loaded hosts, and idle hosts.

Table 2. Nomenclature

Symbols	Values
$T_a, T_b, T_c, T_d$	Threshold Values
$CPU\_Util_i$	CPU Utilization of host $i$
$H = \{h_1, h_2, h_3, \dots, h_k\}$	Number of hosts
$D = \{CPU\_Util_1, CPU\_Util_2, \dots, CPU\_Util_n\}$	CPU Utilization at different time
$k$	Number of Clusters
$R_i (1 \leq i \leq k)$	Number of the $i^{th}$ cluster
$MR = \{MR_1, MR_2, MR_3, \dots, MR_k\}$	Set of midrange values
IQR	Interquartile range of the MR
$TQ_3$	Third Quartile of MR
$TQ_1$	First Quartile of MR
$d$	Level of VM Consolidation
$OVM_l$	Set of $l$ VMs in the overloaded host
$C_{VM}^l (i = 1,2,3, \dots OVM_l)$	The CPU Utilization of the VM $l$
$M_{VM}^l (i = 1,2,3, \dots OVM_l)$	The Memory utilization of VM $l$

2. **Shutdown underutilized hosts:** To save energy, some of the physical machines/hosts should be switched off when not in use by eliminating power consumption in the idle state and should be awakened whenever necessary. To optimize energy consumption and avoid violations of the QoS requirements, it is necessary to efficiently determine when and which physical machines should be deactivated to save energy, or reactivated to handle the increase in the demand of resources. In this paper, all VMs from idle are migrated to lessloaded hosts, and all VMs from littleloaded hosts are migrated to normallyloaded hosts. Then all idle and littleloaded hosts are moved to power save mode.
3. **VM Selection:** After detecting overloaded hosts, it is required to select one or more VMs for migration from the full set of VMs in such a way that after migration the overloaded hosts become normal. To select the VMs to be migrated is a crucial decision as it affects the performance of both the types of hosts (i.e. the overloaded hosts from which the VMs are selected and the hosts to which the selected VMs are placed). In this paper fuzzy soft set algorithm is used to select VMs from the overloaded hosts.
4. **VM Placement:** All the selected VMs from overloaded hosts will now be placed onto the lessloaded hosts.

### 3.1.1 VM Consolidation Algorithm

The overall VM consolidation is given in Algorithm1.

Step 1 clusters the hosts into five: Overload\_hosts, Normallyloaded\_hosts, Lessloaded\_hosts, Littleloaded\_hosts, and Idle hosts based on their threshold values. Step 2 gets the Placement Policy. Steps 3 to 8: Select target hosts from Lessloaded\_hosts to place all VMs from idle host. This host selection will be repeated for all idle hosts. Now idle hosts are moved to power save mode. Steps 9

#### Algorithm 1. VM Consolidation

```
Input: Hosts in datacenter, VM list
Output: Select the Target host
1: for each host  $h_i$  in datacenter do
2:   Classification  $\leftarrow$  Host_Classify
3:   if (Idle hosts) then
4:     for each host  $h_i$  in Idle hosts do
5:       for each VM  $v$  in host  $h_i$  do
6:         Target host  $\leftarrow$  Select_host ( $v$ , Lessloaded_hosts, Placement_Policy)
7:         Migrate  $v$  to Target_host
8:         Move host  $h_i$  to power save mode
9:     else if (Littleloaded_hosts) then
10:      for each host  $h_i$  in Littleloaded_hosts do
11:        for each VM  $v$  in host  $h_i$  do
12:          Target host  $\leftarrow$  Select_host ( $v$ , Normallyloaded_hosts, Placement_Policy)
13:          Migrate  $v$  to Target_host
14:          Move host  $h_i$  to power save mode
15:     else (Overloaded_hosts)
16:       for each host  $h_i$  in Overloaded_hosts do
17:         repeat
18:           Select VM_Migrate_list in  $h_i$  using fuzzy soft set
19:           for each VM  $v$  in VM_Migrate_list do
20:             Target_host  $\leftarrow$  Select_host ( $v$ , Lessloaded_hosts, Placement_Policy)
21:             Migrate  $v$  to Target_host
22:         until host  $h_i$  becomes Normallyloaded
```

to 14: Select target hosts from Normallyloaded\_hosts to place all VMs from Littleloaded\_hosts. This host selection will be repeated for all little loaded hosts. Now Littleloaded\_hosts are moved to power save mode. Steps 15 to 22: Select target host from Lessloaded\_hosts for every VM in the overloaded hosts for migration. Step 18: The selection the VM from the overloaded host for migration is based on fuzzy soft set algorithm. This host selection will be repeated for all overloaded hosts.

### 3.1.2 Finding Threshold Values

The four threshold values  $T_a$ ,  $T_b$ ,  $T_c$ , and  $T_d$  (Zhou, Z., et al., 2018) are determined using, K Means clustering Mid-range Inter-quartile range algorithm (KMI). Let  $H = \{h_1, h_2, \dots, h_s\}$  be a set of  $s$  hosts in the cloud data center. Let  $D = \{CPU\_Util_1, CPU\_Util_2, \dots, CPU\_Util_n\}$  be a dataset where  $CPU\_Util_t \in D$  ( $1 \leq t \leq n$ ) is the CPU utilization of a host at time  $t$ . K-Means clustering algorithm partitions data set  $D$  into  $k$  clusters:  $(R_1, R_2, \dots, R_k)$ . For each cluster  $R_i$ , the algorithm gets the mid-range value using Eqn 3:

$$MR_i = \frac{((Max(R_i) + Min(R_i)))}{2} \quad 1 \leq i \leq k \quad (3)$$

where the parameter  $Max(R_i)$  invokes to the maximum value of  $R_i$  cluster and the parameter  $Min(R_i)$  refers to the minimum value of  $R_i$  cluster. It produces the set  $MR = \{MR_1, MR_2, \dots, MR_i \dots MR_k\}$ . Then the algorithm acquires the Inter-quartile Range (IQR) of the set MR using Eqn 4:

$$IQR = TQ3 - TQ1 \quad (4)$$

where the parameter TQ3 is the third quartile range of MR while parameter TQ1 is the first quartile range of MR. The adaptive four thresholds ( $T_a$ ,  $T_b$ ,  $T_c$ , and  $T_d$ ) in AFT are defined as follows:

$$T_a = \frac{3}{10} (1 - d \times IQR) \quad (5)$$

$$T_b = \frac{6}{10} (1 - d \times IQR) \quad (6)$$

$$T_c = \frac{9}{10} (1 - d \times IQR) \quad (7)$$

$$T_d = (1 - d \times IQR) \quad (8)$$

where  $d$  is the parameter that states how rapidly the system consolidates the VMs. For the virtual machines, smaller the value of the parameter  $d$  consumes lesser the energy and higher the SLA violations and vice versa.

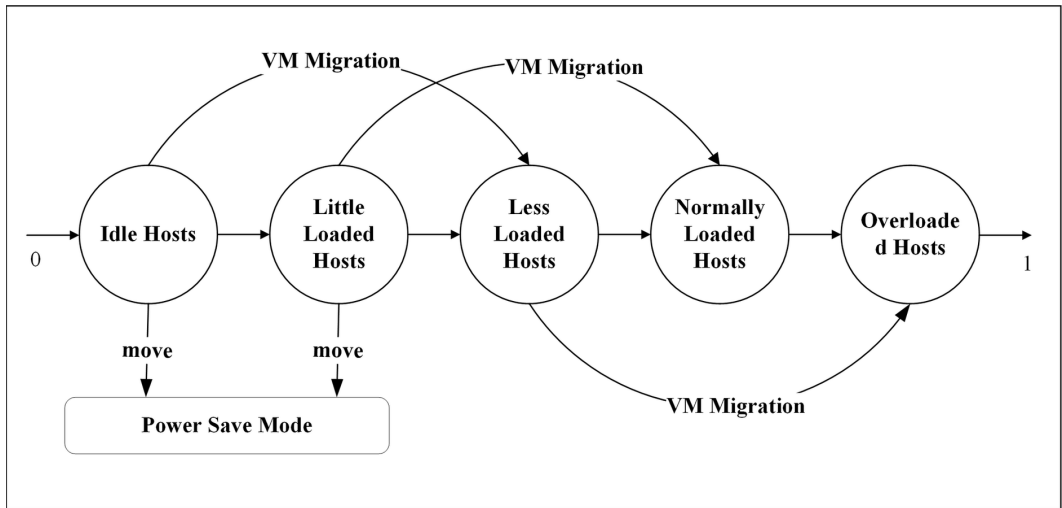
### 3.1.3 Host Classification

An adaptive four threshold method is used to classify the hosts shown in Fig 1. The four threshold values are  $T_a$ ,  $T_b$ ,  $T_c$ , and  $T_d$ .

The following algorithm (Algorithm 2) classifies the hosts into five groups based on CPU utilization: idle hosts between  $(0 - T_a)$ ; less loaded hosts between  $(T_a - T_b)$ ; little loaded hosts between  $(T_b - T_c)$ ; normally loaded hosts between  $(T_c - T_d)$  and overloaded hosts  $(T_d - 1)$ .



Figure 1. Host Classification



Algorithm 2. Host\_Classify

**Input:**

Threshold Values =  $T_a, T_b, T_c,$  and  $T_d$ ;  
 $H$  = Number of hosts;  
 $CPU\_Util_i$  = CPU utilization of  $i^{th}$  host

**Output:** Group of hosts

- 1) for all hosts  $i = 1$  to  $H$  do
- 2)     if  $CPU\_Util_i \geq T_d$  then
- 3)          $host[0] \leftarrow$  Overloaded hosts
- 4)     if  $CPU\_Util_i \geq T_c$  then
- 5)          $host[1] \leftarrow$  Normallyloaded hosts
- 6)     if  $CPU\_Util_i \geq T_b$  then
- 7)          $host[2] \leftarrow$  Lessloaded hosts
- 8)     else if  $CPU\_Util_i \geq T_a$  then
- 9)          $host[3] \leftarrow$  Littleloaded hosts
- 10)    else  $CPU\_Util_i < T_a$
- 11)          $host[4] \leftarrow$  Idle hosts
- 12)    end if
- 13)    end if
- 14)    end if
- 15) end for
- 16) return Group of hosts

### 3.2 Fuzzy Soft Set Based VM Selection Algorithm

The fuzzy logic technique is considered a better way of handling uncertain, imprecise or modelless data in solving control, and decision-making problems. Molodstov showed the application of soft set theory to solve decision-making problems. The Molodstov’s fuzzy soft set based VM selection strategy is proposed in this paper for efficient VM migration. The proposed algorithm considers the

following selection factors: MMT, MRCU, MPCU, and MC values at a time for VM selection. The flow chart of the proposed methodology is shown in Fig 2.

### 3.2.1 Description of Fuzzy Soft Set

Let  $OVH = \{Oh_1, Oh_2, Oh_3, \dots, Oh_n\}$  be a set of overloaded hosts;  $E = \{MMT, MRCU, MPCU, MC\}$  be a set of parameters; Let the universal set  $U = VMlist = \{ovm1, ovm2, \dots, ovm_l\}$  be set of VMs in the overloaded host and  $P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}\}$  be the set of choice of parameters of the set E. The  $(F, P)$  be a pair of the soft set  $p \subseteq E$ . The fuzzy soft set  $(F, E)$  is viewed as a collection of fuzzy approximations as below:

- (F, E) = {MMT of VMs = {ovm<sub>1</sub>/.7, ovm<sub>2</sub>/.9, ..., ovm<sub>l</sub>/.6}
- MRCU of VMs = {ovm<sub>1</sub>/.5, ovm<sub>2</sub>/.7, ..., ovm<sub>l</sub>/.8}
- MPCU of VMs = {ovm<sub>1</sub>/.5, ovm<sub>2</sub>/.7, ..., ovm<sub>l</sub>/.8}
- MC of VMs = {ovm<sub>1</sub>/.5, ovm<sub>2</sub>/.7, ..., ovm<sub>l</sub>/.8}

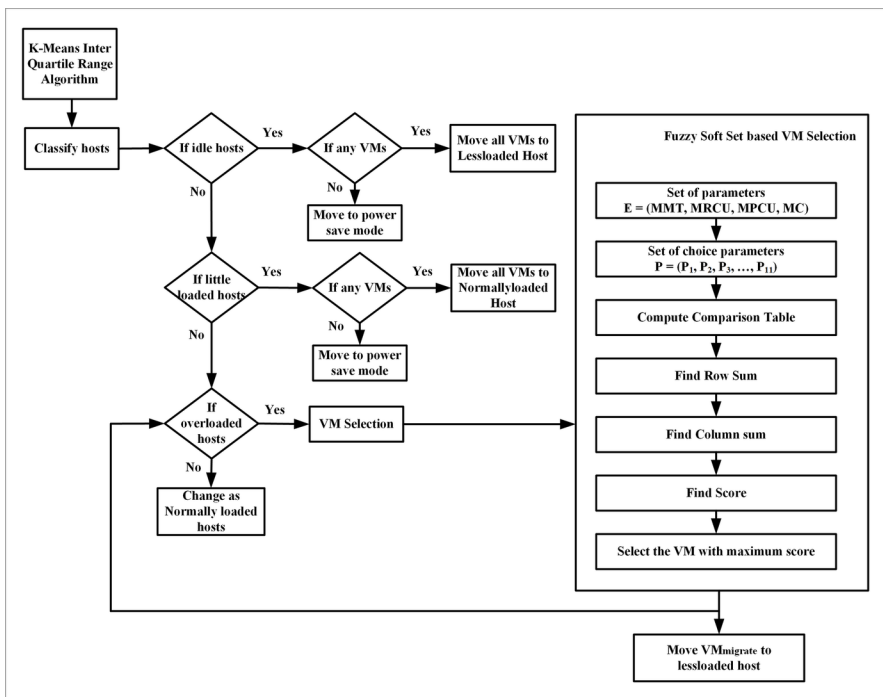
where each approximation has two parts:

- A predicate p.
- An approximate value- fuzzy set  $\gamma$ .

### 3.2.2 Membership Values

The membership values of each parameter of each VM in an overloaded host is calculated using the following formula:

Figure 2. Flow chart of Proposed methodology



$$\text{Membership Value} = \frac{\text{Val} - \text{minVal}}{\text{maxVal} - \text{minVal}} \quad (9)$$

where Val = parameter value of a VM; minVal = minimum parameter value of a VM among all the VMs in the overloaded host; maxVal = maximum parameter value of a VM among all the VMs in the overloaded host.

### 3.2.3 Set of Choice Parameters

The following fuzzy soft sets are considered on the basis of choice parameters for VM migration. The details of the set of choice parameters are shown in Table 3.

### 3.2.4 Tabular Representation

Each fuzzy soft set is represented by a table using definition 6. Each row in the table corresponds to VMs in the overloaded host. Each column corresponds to parameters in the parameter set  $E$ . The entries are  $h_{ij}$  = membership value of  $h_i$  in  $F(e_j)$  where  $h_i \in VM_{list}$  and  $e_j \in E$ .

Suppose there are 5 VMs,  $U = \{ovm_1, ovm_2, ovm_3, ovm_4, ovm_5\}$ . The set of parameters,  $E = \{x_1, x_2, x_3, x_4\}$ . For  $i = 1, 2, 3, 4$ , the set of parameters are “MMT”, “MRCU”, “MPCU”, and “MC”, respectively. Sample representation is given in Table 4.

Table 3. Set of Choice Parameters

Fuzzy Soft Set	Set of Choice Parameters
$(F, P_1)$	$P_1 \subset E$ where $P_1 = \{MMT, MRCU\}$
$(F, P_2)$	$P_2 \subset E$ where $P_2 = \{MMT, MPCU\}$
$(F, P_3)$	$P_3 \subset E$ where $P_3 = \{MMT, MC\}$
$(F, P_4)$	$P_4 \subset E$ where $P_4 = \{MRCU, MPCU\}$
$(F, P_5)$	$P_5 \subset E$ where $P_5 = \{MRCU, MC\}$
$(F, P_6)$	$P_6 \subset E$ where $P_6 = \{MPCU, MC\}$
$(F, P_7)$	$P_7 \subset E$ where $P_7 = \{MMT, MRCU, MPCU\}$
$(F, P_8)$	$P_8 \subset E$ where $P_8 = \{MMT, MRCU, MC\}$
$(F, P_9)$	$P_9 \subset E$ where $P_9 = \{MMT, MPCU, MC\}$
$(F, P_{10})$	$P_{10} \subset E$ where $P_{10} = \{MRCU, MPCU, MC\}$
$(F, P_{11})$	$P_{11} \subset E$ where $P_{11} = \{MMT, MRCU, MPCU, MC\}$

Table 4. Tabular Representation of fuzzy soft set

	MMT	MRCU	MPCU	MC
$ovm_1$	1.0	0.2	1.0	1.0
$ovm_2$	0.4	0.3	0	0.1
$ovm_3$	1.0	1.0	1.0	0.5
$ovm_4$	0.4	1.0	0.2	0.3
$ovm_5$	0.6	1.0	1.0	0.2

### 3.2.5 Comparison Table of a Fuzzy Soft Set

The *comparison* table of each fuzzy soft set is calculated using definition 7. The rows and columns are labelled by the names =  $\{ovm_1, ovm_2, \dots, ovm_l\}$  of the VMs in overloaded host. The entries in the table are  $C_{ij}$  for all  $i, j = 1, 2, \dots, l$  and are given by  $C_{ij}$  = the number of parameters for which the membership value  $h_i$  exceeds or equal to the membership value  $h_j$ . Sample Comparison table is given in Table 5.

### 3.2.6 Calculation of Row-Sum Column-Sum and Score

After that calculating the comparison table and the row sum, column sum and score value are calculated using Table 2. The sample values are shown in Table 6.

### 3.2.7 Algorithm FSS\_VM Selection

The proposed FSS\_VM algorithm consists of two stages. In the first stage, it detects the overloaded host and consolidates VM which is explained in section 3.1. In the second stage, the algorithm selects the VM to be migrated from overloaded host to another host using fuzzy soft set. The selection algorithm is repeated until the host gets normally loaded.

The fuzzy soft set based VM selection algorithm (FSS) is given in Algorithm 3. The algorithm receives the inputs: OVH, VMlist, E, and P. Step 1 get the n overloaded hosts in OVH. Step 2 get the set of p choice parameters in P. Step 3 get the l VMs in the VM<sub>list</sub>. Step 4 select  $j^{th}$  parameters in p. Step 5-7 calculate  $j^{th}$  membership value for l VMs. Step 8 compute the comparison table for l VMs. Step 9-13 find the row-sum, column-sum, and score values are calculated for l VMs in the U. Step 14 find the maximum score value in the V. Step 15-19 if the k value is greater than 1, then select any one virtual machine for migration else return the k value to VM<sub>migrate</sub>. Step 20-22 check whether the host is overloaded or not. If it's overloaded repeat the steps 9-19.

**Table 5. Comparison table**

	$ovm_1$	$ovm_2$	$ovm_3$	$ovm_4$	$ovm_5$
$ovm_1$	4	3	3	3	3
$ovm_2$	1	4	0	1	0
$ovm_3$	3	4	4	4	4
$ovm_4$	1	4	1	4	2
$ovm_5$	2	4	2	3	4

**Table 6. Calculation of Row-sum, Column-sum, Score**

	Row-sum	Column-sum	Score
$ovm_1$	20	11	9
$ovm_2$	7	22	-15
$ovm_3$	23	10	13
$ovm_4$	15	18	-3
$ovm_5$	17	15	2

**Algorithm 3. FSS\_VM Selection ()**

---

**Input:**  $OVH = \{Oh_1, Oh_2, \dots, Oh_n\}$  be set of overloaded hosts;  
 $U = VM_{it} \{ovm_1, ovm_2, \dots, ovm_i\}$  be set of VMs in the overloaded hosts;  
 $E = \{MMT, MRCU, MPCU, MC\}$  be set of parameters;  
 $P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}\}$  be set of choice parameters.

**Output:**  $VM_{migrate}$  to be migrated.

```

1: for each  $n$  in  $OVH$  do
2:   for each  $p$  in  $P$  do
3:     for each  $l$  in  $VM_{list}$  do
4:       for each  $j$  in  $p$  do
5:          $C_{lj} \leftarrow j^{th}$  membership value of  $l$ 
6:       end for
7:     end for
8:     Compute comparison table
9:     for each  $l$  in  $VM_{list}$  do
10:      Find row-sum  $r_l$ 
11:      Find column-sum  $t_j$ 
12:      Find Score  $S_l$ 
13:    end for
14:    Find  $S_k = \max(S_l)$ 
15:    if  $(k \geq 1)$  then
16:       $VM_{migrate} \leftarrow$  Select any one VM for migration
17:    Else
18:       $VM_{migrate} \leftarrow$  Select  $S_k$ 
19:    end if
20:    if ( $n^{th}$  host overloaded) then
21:      Repeat the steps 9 - 19
22:    end if
23:  end for
24: end for

```

---

## 4. EXPERIMENTAL SETUP

It is complicated to manage repetition of large-scale experiments on real-time infrastructure which is required to evaluate and compare the new algorithms. For this reason, simulation has been chosen to evaluate the performance of the algorithms.

### 4.1 Cloudsim Toolkit

Cloudsim is selected as a simulation platform for experimental setup (Beloglazov, A., et al., 2012). It is a modern, purely cloud-based simulation platform. SimGrid, GangSim, Cloudsim Plus are some of the other simulation toolkits. Cloudsim toolkit focused on virtualized environments, supporting on-demand provisioning and its management are in the early stages. Now the cloudsim has been extended to energy-aware simulations. The latest version of cloudsim 4.0 is used in this research.

### 4.2 Host Specification

The data center is set up with two types of hosts (physical nodes). Half of the hosts which are HP ProLiant ML 110 G4 and another half are HP ProLiant ML 110 G5. The details of the host configuration are shown in Table 7.

### 4.3 VM Specification

The features of the virtual machines consistent with Amazon EC2. VM types are depicted in Table 8. There are four types of virtual machines like High-CPU Medium instance, Extra-Large instance, Small instance, and Micro instance.

Table 7. Host Configuration

Hosts	CPU type	Frequency	Cores	RAM
		(GHz)		(GB)
HP ProLiant G4	Intel Xeon 3040	1.86	2	4
HP ProLiant G5	Intel Xeon 3075	2.66	2	4

Table 8. VM Configuration

VM Type	CPU (MIPS)	RAM(GB)
High-CPU medium instance	2500	0.85
Extra large	2000	3.75
Small instance	1000	1.70
Micro instance	500	0.61

#### 4.4 Workload Data Traces

The PlanetLab, is a monitoring infrastructure for CoMon (Park, K., et al., 2006) project which provides traces of real-world workload data. This data is taken from a cloud network that is distributed geographically in 500 different locations. These real-world traces contain VM *utilization* record for every 5-minute interval. Each VM contains 288 ( $=24*(5/60)$ ) data of CPU utilization. Table 9 presents the data of 10 days of 2011 have been used in this experiment.

#### 4.5 Energy Consumption Model

Generally, CPU utilization, memory, disk storage, and cooling system are used to represent the energy consumption of the data center. A non-profit corporation called Standard Performance Evaluation Corporation (SPEC) is formed to establish, maintain, and endorse standardized benchmarks and tools for evaluating performance and energy efficiency for the newest generation of computing systems (Hanson, H., et al., 2007). In this paper, SPEC benchmark energy consumption data is used at different

Table 9. Workload Traces (CPU Utilization)

Date	Number of VMs
03/03/2011	1052
06/03/2011	898
09/03/2011	1061
22/03/2011	1516
25/03/2011	1078
03/04/2011	1463
09/04/2011	1358
11/04/2011	1233
12/04/2011	1054
20/04/2011	1033

load levels. Two types of hosts HP ProLiant ML110 G4 and HP ProLiant ML110 G5 are used for this experiment. The different load levels of energy consumption in these hosts are shown in Table 10.

#### 4.6 Datasets

In this experiment, two types of datasets are used. They are described below.

##### 4.6.1 Dataset 1

The workload dataset 03/03/2011 is considered, Number of VMs are 1052, and Number of hosts are 800, half of which is HP ProLiant ML110 G4, and another half is HL ProLiant ML110 G5. Table 7 lists the specifications of the hosts. The characteristics of the dataset is given in Table 9 in section 4.4.

##### 4.6.2 Dataset 2

The workload dataset 23/03/2011 is considered, Number of VMs are 1516, and Number of hosts are 400, half of which is HP ProLiant ML110 G4, and another half is HL ProLiant ML110 G5. In the Table 7 frequency of the hosts is changed as 2.72 (GHz) for HP ProLiant ML110 G4 and 5.32 (GHz) for HP ProLiant ML110 G5. The characteristics of the dataset are given in Table 9 in section 4.4.

#### 4.7 Evaluation Metrics

The following evaluation metrics are considered to compare the efficiency of proposed and existing methods and are shown in Table 11.

### 5. PERFORMANCE ANALYSIS

The proposed algorithm is compared with the following existing algorithms such as None Power Aware (NPA) (Calheiros, R. N., et al., 2011), DVFS (Hanson, H., et al., 2007), THR-MMT-1.0 (Beloglazov, A., et al., 2012), THR-MMT-0.8 (Beloglazov, A., et al., 2012), MAD-MMT-2.5 (Beloglazov, A., et al., 2012), IQR-MMT-1.5 (Beloglazov, A., et al., 2012), LRR-MMT-1.2 (Beloglazov, A., et al., 2012), LR-MMT-1.2 (Beloglazov, A., et al., 2012), KAM-MMT-1.2 (Zhou, Z., et al., 2016), KAM-MMS-2.0 (Zhou, Z., et al., 2016), KAI-MMS-1.0 (Zhou, Z., et al., 2016), KMI-MPCU-2.0 (Zhou, Z., et al., 2018), KMI-MRCU-1.0 (Zhou, Z., et al., 2018). Before comparing the performance of these energy-aware algorithms, parameter  $d$  is determined for AFT to process the CPU instance tasks. The

Table 10. Host Energy Consumption in Different Workload Levels

Server	HP G4	HP G5
0%	86	93.7
10%	89.4	97
20%	92.6	101
30%	96	105
40%	99.5	110
50%	102	116
60%	106	121
70%	108	125
80%	112	129
90%	114	133
100%	117	135

Table 11. Evaluation Metrics

Evaluation Metrics	Description	Formula
<b>SVTAH</b>	SVTAH decides which active host has reached the 100% CPU utilization during the time.	$SVTAH = \frac{1}{M} \sum_{i=1}^M \frac{T_{pi}}{T_{qi}}$
<b>PDCVM</b>	The overall performance will be degraded due to the VM migrations most of the time.	$PDCVM = \frac{1}{N} \sum_{i=1}^N \frac{P_{dj}}{P_{ri}}$
<b>SLAV</b>	Both the SLA violation rate metrics are equally used to measure the SLA violation independently. It is obtained by multiplying SVTAH and PDCVM	$SLAV = SVTAH \times PDCVM$
<b>ESV</b>	ESV is a joined measurement that includes both energy consumption and rate of SLA violation.	$ESV = EC \times SLAV$
<b>Energy Consumption</b>	It calculates the total energy consumed on the physical hosts of a data center by application workloads.	It is calculated on the basis of the model given in section 4.5.
<b>Energy Efficiency</b>	Energy Efficiency (EE) can be incorporated into forms of Energy Consumption and SLA Violation Rate. Where $P_c$ is Energy consumption. It is formed as	$EE = \frac{1}{P_c \times SLAV}$
<b>Improvement Rate</b>	The percentage improvement of the proposed algorithm is computed.	$\varphi = \left( 1 - \frac{Proposed\ Method}{Existing\ Method} \right) \times 100$

combination of AFT, KMI, FS, and parameter d is denoted as AFT-FS-d. In this experiment, the value of parameter d for AFT-FS-d varies from 0.6 to 1.0, which is an increase by 0.14. For,  $d < 0.6$ , there is no CPU utilization of VMs and for,  $d > 1$ , there is no variations in the objectives. Hence the value of d is considered between 0.6 and 1.0.

Several metrics are used to assess the performance of the algorithms. One of the metrics is the total energy consumption caused by the application workloads by the physical servers of a data center. The metrics SLA Violation Rate, SVTAH and PDCVM defined in Table 11 are used to assess the level of SLA violations caused by the system. Another metric is the number of VM migrations that the VM manager initiates during the VM placement adaptation. The main metrics are energy consumption by physical nodes and SLA Violation, but these metrics are conflict in nature as the cost of the increased level of SLA violations can usually decrease the energy. The resource management system’s objective is to minimize both energy and SLA Violations.

### 5.1 Dataset 1

The dataset1 consists of 800 heterogeneous hosts, among which half are HP ProLiant ML110 G4 and the other half are HP ProLiant ML 110 G5. The host configuration details are given in Table 7. four kinds of VMs are considered, such as High-CPU, Extra-large, small instance and micro instance. For this experiment, real-time workload data traces are used as part of PlanetLab’s CoMon project monitoring infrastructure (Beloglazov, A., et al., 2012). The CPU utilization workload data was taken from more than 500 locations around the world. In this experiment, 1052 VMs are chosen from “03/March/2011” dataset in workload traces.



Table 12 illustrates the energy efficiency, ESV, energy consumption, SLA violations, SVTAH, PDCVM, and number of VM migrations obtained by the proposed Fuzzy Soft Set (FSS) based VM Selection algorithm and the other existing algorithms for the given dataset1. The proposed algorithm is tested for all 11 combination of choice parameters and also compared with existing algorithms. In terms of energy efficiency, higher the value is considered as best. For energy consumption, SLA Violation, SVTAH, and PDCVM lesser value is considered as best. For multiple VM migrations, excessive or too little VM migration is not good for energy efficiency.

NPA does not take any measure during processing tasks so it consumes 2419.2 Kwh. DVFS technique reduces the energy consumption to 613.6 Kwh. Compared to NFA, DVFS is effective. However, both NFA and DVFS do not involve in VM migration. So, the notation “-” to express the non-existence energy efficiency, SLA violation, SVTAH, PDCVM, and number of VM migrations is used.

The proposed Fuzzy Soft Set based VM Selection algorithm reduces energy consumption by more than 5 times than all the other existing algorithms and the results are shown in Fig 3. Compared with existing algorithms the fuzzy soft set based VM selection algorithm reduces energy consumption by more than 76%. All FSS\_P1, FSS\_P2, FSS\_P3, FSS\_P4, FSS\_P5, FSS\_P6, FSS\_P7, FSS\_P8, FSS\_P9, FSS\_P10, and FSS\_P11, outperforms NPA, DVFS, THR-MMT-1.0, THR-MMT-0.8, MAD-MMT-2.5, IQR-MMT-1.5, LRR-MMT-1.2, LR-MMT-1.2, KAM-MMS-2.0, KAI-MMS-1.0, KMI-MPCU-2.0 and KMI-MRCU-1.0. Some of the proposed fuzzy soft set based VM selection algorithm such as FSS\_P2, FSS\_P6, FSS\_P8, FSS\_P9, and FSS\_P11 gives much better result in energy consumption.

The minimum amount of energy consumed by the proposed method is 20.66 kWh whereas the minimum consumption by all the other existing methods is 67.55 kWh. Therefore, it is evident that fuzzy soft set based VM selection algorithm enables a 76.46% reduction in energy consumption.

The SLA Violation Rate is a product of SVTAH and PDCVM. The proposed Fuzzy soft set based VM selection algorithm reduces SLA Violation rate by more than 7 times than all the other existing algorithms as it has obtained minimum SVTAH and minimum PDCVM. Compared with existing algorithms the fuzzy soft set based VM selection algorithm reduces SLA violation rate by more than 95%.

Fig 4. Shows that the SLA Violation Rate in Fuzzy soft set based VM selection algorithm outperforms THR-MMT-1.0, THR-MMT-0.8, MAD-MMT-2.5, IQR-MMT-1.5, LRR-MMT-1.2, LR-MMT-1.2, KAM-MMS-2.0, KAI-MMS-1.0, KMI-MPCU-2.0 and KMI-MRCU-1.0.

Minimum SLA violation obtained by the proposed method is  $9.75 \times 10^{-4}$  whereas the minimum of all existing methods is  $2.61 \times 10^{-4}$ . Therefore, the fuzzy soft set based VM selection algorithm has got 99.92% reduction in SLA violation.

Energy-SLA Violation (ESV) is a product of EC and SLAV. If the ESV is minimum, maximum energy efficiency can be obtained. The fuzzy soft set based VM selection algorithm obtained minimum ESV than others. Fig 5. Illustrates the energy efficiency of all energy-aware algorithms that are calculated based on the Table 11. Compared with existing algorithms the fuzzy soft set based VM selection algorithm improves energy efficiency by more than 5 times. The maximum efficiency obtained by proposed fuzzy soft set based VM selection algorithm is 46677.72, and all other algorithms is 16522.75. So, the fuzzy soft set based VM selection algorithm is the most energy efficient for VM selection.

Efficient VM consolidation is also based on less number of VM migrations. Number of VM migrations that took place for each algorithm is given in Fig 6. Minimum number of VM migration is caused by the fuzzy soft set based VM selection algorithm 2336 when compared to all the existing algorithms. The fuzzy soft set based VM selection algorithm resulted in 88% reduction in migration than the other existing algorithms and has given competitive results compared to AFT-FS (Baskaran, N., et al., 2018).

Table 12. Comparison of VM Selection Algorithms for 800 Hosts

POLICY	ENERGY EFFICIENCY ( $\times 10^3$ )	ESV	ENERGY CONSUMPTION (kWh)	SLA ( $\times 10^{-4}$ )	PDCVM (%)	SVTAH (%)	NUMBER OF MIGRATIONS
NPA	0.00	0	2419.2	0.00000	0	0	0
DVFS	0.00	0	613.6	0.00000	0	0	0
THR-MMT-1.0 (Beloglazov, A., <i>et al.</i> , 2012)	37.10	2.70E-02	99.95	2.69700	0.2697	0.001	19852
THR-MMT-0.8 (Beloglazov, A., <i>et al.</i> , 2012)	167.84	5.96E-03	119.4	0.49900	0.0499	0.001	26567
MAD-MMT-2.5 (Beloglazov, A., <i>et al.</i> , 2012)	167.01	5.99E-03	114.27	0.52400	0.0524	0.001	25923
IQR-MMT-1.5 (Beloglazov, A., <i>et al.</i> , 2012)	168.13	5.95E-03	117.08	0.50800	0.0508	0.001	26420
LRR-MMT-1.2 (Beloglazov, A., <i>et al.</i> , 2012)	408.21	2.45E-03	87.93	0.27860	0.0398	0.0007	12820
LR-MMT-1.2 (Beloglazov, A., <i>et al.</i> , 2012)	520.74	1.92E-03	88.17	0.21780	0.0363	0.0006	11850
KAM-MMS-2.0 (Zhou, Z., <i>et al.</i> , 2016)	6936.69	1.44E-04	83.33	0.01730	0.0173	0.0001	6808
KAI-MMS-1.0 (Zhou, Z., <i>et al.</i> , 2016)	4723.92	2.12E-04	104.28	0.02030	0.0203	0.0001	7519
KMI-MPCU-2.0 (Zhou, Z., <i>et al.</i> , 2018)	5104.78	1.96E-04	67.55	0.02900	0.029	0.0001	12607
KMI-MRCU-1.0 (Zhou, Z., <i>et al.</i> , 2018)	31648.37	3.16E-05	87.77	0.00360	0.009	0.00004	2821
AFT-FS-0.6 (Baskaran, N., <i>et al.</i> , 2018)	16674.75	6.00E-06	21.38	0.02805	0.00051	0.00055	3602
FSS_P1	26854.71	3.72E-05	20.83	0.01788	0.000258	0.006929	2378
FSS_P2	28027.02	3.57E-05	20.6	0.01732	0.000248	0.006984	2336
FSS_P3	30359.98	3.29E-05	20.94	0.01573	0.000225	0.006991	2375
FSS_P4	46677.72	2.14E-05	21.97	0.00975	0.000125	0.007801	2499
FSS_P5	26854.71	3.72E-05	20.83	0.01788	0.000258	0.006929	2378
FSS_P6	28027.02	3.57E-05	20.6	0.01732	0.000248	0.006984	2336
FSS_P7	39709.32	2.52E-05	20.68	0.01218	0.000173	0.007039	2374
FSS_P8	28027.02	3.57E-05	20.6	0.01732	0.000248	0.006984	2336
FSS_P9	28027.02	3.57E-05	20.6	0.01732	0.000248	0.006984	2336
FSS_P10	38091.16	2.63E-05	20.94	0.01254	0.000179	0.007004	2421
FSS_P11	28027.02	3.57E-05	20.6	0.01732	0.000248	0.006984	2336

## 5.2 Dataset 2

The dataset2 consists of 400 heterogeneous hosts. The host configuration details are shown as dataset 2 in section 4.6.2. Four kinds of VMs are considered - like High-CPU, Extra-large, small instance and micro instance. In this experiment, 1516 VMs are chosen from “22/March/2011” dataset in workload traces.

Figure 3. Energy Consumption using 800 Hosts

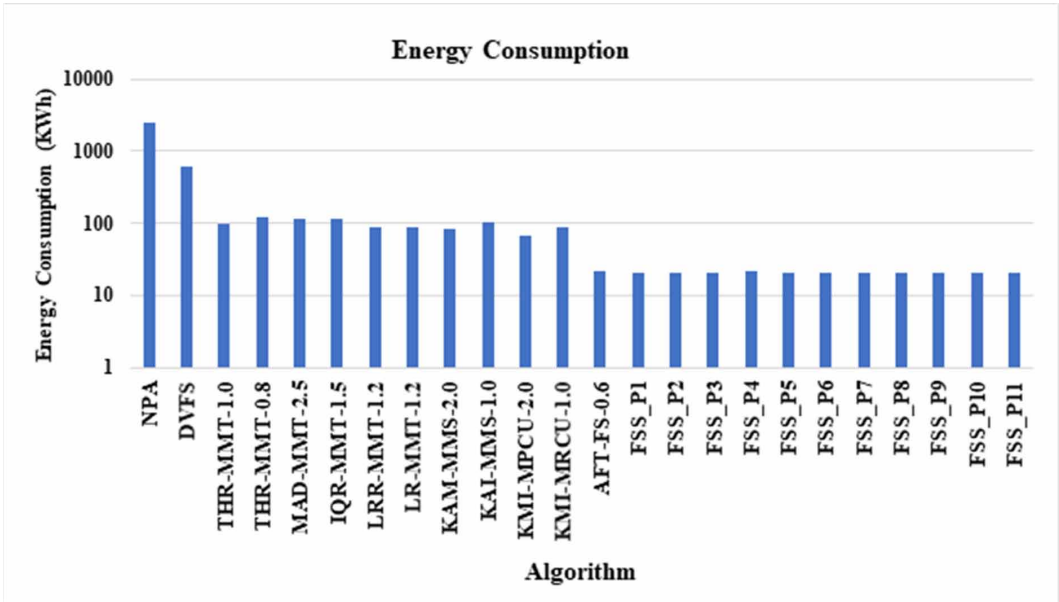


Figure 4. SLA Violation Rate using 800 Hosts

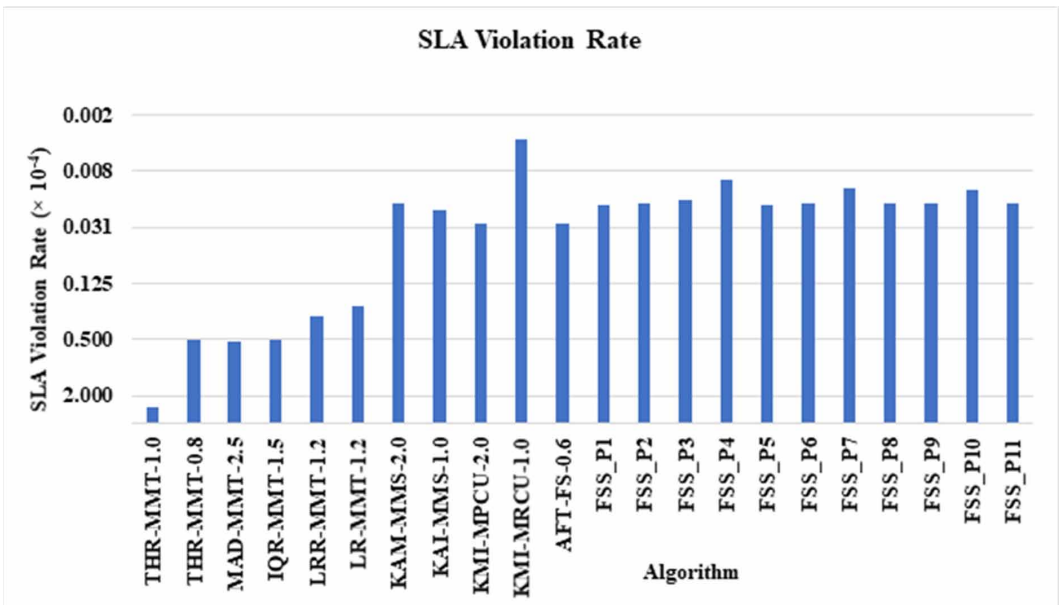


Table 13 illustrates the energy efficiency, ESV, energy consumption, SLA violations, SVTAH, PDCVM, and number of VM migrations obtained by proposed Fuzzy Soft Set (FSS) based VM Selection algorithm for 400 hosts and existing algorithms such as THR-MMT-0.8, KMI-MPCU-2.0, KMI-MRCU-1.0, IQR-MC-1.5, and AFT-FS-0.6.

Figure 5. Energy Efficiency using 800 Hosts

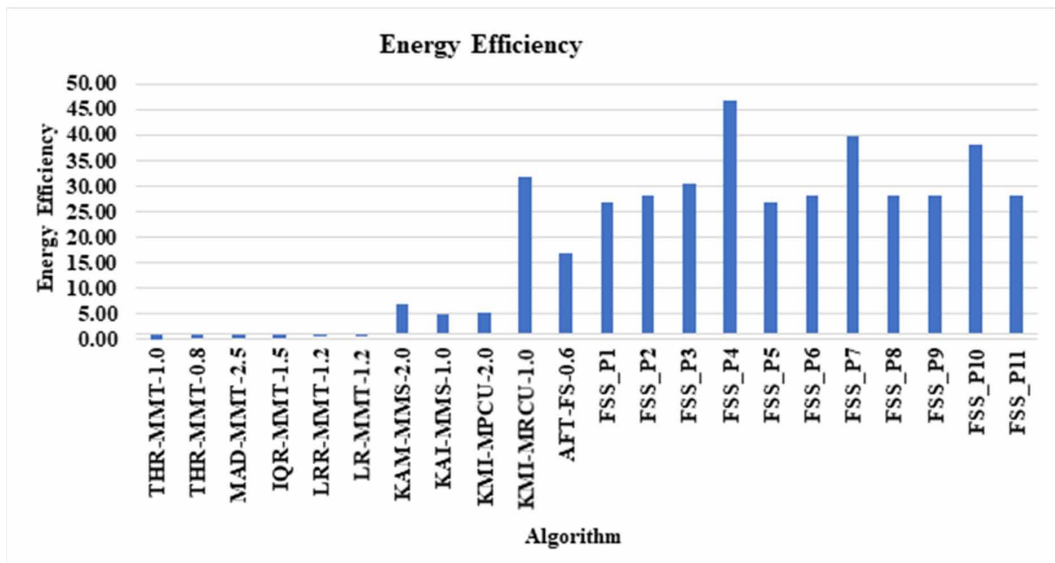
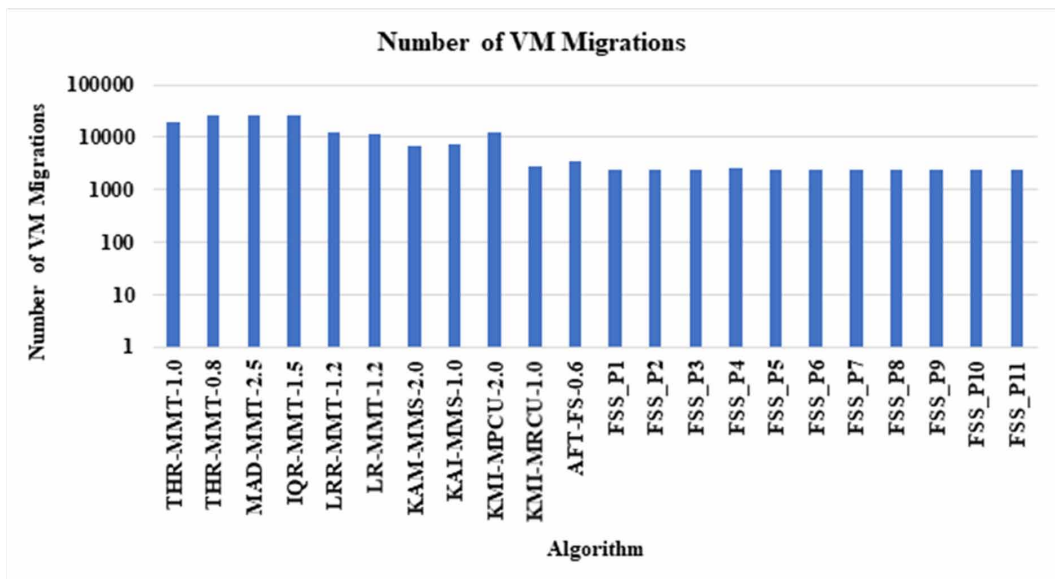


Figure 6. Number of VM Migrations using 800 Hosts

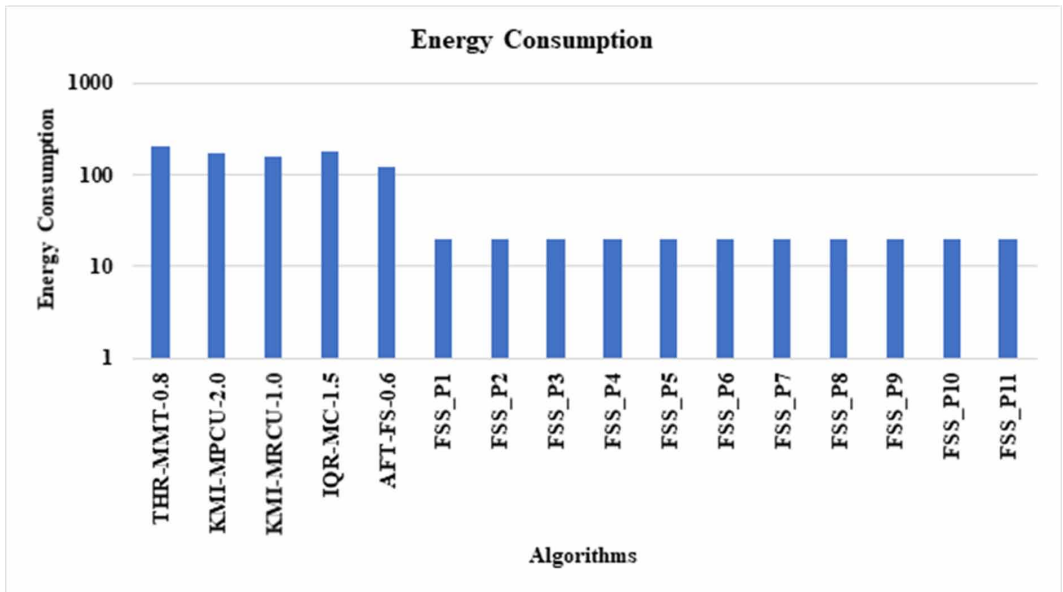


The proposed Fuzzy Soft Set based VM Selection algorithm reduces energy consumption by more than 4 times than all the other existing algorithms and the results are shown in Fig 7. Compared with existing algorithms the fuzzy soft set based VM selection algorithm reduces energy consumption by more than 89%. FSS\_P4 outperforms THR-MMT-1.0, KMI-MPCU-2.0, KMI-MRCU-1.0, IQR-MC-1.5 and AFT-FS-0.6.

Table 13. Comparison of VM Selection Algorithms for 400 Hosts

POLICY	ENERGY EFFICIENCY ( $\times 10^3$ )	ESV	SLA ( $\times 10^{-4}$ )	ENERGY CONSUMPTION	PDCVM (%)	SVTAH (%)	NUMBER OF MIGRATIONS
THR-MMT-0.8	161.624	6.19E-03	3.07E-01	201.8	0.0006	0.0511	29165
KMI-MPCU-2.0	85.865	1.16E-02	0.689	169.03	0.001	0.0689	23035
KMI-MRCU-1.0	8.714	1.15E+00	7.35E+01	156.07	0.112115882	0.06558	5131
IQR-MC-1.5	4.463	2.24E-01	1.27E+01	177.1	0.195358462	0.00648	5048
AFT-FS-0.6	8.812	1.13E-01	9.45E+00	120.09	0.148497106	0.00636	3891
FSS_P1	23828.401	4.20E-05	2.11E-02	19.91	0.000274	0.00769	2322
FSS_P2	41840.082	2.39E-05	1.23E-02	19.47	0.00016	0.00767	2257
FSS_P3	32896.569	3.04E-05	1.54E-02	19.76	0.000199	0.00773	2302
FSS_P4	41495.496	2.41E-05	1.24E-02	19.46	0.000161	0.00769	2266
FSS_P5	23828.401	4.20E-05	2.11E-02	19.91	0.000274	0.00769	2322
FSS_P6	41840.082	2.39E-05	1.23E-02	19.47	0.00016	0.00767	2257
FSS_P7	23581.903	4.24E-05	2.15E-02	19.70	0.00028	0.00769	2339
FSS_P8	28801.475	3.47E-05	1.78E-02	19.52	0.00023	0.00773	2321
FSS_P9	28801.475	3.47E-05	1.78E-02	19.52	0.00023	0.00773	2321
FSS_P10	23581.903	4.24E-05	2.15E-02	19.70	0.00028	0.00769	2339
FSS_P11	28801.475	3.47E-05	1.78E-02	19.52	0.00023	0.00773	2321

Figure 7. Energy Consumption using 400 Hosts



The minimum amount of energy consumed by the proposed method is 19.46 kWh whereas the minimum consumption by all the other existing methods is 120.09 kWh. Therefore, it is evident that fuzzy soft set based VM selection algorithm enables an 89.01% reduction in energy consumption.

The SLA Violation Rate is a product of SVTAH and PDCVM. The proposed Fuzzy soft set based VM selection algorithm reduces SLA Violation rate by more than 7 times compared to all the other existing algorithms as it obtained minimum SVTAH and minimum PDCVM. Compared with existing algorithms the fuzzy soft set based VM selection algorithm reduces SLA violation rate by more than 96%.

In terms of SLA Violation Rate, Fuzzy soft set based VM selection algorithm outperforms THR-MMT-1.0, KMI-MPCU-2.0, KMI-MRCU-1.0, IQR-MMT-1.5, and AFT-FS-0.622. This is shown in Fig 8.

Minimum SLA violation obtained by the proposed method  $1.23 \times 10^{-2}$  whereas the minimum of all existing methods is  $6.19 \times 10^{-3}$ . Therefore, the fuzzy soft set based VM selection algorithm got 96% to reduction in SLA violation.

The fuzzy soft set based VM selection algorithm obtained minimum ESV than others. The energy efficiency of all energy-aware algorithms are calculated based on the Table 11. Fig 9. exhibits the comparison with existing algorithms and the fuzzy soft set based VM selection algorithm improves energy efficiency by more than 5 times. The maximum efficiency obtained by proposed fuzzy soft set based VM selection algorithm is 41840.082 and all other algorithms is 161.624 So, the fuzzy soft set based VM selection algorithm is the most energy efficient for VM selection.

Efficient VM consolidation is also based on minimum number of VM migrations. Number of VM migrations that took place for each algorithm is given in Fig 10. Minimum number of VM migration is caused by the fuzzy soft set based VM selection algorithm 2257 when compared to all existing algorithms. The fuzzy soft set based VM selection algorithm resulted in 92%reduction in migration than the other existing algorithms and has given competitive results compared to AFT-FS.

Figure 8. SLA Violation Rate using 400 Hosts

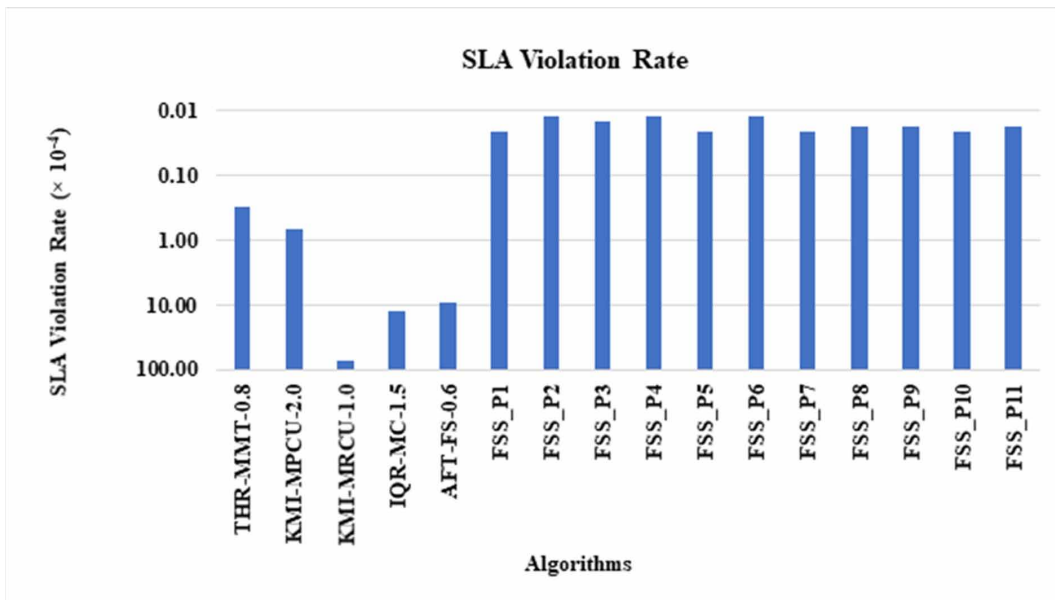


Figure 9. Energy Efficiency using 400 Hosts

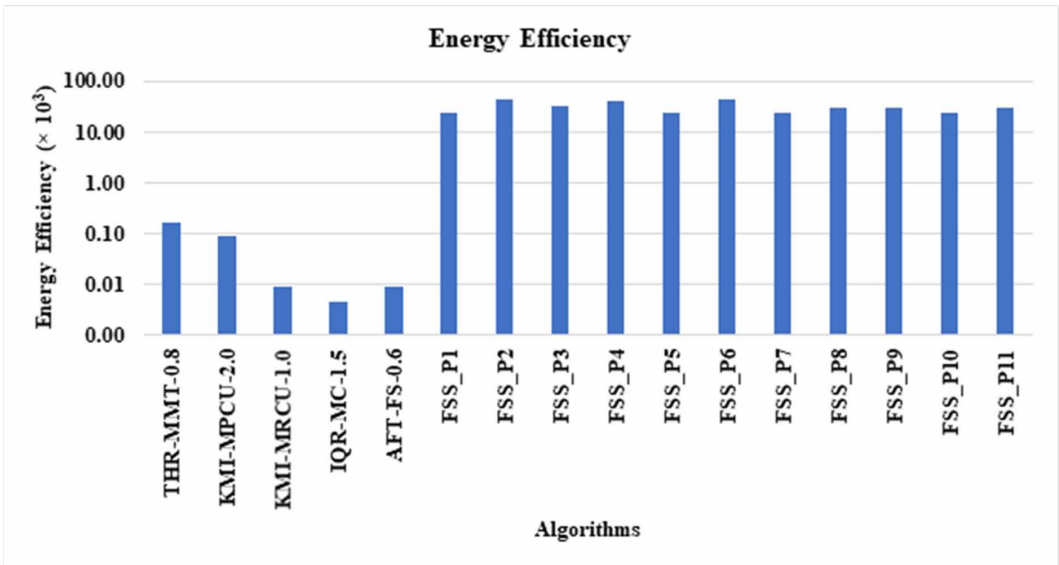
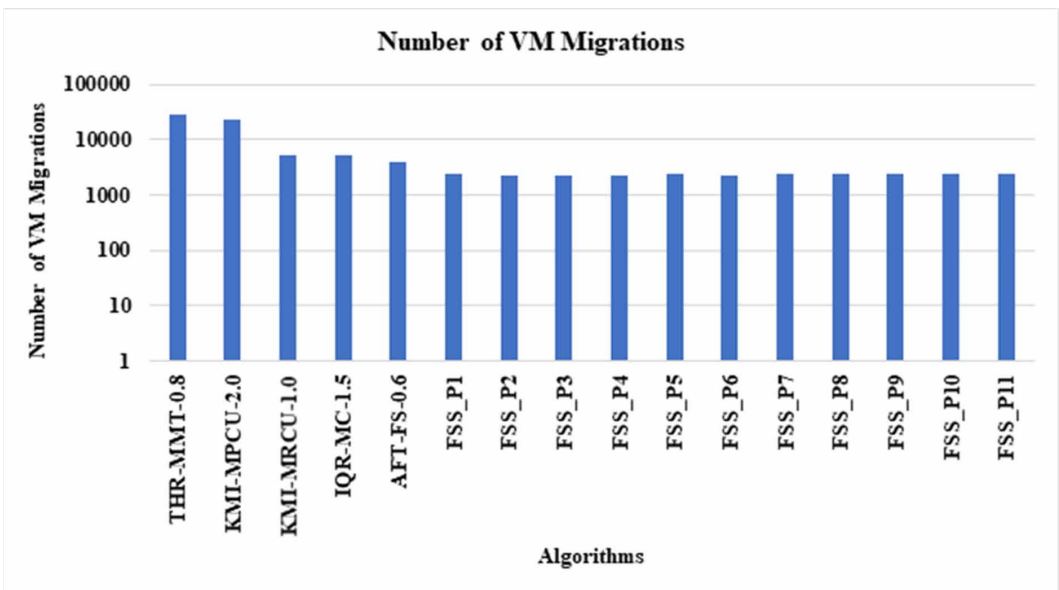


Figure 10. Number of VM Migrations using 400 Hosts



### 5.3 Observations

From the simulation results the following observations are made:

- The four algorithms (THR-MMT-1.0, THR-MMT-0.8, MAD-MMT-2.5, IQR-MMT-1.5) and the two algorithms (KMI-MPCU-2.0, KMI-MRCU-1.0) are based on two threshold and three threshold methods respectively, while algorithm FSS\_VM selection is based on four thresholds.

In our previous work (Baskaran, N., et al., 2018), it is identified that the adaptive four threshold algorithm is more effective than two and three threshold algorithms.

- The four algorithms (THR-MMT-1.0, THR-MMT-0.8, MAD-MMT-2.5, IQR-MMT-1.5) consider the objective of minimizing the energy consumption during the VM placement, while the algorithm AFT-FS considers maximum energy efficiency in addition to energy consumption. From the experimental results it is proved that the latter has better performance than the formers.
- During the VM selection, the six algorithms (THR-MMT-1.0, THR-MMT-0.8, MAD-MMT-2.5, IQR-MMT-1.5, KMI-MPCU-2.0, KMI-MRCU-1.0) do not consider the correlation whereas AFT-FS and FSS\_VM selection algorithm considers CPU, memory utilization and correlation of CPU utilization between the VMs.
- The algorithms (MAD, LR, IQR) consider any one of the factors to select VM for migration. The algorithms MRCU and MPCU consider two factors such as the CPU and the memory utilization for VM selection; whereas the proposed method FSS\_VM selection considers four factors such as RAM usage, CPU utilization, memory utilization, and maximum correlation to select VM for migration.
- The proposed fuzzy soft set algorithm is applied to all the combination of parameters FSS\_P<sub>1</sub>, FSS\_P<sub>2</sub>, FSS\_P<sub>3</sub>, FSS\_P<sub>4</sub>, FSS\_P<sub>5</sub>, FSS\_P<sub>6</sub>, FSS\_P<sub>7</sub>, FSS\_P<sub>8</sub>, FSS\_P<sub>9</sub>, FSS\_P<sub>10</sub>, and FSS\_P<sub>11</sub>, (Selection policies). The combinations are given in section 3.3.2. The experimental results showed that all combinations produce better performance compared to all the existing algorithms.
- The evaluation metrics are tested for all the 11 fuzzy soft sets. In this section the performance of all sets are compared.

### 5.3.1 Non-Dominated Sorting

Some fuzzy soft sets are good in the objective energy consumption, some are good in producing minimum SLA Violation rate, and some are good in minimizing VM migrations objective. For example, among the choice parameters FSS\_P<sub>2</sub>, FSS\_P<sub>4</sub>, FSS\_P<sub>8</sub>, FSS\_P<sub>9</sub>, and FSS\_P<sub>11</sub> optimize energy consumption, FSS\_P<sub>4</sub> minimizes SLA violation rate, and FSS\_P<sub>2</sub>, FSS\_P<sub>6</sub>, FSS\_P<sub>8</sub>, FSS\_P<sub>9</sub>, and FSS\_P<sub>11</sub> reduce number of VM migrations for dataset 1(800 hosts, 1052 VMs). Similarly, FSS\_P<sub>4</sub> optimize energy consumption, FSS\_P<sub>2</sub>, FSS\_P<sub>6</sub>, minimizes SLA violation rate, and FSS\_P<sub>2</sub>, FSS\_P<sub>6</sub>, reduce number of VM migrations for dataset 2(400 hosts, 1516 VMs). Since the objectives are conflicting in nature, decreasing energy consumption increases SLA violation rate and VM migrations. Hence, the problem is viewed as multi-objective problem (MOP). In MOPs, no single solution is the best. Instead multiple solutions need to be considered.

The aim is to determine a set of non-dominated solutions, i.e. a set of non-dominated fuzzy soft set solutions which optimize all the objectives. Every solution in the set is an acceptable solution. In MOP, a solution  $x_1$  is said to dominate the other solution  $x_2$  if the following two conditions are satisfied:

1. The solution  $x_1$  is no worse than  $x_2$  in all objectives:

$$\forall i \in \{1, 2, 3, \dots, K_{obj}\} \rightarrow f_i \leq f_i(x_2) \quad (10)$$

2. The solution  $x_1$  is strictly better than  $x_2$  in at least one objective:

$$\exists j \in \{1, 2, 3, \dots, K_{obj}\} \rightarrow f_j < f_j(x_2) \quad (11)$$

If anyone of the above conditions is violated, the solution  $x_1$  does not dominate the solution  $x_2$ . If  $x_1$  dominated  $x_2$  then the solution  $x_1$  is called a non-dominated solution.



### 5.3.2 Pareto Optimal Solutions

The solutions that are non-dominated within the entire search space are known as Pareto optimal solution and constitute the pareto optimal set. Based on the pareto definition the following fuzzy soft sets are identified as non-dominated sets. Such as FSS\_P<sub>2</sub>, FSS\_P<sub>4</sub>, FSS\_P<sub>6</sub>, FSS\_P<sub>7</sub>, FSS\_P<sub>8</sub>, FSS\_P<sub>9</sub>, and FSS\_P<sub>11</sub> for dataset1 and FSS\_P<sub>2</sub>, FSS\_P<sub>4</sub>, and FSS\_P<sub>6</sub> for dataset2. The identified non-dominated fuzzy soft sets that the VMs can be selected based convey on the combination of parameters in these sets for migration. Through these sets one can achieve better performance.

## 6. CONCLUSION

In this paper, a fuzzy soft set based VM selection (FSS\_VM selection) algorithm is proposed for efficient selection of VMs for migration in cloud data centers. The proposed FSS\_VM selection method first classifies the host based on four thresholds, identifies the overloaded hosts, consolidates the underloaded hosts for maximizing energy efficiency, and then applies a fuzzy soft set method to identify the VM to be migrated to another host in order to make the host a normally loaded host. The proposed method uses all selection policies wherein the factors CPU utilization, memory usage, RAM usage, and correlation values are considered. The proposed method is experimentally tested with the several existing algorithms for two different datasets. The first dataset consists of 800 hosts and 1052 VMs and second dataset consists of 400 hosts and 1516 VMs.

All the VM selection algorithms are compared in terms of Energy Efficiency (EE), Energy Consumption (EC), Service Level Agreement (SLA) violation rate, and VM migrations. The results show that FSS\_VM Selection algorithm outperforms other algorithms in terms of percentage of improvement of energy by (76.46%); SLA violation rate by (99.92%); number of VM migrations (2336) for dataset1; For the dataset2 the percentage of improvement of energy by (89.01%); SLA violation rate by (96%); and number of VM migrations (2257) than other algorithms. Hence FSS\_VM Selection algorithm is the most energy efficient algorithm for VM consolidation.

A performance analysis is made between the 11 fuzzy soft sets each having a different combination of choice parameters. Some fuzzy soft sets optimize energy consumption whereas others optimize SLAV rate and VM migrations. Hence, the problem is viewed as multi-objective problem and the objectives are conflicting in nature, with multiple objectives; decrease in energy consumption increases SLA violation rate and VM migrations. In MOPs, no single solution is the best. So, multiple non-dominated fuzzy soft sets (Pareto-optimal sets) that are good in all the objectives are determined. Every solution in these sets is an acceptable solution. Pareto-optimal fuzzy soft sets are: FSS\_P<sub>2</sub>, FSS\_P<sub>4</sub>, FSS\_P<sub>6</sub>, FSS\_P<sub>7</sub>, FSS\_P<sub>8</sub>, FSS\_P<sub>9</sub>, and FSS\_P<sub>11</sub> for dataset1 and FSS\_P<sub>2</sub>, FSS\_P<sub>4</sub>, and FSS\_P<sub>6</sub> for dataset2. VMs can be selected based on the combination of parameters in these sets for migration.

The VMs selected by the proposed algorithm should be placed to the appropriate hosts during migration. But the proposed system gives such a limitation like more time complete the process. So, it increased the time complexity. The future work will reduce the time complexity and concentrate on developing an efficient VM placement strategy for VM migration since it has the greater impact on improving QoS in VM placement.

## REFERENCES

- Aktaş, H., & Çağman, N. (2007). Soft sets and soft groups. *Information Sciences*, 177(13), 2726–2735. doi:10.1016/j.ins.2006.12.008
- Baskaran, N., & Eswari, R. (2018, July). Adaptive Threshold-Based Algorithm for Multi-objective VM Placement in Cloud Data Centers. In *International Conference on Frontier Computing* (pp. 118-129). Springer.
- Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5), 755–768. doi:10.1016/j.future.2011.04.017
- Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation*, 24(13), 1397–1420. doi:10.1002/cpe.1867
- Brown, R. (2008). Report to congress on server and data center energy efficiency. Public Law, 109–431.
- Buyya, R., Yeo, C. S., & Venugopal, S. (2008, September). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *2008 10th IEEE international conference on high performance computing and communications* (pp. 5-13). IEEE.
- Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software, Practice & Experience*, 41(1), 23–50. doi:10.1002/spe.995
- Chase, J. S., Anderson, D. C., Thakar, P. N., Vahdat, A. M., & Doyle, R. P. (2001). Managing energy and server resources in hosting centers. *Operating Systems Review*, 35(5), 103–116. doi:10.1145/502059.502045
- Hanson, H., Keckler, S. W., Ghiasi, S., Rajamani, K., Rawson, F., & Rubio, J. (2007, August). Thermal response to DVFS: analysis with an Intel Pentium M. In *Proceedings of the 2007 international symposium on Low power electronics and design (ISLPED'07)* (pp. 219-224). IEEE. doi:10.1145/1283780.1283827
- Lim, M. Y., Rawson, F., Bletsch, T., & Freeh, V. W. (2009, June). Padd: Power aware domain distribution. In *2009 29th IEEE International Conference on Distributed Computing Systems* (pp. 239-247). IEEE.
- Maji, P. K., Biswas, R., & Roy, A. R. (2001). Intuitionistic fuzzy soft sets. *Journal of Fuzzy Mathematics*, 9(3), 677-692.
- Mhedheb, Y., Jrad, F., Tao, J., Zhao, J., Kołodziej, J., & Streit, A. (2013, December). Load and thermal-aware VM scheduling on the cloud. In *International Conference on Algorithms and Architectures for Parallel Processing* (pp. 101-114). Springer. doi:10.1007/978-3-319-03859-9\_8
- Molodtsov, D. (1999). Soft set theory—First results. *Computers & Mathematics with Applications* (Oxford, England), 37(4-5), 19–31. doi:10.1016/S0898-1221(99)00056-5
- Monil, M. A. H., & Rahman, R. M. (2015, September). Fuzzy logic based energy aware VM consolidation. In *International Conference on Internet and Distributed Computing Systems* (pp. 31-38). Springer. doi:10.1007/978-3-319-23237-9\_4
- Monil, M. A. H., & Rahman, R. M. (2016). VM consolidation approach based on heuristics, fuzzy logic, and migration control. *Journal of Cloud Computing*, 5(1), 8. doi:10.1186/s13677-016-0059-7
- Monil, M. A. H., & Rahman, R. M. (2017). Fuzzy logic-based VM selection strategy for cloud environment. *International Journal of Cloud Computing*, 6(2), 163–186. doi:10.1504/IJCC.2017.086019
- Park, K., & Pai, V. S. (2006). CoMon: A mostly-scalable monitoring system for PlanetLab. *Operating Systems Review*, 40(1), 65–74. doi:10.1145/1113361.1113374
- Pinheiro, E., Bianchini, R., Carrera, E. V., & Heath, T. (2001). *Load balancing and unbalancing for power and performance in cluster-based systems*. Academic Press.
- Taheri, M. M., & Zamanifar, K. (2011, December). 2-phase optimization method for energy aware scheduling of virtual machines in cloud data centers. In *2011 International Conference for Internet Technology and Secured Transactions* (pp. 525-530). IEEE.

- Verma, A., Ahuja, P., & Neogi, A. (2008, December). pMapper: power and migration cost aware application placement in virtualized systems. In ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing (pp. 243-264). Springer. doi:10.1007/978-3-540-89856-6\_13
- Wang, X., Liu, X., Fan, L., & Jia, X. (2013). A decentralized virtual machine migration approach of data centers for cloud computing. *Mathematical Problems in Engineering*, 2013, 2013. doi:10.1155/2013/878542
- Zhou, Z., Abawajy, J., Chowdhury, M., Hu, Z., Li, K., Cheng, H., Alelaiwi, A. A., & Li, F. (2018). Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*, 86, 836–850. doi:10.1016/j.future.2017.07.048
- Zhou, Z., Hu, Z., & Li, K. (2016). Virtual machine placement algorithm for both energy-awareness and SLA violation reduction in cloud data centers. *Scientific Programming*.

*Nithiya Baskaran is a Research Scholar in the Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamilnadu, India. She received her M.E degree from Anna University, Chennai, India. Her current research interests include Cloud Computing, Evolutionary Computing, Heuristic and Meta-heuristic Approaches, Nature Inspired Algorithms, Fuzzy Logic and Optimization.*

*R. Eswari is working as Assistant Professor in the Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India. She obtained her Bachelor of Engineering in Computer Science and Engineering from Bharathidasan University, Tiruchirappalli and Master of Engineering in Computer and Communication from Anna University, Chennai. She started her research in optimization techniques and awarded Ph.D. in Scheduling algorithms for distributed systems by National Institute of Technology, Tiruchirappalli. She has over 10 years of teaching and research experience and published more than 20 research papers. Her research interests include Computational Intelligence, Optimization, Multi-objective optimization, Cloud Computing and Wireless Sensor Networks.*