Singaporean Journal of Scientific Research

Singaporean Journal of Scientific Research(SJSR)
Journal of Technology and Engineering System(JTES)
Vol.8.No.3 2016,Pp.238-257
available at :www.iaaet.org/sjsr
Paper Received : 28-10-2016
Paper Accepted: 19-11-2016
Paper Reviewed by: 1.Prof. Cheng Yu 2. Dr.M. Akshay Kumar
Editor : Dr. Chu Lio

Virtual Machine Association Method based on Heuristics, Fuzzy logic, and Migration Control

[1]N.Gopinath and [2]K.Anbunathan
[1]Asst.Professor,[2]Assoc. Professor
[1,2]Jeppiaar Maamallan Engineering College
[1,2]Sriperumpudur,Kancheepuram,Tamilnadu, India

Abstract

To come across the increasing request of computational power, at present IT service providers' should choose cloud based services for its flexibility, reliability and scalability. Furthermore datacenters are being built to cater customers' need. However, the datacenters put away large amounts of energy, and this draws negative care. To address those matters, researchers propose energy efficient algorithms that can minimize energy consumption while keeping the quality of service (QoS) at a acceptable level. Virtual machine consolidation is one such technique to safeguard energy-QoS balance. In this research, here explore fuzzy logic and heuristic based virtual machine merging approach to achieve energy-QoS balance. A Fuzzy virtual machine selection method is recommended in this research. It selects virtual machine from an overloaded host. Additionally, here incorporate migration control in Fuzzy virtual machine selection method that will enhance the performance of the selection strategy. A new overload detection algorithm has also been proposed based on mean, median and standard deviation of use of virtual machines. We have used CloudSim toolkit to simulate our experiment and assess the performance of the proposed algorithm on real-world work load traces of Planet lab virtual machines. Simulation results demonstrate that the offered method is most energy efficient compared to others.

Keywords: Cloud, Datacenter, Dynamic virtual machine consolidation, CloudSim toolkit, Planet lab virtual machine data, Fuzzy logic.

1. Introduction

Cloud computing can be classified as a novel era of computing which has transformed the IT industry with its pay-as-you-go services. Its dynamic provisioning of computing services by using Virtual machine (virtual machine) technologies make available opportunity for consolidation and environment separation. Having the viable business view, all the tech-giants have already started providing cloud services. IT companies are now moving from traditional CAPEX model to the OPEX model. To enable and ensure the global development of computing need, cloud service providing companies are now using warehouse sized datacenters to meet user demand which experiences considerable amount of energy. At the start of the cloud computing era, cloud service providers absorbed mainly on catering the computing demand that lead to expansion of cloud infrastructures; hence energy consumption. Therefore, energy consumption by data centers worldwide was risen by 56 % from 2005 to 2010 [4]. In 2010 it was accounted to be between 1.1 and 1.5% of the total energy use and CO_2 emissions of the ICT industry were estimated to be 2% of the global emissions which was equivalent to the emissions of the aviation industry [4]. Additionally, an average size data center consumes as much energy as 25,000 households [1].American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) estimated that infrastructure and energy costs donated about 75%, whereas IT contributed just 25% to the overall cost of working a data center [2] in 2014. So, to cater the increasing need of computing, energy aware technique should be applied in cloud computing infrastructure otherwise the energy need will be enormous and will be threatening to the environment[20]. To handle this problem, datacenter resource needs to be utilized in an efficient manner. An efficient method will not only decrease the energy consumption but also keep the performance up to the mark. Both in hardware and software there are more than a few techniques being used for energy consumption of a cloud system. In hardware level,Dynamic Component Deactivation(DCD and Dynamic Performance Scaling (DPS) are two such techniques, while in virtualization level, several techniques have been proposed e.g., the Energy Management for Hypervisor-based virtual machines and Kernel-based Virtual machine (Kvirtual machine) [28].

Virtual machine Consolidation is one of the techniques which draw researchers' attention and is an active field of research in recent time. As we know that inactive host or host in sleep mode causes minimal energy; therefore, energy consumption can be abridged considerably. By adopting virtual machine consolidation, more energy could be conserved by shutting down underutilized datacenters. However, to achieve this outcome, here it is needed to combine different virtual machines in one server and migrate virtual machines from one host to the other which may lead to SLA (Service Level Agreement) violation. So, algorithms must be designed in such a way that not only reduces power consumption but also serves desired QoS(such as SLA). In a virtual machine consolidation method, selecting the virtual machine to migrate is a challenging job and researchers came up with different solutions. In real world the computation need is very dynamic; therefore, decision is dependent on several criteria. In our research we have applied fuzzy logic. When the overload will be detected, our proposed fuzzy logic and heuristic based algorithm will decide the virtual machine to migrate from the source datacenter to achieve minimum energy consumption by keeping the SLA violation at minimal level.

The remainder of this paper is organized as follows. In section Motivation, motivation has been clarified. In section Proposed method, our proposed methods and algorithms are given. In section Experimental setup, experimental setup is given and in section Experimental result, experimental result and comparisons have been presented. In section Related works, related works are discussed

Finally, in section Conclusion, we have discussed about future work and concluded our paper.

2. Motivation

Virtual machine consolidation algorithm needs to be designed in such a way that there will be minimum energy consumption, minimum violation of SLA, efficient virtual machine migration and minimum number of active hosts in a given time. virtual machine migration causes SLA violation because when a virtual machine is migrated from one host to other it has to transfer its primary memory to the destination host and in the transfer process the requested CPU could not be delivered as the virtual machine will be in a transition state. For this reason, along with power consumption, we need to make sure that the number of virtual machine migration is minimal which will in fact reduce the SLA violation. A desired virtual machine consolidation approach will reduce energy consumption and as well as, it will reduce the negative impact on QoS. To address these issues, virtual machine consolidation has been considered as a bin packing problem in some researches, e.g., [3, 17, 19]. On the other hand, there are researches where virtual machine consolidation has been broken down in separate problems where bin packing solution is considered as one of the sub-problems of virtual machine consolidations, i.e., virtual machine placement [1, 2, 4, 8, 9, 12, 15, 16]. virtual machine consolidation has been broken down in four sub-problems and dealt in researches are the followings:

1. Identify the under loaded datacenter to put them in sleeping mode by migrating all the virtual machines to other active datacenter (Under load detection).
2. Determine the host that is overloaded. Migrate some virtual machines from the identified overloaded datacenter to other datacenters while preserving QoS (Overload detection).
3. Decide the virtual machine(s) that should be migrated (virtual machine selection).
4. Place the selected virtual machines on other active or reactivated hosts\ (virtual machine placement).

Breaking down into sub-problems has two key advantages.(1) Problems get simplified if it is divided into sub-problems and provides the opportunity to break the virtual machine consolidation problem to four problems and devise separate algorithms. By doing that, performance of each algorithm can be measured and analyzed to investigate for identifying the better approach.

As in this research we will mostly focus two sub-problems problems, one is host overload detection and another is virtual machine selection.(2)It enables the option of distributed execution of the algorithms by executing the under load /overload detection and virtual machine selection algorithms. Distributed virtual machine consolidation makes the scaling easier. When a new node is added it automatically gets included in the algorithm which is essential for large-scale Cloud providers. These approaches are designed in CloudSim (an open source Cloud Simulation designed by CLOUDS lab of University of Melbourne[5]). Researchers have developed their algorithms in CloudSim[1,2,4] which can be accessed and used for further research. However, there are places where the improvements could be done to yield better results by saving more power yet delivering the expected QoS. The driving factors which motivate to conduct this research are the following:

- For virtual machine selection, there are several virtual machine selection approaches are proposed in research [1,2,4], namely Maximum Correlation (MC), Minimum Migration Time (MMT), and Random Selection (RS). The maximum correlation (MC) approach selects the virtual machine to migrate which has the highest correlation value among all the virtual machines of a host. The minimum migration time approach (MMT) selects the virtual machine to migrate which has the least memory as it will be migrated faster. And the random selection approach (RS) selects the virtual machine randomly from a host. The approaches offer different results. One method (MC) provides more power savings but lacks in QoS. Another method (MMT) provides better performance KPI, i.e., QoS incurring more power [4]. As the situation is uncertain and in real world the computation need is very dynamic, fuzzy logic can be applied with different inputs to achieve the tradeoff between energy consumption and QoS.

- Migration control can be applied to select the virtual machine to migrate. Refraining from steady resource consuming virtual machine migration can lead to better performance in dynamic virtual machine consolidation [3]. But constantly high resource consuming virtual machine should not be migrated as they consume large number of resources. So, migration control can be applied on two types of virtual machines; steady resource consuming virtual machine and high resources consuming virtual machines. These phenomena can be taken into account while designing a virtual machine selection method.
- To decide whether a host is overloaded or not, there are several algorithms proposed [1,2,4] (e.g., Inter Quartile Range (IQR): which decides the threshold of a host to be marked as overloaded using interquartile range, Median Absolute Deviation (MAD) uses median absolute deviation and THR provides threshold for a host to be marked as overloaded. Local Regression (LR) and Local Robust Regression (LRR) provide prediction of host utilization,). These statistical measures provide a threshold (IQR, MAD and THR) and prediction (LR and LRR) for a host to be identified as overloaded. In parallel, these algorithms rely on mean and standard deviation of resource utilization that gives an indication of future load of a virtual machine which also can be an approach independently for overload detection [15]. However, mean and standard deviation is very much influenced by terminal values. Terminal values indicate the outlier values or the values that are too large or too small and do not represent the normal values. As virtual machine's resource utilization can be very dynamic in real world, instead of mean we can use median and we can modify the formula for standard deviation using median instead of mean. An overload detection algorithm can be designed from this.
- When virtual machine needs to be migrated to another datacenter in virtual machine placement phase or under load detection phase, the destination host needs to be judged whether it will be overloaded in future by using overload detection method.

3. Proposed method

In this work we have designed Fuzzy virtual machine Selection with migration control algorithm and Mean, Median & standard deviation based over load detection algorithm. However, before going in detail, overview of virtual machine consolidation is presented. The algorithm below portrays the basic virtual machine consolidation approach designed in CloudSim.

Algorithm 1 provides a basic flow of virtual machine consolidation. At first the hosts are created. Then the virtual machine data is taken as input. Based on the real life data of virtual machine and cloudlets are created. Then virtual machines are assigned to host and cloudlet is assigned to virtual machine. Based on dynamic consolidation technique, status is checked for every scheduled interval. For every scheduled interval, under load detection algorithm is executed and less utilized hosts are put into sleeping mode by transferring all virtual machine to other active virtual machine. Then overload detection is executed, and overloaded hosts are identified. At later steps, virtual machine is selected from the overloaded hosts to migrate. Then those virtual machines are placed into available hosts or if needed a host is switched on from sleeping mode. After each iteration (the iteration time can be varied in CloudSim, most of the research have used 5 min as iteration interval [1, 2, 4]) a log is created to calculate energy consumption and QoS. At the end of the simulation, Energy consumption and QoS is shown. In next sections our proposed algorithms are discussed. More details of the iteration is discussed in section Experimental setup

Algorithm 1.Basic virtual machine consolidation

1. Input number of hosts;
2. Interface with real cloud data;
3. virtual machine is created and assigned to hosts;
4. Cloudlet is created and assigned to virtual machines;
5. For every specified time interval;
6. Execute under load detection;
7. Identify overload host through overload detection;
8. virtual machine is selected for migration from overloaded host;
9. virtual machine is placed in available datacenters;
10. Preserve history and calculate QoS;
11. End
12. Simulation ends and provides Energy consumption and other QoS value

A. Fuzzy virtual machine selection with migration control Fuzzy technique is an attractive approach to handle uncertain, imprecise, or un-modeled data in solving control and intelligent decision-making problems. Different virtual machine selection methods offer different advantages. It will be worthy if we could generate a method which will have the benefits of all of them by combining them together. Fuzzy logic can be an ideal tool for this. It will consider all the options and depending on those options a fuzzy value will be generated based on the predetermined rule of inferences.

To develop the fuzzy virtual machine selection method we have selected three distinguished inputs and each of them offers some advantages over others and different researches have already proven them. Minimum migration time and Maximum Correlation can be found at [2, 4] and the idea steady resource consuming virtual machine is adopted from[3]. The following subsections will be focusing on the variables we will be using as input to our fuzzy systems, membership function generated, inference rules and algorithms for computation.

1. Minimum migration Time Minimum Migration Time (MMT) policy selects the virtual machine which can be migrated within minimum time limit [2,4]. The migration time is limited by the memory the virtual machine is using and the spare bandwidth. At any moment t, The MMT with Migration Control policy finds virtual machine x that will be selected for migration by the formula (1). RAM(x) is the Radom Access Memory (RAM) utilization of virtual machine x and RAM(y) is the RAM utilization of virtual machine y. NET_h means the available bandwidth for migration and V_h is the set of virtual machines of host h. So the this method compares the migration time and selects the virtual machine x with minimum migration time among all virtual machines reside in host h.

$$x \in V_h | \forall_y \in V_h, \frac{RAM(x)}{NET_h} \le \frac{RAM(y)}{NET_h} \qquad (1)$$

This policy gives us the lowest SLA among all virtual machine selection models. Migration time will be considered as one input of our fuzzy system.

2. Correlation

This method works based on the idea that the higher the correlation between the resource usages by applications running on an oversubscribed server, the higher the probability of server being overloaded [14]. It means that if the correlation of the CPU utilization of virtual machines of a particular host is high then the probability of this host being overloaded is also high [4, 14]. Based on this research outcome, correlation is considered as a metric as it will provide the information about the virtual machine(s) that is going to cause the host to be overloaded. It is a predictive measure and consequently it will safer if such a virtual machine could be migrated to other host where it will not have higher correlation with other virtual machines. In the subsequent portion, it is described how the correlations of virtual machines are calculated. An augmented matrix is created for all virtual machines of host using last n cycles' CPU utilization and correlation value is calculated. The highest the correlation value of virtual machine, the higher the probability of that virtual machine makes the host to be overloaded.

As described in [4], let there are n number of virtual machines. Let Y be one out of those n virtual machines for which we want to determine the maximum correlation with other n-1 virtual machines. Here our objective is to evaluate the correlation of Y with the rest of virtual machines. The (n-1)* n augmented matrix is denoted by X. Each value in the matrix X represents the observed values of (n-1) virtual machines and y vector represents (n-1) observations of virtual machine Y.

$$X=\begin{bmatrix} 1 & x_{1,1} & \cdot & \cdot & x_{1,n-1} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n-1,1} & \cdot & \cdot & x_{n-1,n-1} \end{bmatrix} y = \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_n \end{bmatrix} \quad (2)$$

A vector of predicted value of virtual machine y is denoted by ŷ and expressed in Eq. 3.

$$\hat{y} = Xb, \quad \text{where } b = (X^T X)^{-1} X^T y \quad (3)$$

As we can find the predicted vector ŷ of Y, the multiple correlation coefficient ($R_{y,1....N-1}$) also can be determined as this is equal to the squared correlation coefficient of the observed values y and predicted values of ŷ of virtual machine Y. So the correlation coefficient can be defined by Eq. 4. Here m_y and $m_{\hat{y}}$ are the sample mean of the values of y and ŷ

$$R^2_{Y,X_1,....X_{n=1}} = \frac{\sum_{i=n}^{n}(y_i - m_y)^2(\hat{y}_i - m_{\hat{y}})^2}{\sum_{i=n}^{n}(y_i - m_y)^2 \sum_{i=n}^{n}((\hat{y}_i - m_{\hat{y}})^2} \quad (4)$$

Now the multiple correlation coefficient can be easily found for any virtual machine X_i by $R^2_{Y,X_1,...X_{n-i},X_{n+i},....X_n}$. According to this method the virtual machine that has the highest correlation with other virtual machines' CPU utilization will be migrated. More details of this method could be found in [1, 14].

3. Migration control metric for steady resource consuming virtual machine It has been proven that migration control provides better result in energy aware virtual machine consolidation and this approach also saves the unwanted traffic load [3]. Migration control can be done in various ways. We can stop migrating the high CPU using virtual machines or we can restrict steady resource consuming virtual machine from migration.

In this work we will take steady resource consumption as a non-migration factor. If a virtual machine's requirement highly fluctuates over time, then it can cause the host to be overloaded. In dynamic virtual machine consolidation approach, virtual machines are resized in each iteration according to their need. So when a virtual machine requests CPU which is abruptly high then host may not have such CPU available at that time and SLA violation will occur. As virtual machine migration is triggered from an overloaded host we do not want to migrate such virtual machine from the overloaded host whose demands of CPU is not changed suddenly. In other words, if a virtual machine is steady resource consuming over some iteration it means that it will be the least possible virtual machine to make this host overloaded and we can expect the same behavior in the next iteration. We have used standard deviation for calculation of steady state resource consumption. If the standard deviation is high it means that the CPU request changes abruptly and we can call virtual machines with low standard deviation as steady resource consuming virtual machines.

Let us consider a host h and V_h be the set of virtual machines in host h. $CPUu(x_t)$ is the CPU utilization of virtual machine X at time t. $CPUu(x_{t-1})$, $CPUu(x_{t-2})$ ….. $CPUu(x_{t-n})$ are the CPU utilizations of previous n time frames of virtual machine X. Migration control parameter can be given by Eq. 5. Here CPUaverage means average CPU utilization in last n time frames. The standard deviation of CPU usage of virtual machine X can be determined by this equation. This parameter will surely indicate the fluctuation of CPU usage of the particular virtual machine X.

$$\text{Stdev}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(CPU_i - CPU_{Average})^2} \quad (5)$$

4. Fuzzy Membership function

A FIS (Fuzzy Inference System) is developed to provide fuzzy virtual machine selection decision using three metrics as input. Member ship function needs to be defined to develop the FIS. We are using 4 linguistic variables including virtual machine selection as output. Range of these membership function is chosen from the real cloud simulation data of PlanetLab. In order to do the so, we have run the simulation and collected data of all these variables and proportioned to decide the range. As the ranges have been collected from real world data by doing statistical proportion operation (e.g. top 30 % values are high) for deciding different level (i.e. high, medium and low), using trapezoidal membership function is logical as it deals with ranges with flat region better. As the range of values should be counted as medium or low or high, not a peak value, triangular function is being not used and sigmoid function is not used as it does not define the flat region like trapezoidal function does. Membership function of the linguistic variables are given below:

- RAM: T(RAM) = {Low, Medium, High}
- Correlation: T(Correlation) = {Low, Medium, High}
- Standard Deviation: T(Stdev) = {Low, Medium, High}
- virtual machine selection: T(Vmselection) = {Low, Medium, High, Very High}

Equation for the Trapezoid membership function [27] can be expressed as Eq. 6.

$$\mu(z) = \begin{cases} 0 \;, & z \le a \\ \left(\frac{z-a}{b-a}\right) & a \le z \le b \\ 1 & b \le z \le c \\ \left(\frac{d-z}{d-c}\right) & c \le z \le d \\ 0 & d \le z \end{cases} \quad (6)$$

All the membership function graphs (Figs. 1, 2, 3 and 4) of the linguistic variables are given below and Table 1 shows the type of the membership function, the equation and the parameters.

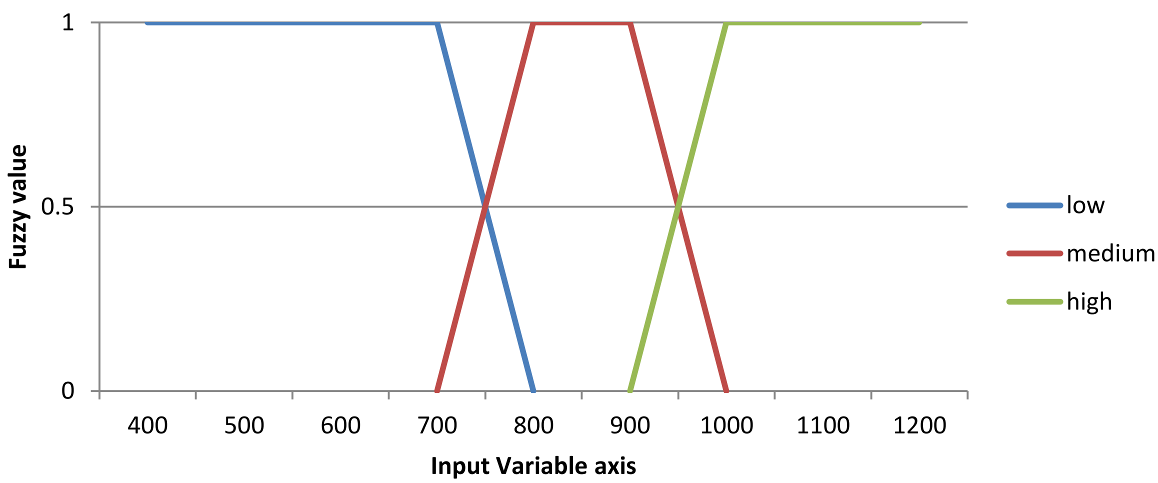
Table 1:Membership Function

Variables	Parameter		
	Level	Function type	Parameter
RAM	Low	Trapezoidal	a=0,b=0,c=5,d=750
	Medium	Trapezoidal	a=700,b=800,c=900,d=1000

	High	Trapezoidal	a=900,b=1100,c=1800,d=1800
Correlation	Low	Trapezoidal	a=0,b=0,c=5,d=.6
	Medium	Trapezoidal	a=5,b=.6,c=.8,d=.85
	High	Trapezoidal	a=.8,b=.85,c=1,d=1
Stdev	Low	Trapezoidal	a=0,b=0,c=3,d=3.75
	Medium	Trapezoidal	a=3.25,b=4,c=6.75,d=7.5
	High	Trapezoidal	a=7.5,b=8.5,c=100,d=100
virtual machine Selection	Low	Trapezoidal	a=0,b=0,c=.3,d=.35
	Medium	Trapezoidal	a=.3,b=.35,c=.6,d=.65
	High	Trapezoidal	a=.6,b=.65,c=.8,d=.85
	Very high	Trapezoidal	a=.8,b=.85,c=1,d=1

As mentioned earlier, values and ranges of these membership functions are generated by heuristic approach. The source is 1-day (among 10 days) data of thousands of virtual machine data of PlanetLab cloud network [25] (more discussed in section Experimental setup). For example, to deduce the standard deviation membership function, we have generated standard deviation value utilization of each of the thousand virtual machines. As these trace contains 288 (every 5 min from 1 day) data per virtual machine, total sample size is about 288,000 (288 trace data*1000 virtual machines). Using a window size of 10, standard deviation is calculated for the total data and by doing ration the high, medium and low ranges are selected. The minimum migration time and correlation is also done in same way.

Membership function:RAM



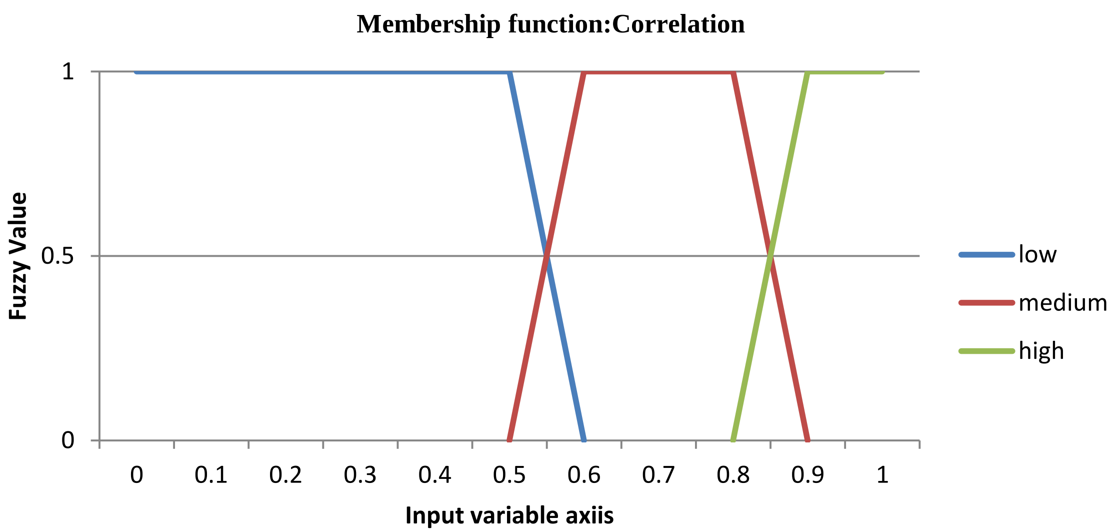
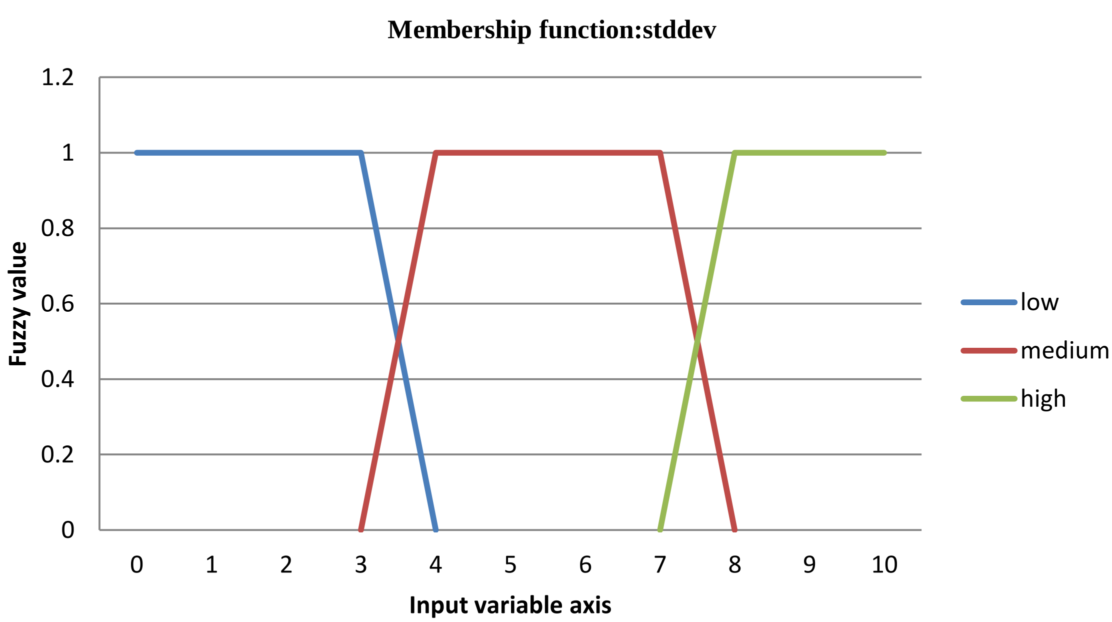


Fig 1: Membership function : Ram and Correlation



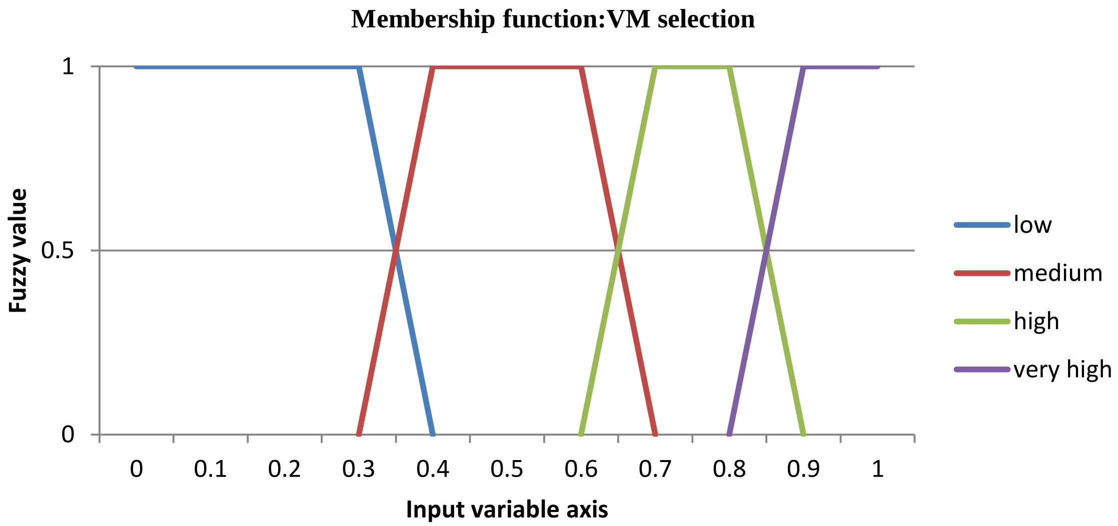


Fig 2: Membership function Standard deviation and Fuzzy output: virtual machine Selection

5. Fuzzy Inference Rule

Fuzzy inference rules are generated from the given linguistic variables. We have given equal weight on the variables to influence the virtual machine selection value. If RAM is low it gets high priority as it makes the migration faster. If correlation is high then it gets high priority in migration as the higher the correlation is, the higher the probability of overloading the host. Finally, if the standard deviation is high then it will get high priority in migration compared to its steady state counterparts. The following Table 2 depicts the inference rules.

Table 2:Fuzzy interference rule

Input			Output
RAM	Correlation	stdev	virtual machine selection
Low	High	High	Very high
Low	High	Medium	Very high
Low	High	Low	High
Low	Medium	High	Very high
Low	Medium	Medium	High
Low	Medium	Low	Medium
Low	Low	High	High
Low	Low	Medium	Medium
Low	Low	Low	Low
High	High	High	Very high

Medium	High	Medium	High
Medium	High	Low	Medium
Medium	Medium	High	High
Medium	Medium	Medium	Medium
Medium	Medium	Low	Low
Medium	Low	High	Medium
Medium	Low	Medium	Low
Medium	Low	Low	Low
High	High	High	High
High	High	Medium	Medium
High	High	Low	Low
High	Medium	High	Medium
High	Medium	Medium	Low
High	Medium	Low	Low
High	Low	High	Low
High	Low	Medium	Low
High	Low	Low	Low

B. Fuzzy virtual machine selection with migration control Algorithm Combination of Fuzzy virtual machine selection method and migration control is given in Eq. 7 and Eq. 8. These equations indicate that a virtual machine will be nominated for migration if it produces lower CPU usage than the migration control threshold and possesses highest fuzzy output value. If all virtual machines of an overloaded host produce more CPU usage than the migration control threshold, then the virtual machine that produces highest fuzzy output value will be migrated. It is described in detail below. Here the virtual machine x is selected for migration if the fuzzy output value of virtual machine x is greater than all other virtual machines. However, there is a condition that is as follows. If the current time is t and in last n cycles CPU utilization of virtual machine x is CPUu (x_t), CPUu (x_{t-1}), CPUu (x_{t-2})…CPUu (x_{t-n}), then the average CPU utilization must be less than CPUthreshold to satisfy migration control, i.e., not to migrate the highly utilized virtual machines. The Eq. 8 means if the average CPU utilization is above threshold for all the virtual machines then the virtual machine x with maximum fuzzy output value will be selected for migration.

$x \in V_h | \forall y \in V_h$ Fuzzy Output(x) $\geq$ Fuzzy Output(y)

Only if;

$$\frac{CPU_u(x_t)+CPU_u(x_{t-1})+CPU_u(x_{t-2})+\cdots CPU_u(x_{t-n})}{n+1} \leq CPU_{threshold} \quad (7)$$

However, if every virtual machine vm satisfies the following condition that means average utilization is more than the threshold,

$$\frac{CPU_u(vm_t)+CPU_u(vmx_{t-1})+\cdots CPU_u(vm_{t-n})}{n+1} \geq CPU_{threshold}$$

then this technique will select the virtual machine that produces the highest fuzzy output value;

$x \in V_h | \forall y \in V_h$ Fuzzy Output(x) $\geq$ Fuzzy Output(y) (8)

Algorithm 2:Fuzzy virtual machine selection Algorithm (FS)

Input: overload host h, window size n

Output: Virtual machine to be migrated virtual $machine_m$

1. Input overload host h;
2. virtual $machine_h$=Get migratable Vm(h);
3. virtual $machine_{hex}$=Exclude Vm in Migration;

4. Utlim(virtual machine$_{hex}$)=Utilization Matrix (virtual machine$_{hex}$);
5. Metric(n)=Correlation Coefficient(UtilM (virtual machine$_{hex}$));
6. For each virtual machine V_i of virtual machine$_{hex}$
7. CPU$_{hist}$=Get Mc Param From CPU history(v_i);
8. CPU$_{mc}$=Get migration control(CPU$_{hist}$);
9. STDEV(V_i)=Standard Deviation(CPU$_{hist}$);
10. RAM(V_i)=Get RAM(V_i);
11. MC(V_i)=Metric(V_i);
12. Output$_{fuzzy}$=Evaluate Fuzzy(STDEV(V_i),RAM(V_i),MC(V_i));
13. If output$_{fuzzy}$ is highest till now
14. virtual machine$_{highest}$=V_i;
15. If CPU$_{mc}$< CPU$_{threshold}$ then virtual machine$_m$=V_i;
16. End
17. Find;
18. If virtual machine$_m$ is null: virtual machine$_m$=virtual machine$_{highest}$;
19. Return virtual machine$_m$

The Algorithm 2 depicts how Fuzzy virtual machine Selection algorithm (FSMC) works. There are two inputs of the algorithm: the host h and window size n (CloudSim Default window size has been used). The overloaded host is detected by previous phase: Overload detection. After having the host h at step-1, at step-2, Get Migratable Vm(h) function pulls all the virtual machine which are currently placed on that host h. At step-3, Exclude Vm In Migration function excludes all the virtual machine which are already in migration for that host and assigned to virtual machine$_{hex}$. At step-4, the function Utilization Matrix calculates utilization matrix and stores at UtilM (virtual machine$_{hex}$). At step-5, function Correlation Coefficient calculates correlation coefficient based on UtilM (virtual machine$_{hex}$) and stores at Metric(n). At step-7, for each virtual machine V_i, CPU usage history of V_i is fetched using the function Get Mc Param From CPU History (V_i) for last n iteration as per CloudSim settings. At Step-8, Migration control parameter is calculated. To determine the steadiness of a virtual machine's CPU usage, at Step-9, STDEV(V_i), Standard deviation is calculated using Standard Deviation function from CPU$_{hist}$. At Step-10, current usage of RAM will be fetched for V_i and will check for if the one is the lowest up to now. At Step-11, Correlation for this virtual machine will fetched. At Step-12, fuzzy output Outputfuzzy is determined using Evaluate Fuzzy function where inputs of this function are STDEV(V_i), RAM (V_i) and MC (V_i). At step-13, If this one is the highest till now, at step-14, virtual machine$_{highest}$ will be updated. At step-15, virtual machine virtual machine$_m$ will be updated if CPU$_{mc}$ is smaller than CPU$_{threshold}$. If all virtual machine is greater than the threshold and the highest fuzzy output virtual machine is selected for migration at step-18 and finally step-19 returns the virtual machine to be migrated.

C. Mean, Median and Standard deviation based Overload Detection(MMSD)

Overload detection algorithm ensures that when a host is overloaded, then the algorithm will find it. Moreover, it will provide intelligent measure so that the datacenter does not get overloaded. There are several overload detection algorithms proposed in [1, 2, 4], 1) A Static CPU Utilization Threshold (THR): where overload decision is based on a static threshold; 2) Adaptive Median Absolute Deviation (MAD): the overload threshold is calculated dynamically using median absolute deviation; 3) Adaptive Interquartile Range (IQR): overload threshold is calculated dynamically using interquartile range method; 4) Local Regression(LR); and 5) Robust local Regression(LRR). LR and LRR are predication methods which will predict whether the host is going to be overloaded or not.

In this work, a new overload detection algorithm has been devised. When overloaded, a host incurs SLA violation. To be precise, a host incurs SLA violation when the required CPU utilization is greater than the actual utilization capacity. To avoid SLA violation, we have to design an overload detection mechanism which will predict this scenario. Host utilization is calculated from the virtual machine utilization. If the summation of mean (μ) and standard deviation (δ) of last n iteration is greater than the maximum capacity of the virtual machine then it can be inferred that this virtual machine will request more utilization than allocated in future [15]. We apply that idea in our research.

$\mu + \delta > 1$ (9)

Equation 9 means that if the summation of mean utilization of a virtual machine for last n cycles and standard deviation of utilization of that virtual machine for last n cycles is higher than the allocated CPU, then in next iteration this virtual machine can go beyond the maximum capacity of that virtual machine. As our objective is to keep SLA violation at the lowest level, we can calculate predicted utilization of all virtual machine of a corresponding host using Eq. (9) and check whether the total predicted utilization of a host is greater than the capacity or not. If the predicted value is greater, then the host is at risk of being overloaded and SLA violation. This technique we are going to apply in our overload detection algorithms However, mean and standard deviation is very much influenced by terminal values that refer the outlier values or values that are too large or too small. So, when the standard deviation is high (i.e. the value falls in the high range of standard deviation membership function of fuzzy virtual machine selection method) meaning that there is a possibility of high values present in the last n cycles. From the fuzzy membership variable Stddev, high range is considered. To avoid terminal values, when standard deviation is high, we replace mean with median and standard deviation formula is changed by replacing mean with median by Eq. (10). Hence, the prediction formula can be represented by Eq. (11). So it ensures, if in last n cycles any sudden fluctuation, i.e., very low or very high CPU utilization is found, this will not impact on the overall decision. δ_{Median} is the standard deviation calculated from median instead of mean. Like Eq. 9, Eq. 11 provides the prediction of a host. If the summation of Median and δ_{Median} is greater than 1 i.e. more than the total CPU then the host is considered to be overloaded.

$$\delta_{Median} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(CPU_i - CPU_{median})^2} \quad (10)$$

$$Median + \delta_{Median} > 1 \quad (11)$$

Algorithm 3:MMSD based overloaded detection(MMSD)

Input: host h

Output: True or false indicating Overloaded or not

1. Input overloaded host h;
2. virtual machine$_h$=Get V_m(h);
3. For each virtual machine$_i$ of virtual machine$_h$
4. Total Requested MIPS+=Get current MIPS(**V_i**);
5. Stddev=Get Stddev of Utilization(**V_i**);
6. Mean=Get mean of utilization(V_i);
7. Median=Get median of utilization(V_i);
8. Stddev$_{median}$=Get stddev using median(V_i);
9. If Stddev < StdDev Threshold
10. Total predicated MIPS+=Mean +stddev;
11. Else Total Predicated MIPS+=Median+Stddev$_{median}$;
12. End;
13. Utilization=Total requested MIPS/total MIPS of Host;

14. Prediction=Total Predicted MIPS/total MIPS of Host;
15. If Utilization>1 or prediction>1
16. Return True;
17. Else Return False.

Algorithm 3 describes how MMSD works. The input of the algorithm is the host of interest and the output of the algorithm is to determine whether the host is overloaded or not. At second step the virtual machines are identified which are currently active on the host. For every virtual machine a loop is started at step 3. Total requested MIPS (Million Instructions Per Second) is accumulated to get the total requested MIPS of the host. Then Mean, Standard deviation, Median and Standard deviation from median are calculated. Now the predicted MIPS is calculated by the standard deviation value. If the standard deviation is greater than the threshold (this threshold is taken from the fuzzy member ship variable Stddev's High value which starts from 8.5) then Eq. (11) is followed else Eq. (9) is followed. Then utilization of the host and predicted utilization of the host is calculated. If any of these are beyond 1(meaning 100%) then the host is marked as overloaded by returning true otherwise returning false.

D. Underload detection and handling

There is an underload detection and handling algorithm in CloudSim. The algorithm is simple, it sorts the hosts according to their utilization and starts with the lowest utilized host. If all virtual machines of a particular low utilized host can be placed to any some of active hosts using virtual machine placement method then the host is put to sleep mode by migrating all virtual machines to other hosts. virtual machine placement will be discussed later section. It is worthwhile to mention that before migrating virtual machine to other active hosts, the destination host is checked whether it will be overloaded by the new assignment with our newly designed overload detection algorithm.

E. Virtual Machine placement Algorithm

In CloudSim toolkit power aware BFD (Best Fit Decreasing) algorithm is used for virtual machine placement. When overload detection or under load detection finds virtual machines to migrate, virtual machine placement algorithm assigns the virtual machine in such way that power consumption is increased minimally [4]. virtual machine is placed in the host with decreasing utilization order. In this work it has been ensured that if a new virtual machine placement is considered, then our newly overload detection algorithm certifies that the destination host will not be overloaded in next iteration.

4. Experimental setup

In this experiment, we have implemented our algorithms in CloudSim 3.0.3 and analyze the performance of our proposed method. We have considered 800 heterogeneous physical nodes, half of which are HP ProLiant G4 and the rest are HP ProLiant G5 servers. Energy consumption is calculated based on HP ProLiant G4 and HP ProLiant G5 CPU usage and power consumption that is represented in Table 3 [4]. These servers are assigned with 1860MIPS (Million instruction per second) and 2660 MIPS for each core of G4 and G5 servers respectively. Network bandwidth is considered as 1GB/s. The virtual machines which were created were single core. virtual machine were of 4 types, for example, High-CPU Medium Instance (2500 MIPS, 0.85 GB); Extra Large Instance (2000 MIPS, 3.75 GB); Small Instance (1000 MIPS, 1.7 GB); and Micro Instance (500 MIPS, 613 MB). Fuzzy rules are defined and integrated to Cloud- Sim using JFuzzy Logic Tool [13].

In this work we have used real world work load data that is provided from CoMon project, a monitoring infrastructure for PlanetLab [25]. This data is collected from more than thousand virtual machines of different servers that are located in 500 different locations. The workload is representative of an IaaS cloud environment such as Amazon EC2, where virtual machines are created and managed by several independent users. Table 4 presents the day wise virtual machine number for this data.

Table 4: Day wise planet lab data

Date	Number of virtual machines
3 March	1052
6 March	898
9 March	1061
22 March	1516
25 March	1078
3 April	1463
9 April	1358
11 April	1233
12 April	1054
20 April	1033

These real world traces contain virtual machine utilization records in every 5-min interval. Ten days' data of year 2011 have been used in this experiment. Each virtual machine contains 288 (=24*(5/60)) data of CPU utilization. The simulation checks CPU data every 5 min interval and those trace data is plugged into dynamic virtual machine consolidation.

The main target of virtual machine consolidation is to reduce energy consumption and at the same time the QoS should be at an acceptable level. The energy consumption metric is discussed below and for QoS parameter, several metrics are stated which are used in several researches [2,4]. The main QoS is SLA violation. In virtual machine consolidation SLA violation occurs due to host overload and virtual machine migration. To quantify SLA violation for overloaded host, the metric Overload time fraction (OTF) is used and on the other hand to quantify the SLA violation due to virtual machine migration, the PDM (performance degradation due to migration) is defined. SLAV (SLA violation) is the product of OTF and PDM. Moreover, the number of virtual machine migration indicates the efficiency of the consolidation method which is also described as a metric. But the main objective of our research is to obtain Energy-QoS trade off and that is defined by the metric ESV which is the product of energy consumption and SLA violation (SLAV). So the method providing the lowest ESV and at the same, the lowest energy consumption and the lowest SLA violation, is undoubtedly the best method. Based on these six metrics proposed method will be verified and they are described in more detail and mathematically below.

1) Energy Consumption(kWh)

This is the main metric as the target of virtual machine consolidation is to reduce energy consumption. Energy consumption is computed by taking into account all hosts throughout the simulation by mapping of CPU and energy consumption from Table 3. At each iterations the CPU utilization is measured and power consumption is calculated from Table 3 and at the end of the simulation energy consumption is measured by accumulating all hosts energy consumption.

Table 3:Power consumption of different level utilization

Machine Type	Power Consumption based on CPU utilization					
	0%	20%	40%	60%	80%	100%
HPG4(watt)	86	92.6	99	106	112	117
HPG5(watt)	93.7	101	110	121	129	136

2)Number of virtual machine migration

This metric counts the number of virtual machines migrated during the siumaltion.

virtual machine migration is an important factor because unnecessary migration causes SLA violation and network traffic.

3) OTF

Overload time fraction [4], OTF is a measure of SLA violation. it provides a measure of the fraction of time a host experienced 100 % CPU utilization leading to SLA violation. In Eq. (12), if N is the number of hosts, T_{si} is the total time when host i experienced 100 % utilization leading SLA Violation, T_{ai} is the total active time of host i, then OTF is defined by:

$$\text{OTF}=\frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}}$$

$$\text{PDM}=\frac{1}{M}\sum_{j=1}^{M}\frac{C_{dj}}{C_{mj}} \qquad (12)$$

4) PDM

Performance degradation due to migration(PDM) [4] is a measure of SLA violation. It measures the total SLA violation due to virtual machine migration. When a host is overloaded, virtual machines are migrated from that host to non-overloaded host. At the time of migration, that virtual machine is not capable of serving user needs, hence, it causes SLA violation. This metric calculates the SLA violation caused by migration. From Eq. (12), if M is the number of total virtual machines, C_{dj} stands for the CPU request at the time of migration of virtual machine j and C_{rj} stands for total CPU requested by virtual machine $_j$, then PDM is defined by the Eq. (12).

5) SLAV

Service level agreement violation, SLAV, is combined impact of OTF and PDM. It provides a SLA violation measure for the simulation which is a product of OTF and PDM i. e., SLAV = OTF*PDM.

6) ESV

Energy consumption and SLA is already defined. It is perceivable that if we try to reduce too much energy consumption the SLA violation will be increased, because consolidating many virtual machines in a host increase the probability of overload. So it is desirable to obtain a method which will consume less power and still incur less SLA violation. To measure this, ESV is introduced. It is the combination of Energy consumption and SLA violation, i.e., ESV = Energy*SLAV. So this can be treated as one metric to make an overall measurement. If the product of energy consumption and SLA violation is lower, it means that the approach reduces energy consumption and making less SLA violation.

6.Experimental result

In our experiment, using CloudSim, we have experimented with five Overload detection algorithms (IQR, LR, LRR, MAD and THR) and three virtual machine selection (MC, MMT, RS) methods. So in combination there are 15 methods (IQR_MC, IQR_MMT, IQR_RS, LR_MC, LR_MMT, LR_RS, LRR_MC, LRR_MMT, LRR_RS, MAD_MC, MAD_MMT, MAD_RS, THR_MC, THR_MMT, THR_RS) which will be compared against our proposed MMSD_FS method based on aforementioned performance metrics. Based on the result for 10 days Box graphs have been prepared to compare the results. It is represented in Figs. 3, 4, 5, 6, 7 and 8. We discuss the performance with respect to each metric are given below.

A. Energy Consumption

Main objective of this research is to design a virtual machine consolidation algorithm so that the energy consumption is reduced. By comparing the proposed and existing methods in the Fig. 3, it is found that energy consumption is significantly reduced in proposed (MMSD_FS) method. Minimum energy consumption by the proposed method is 102 Kwh where the minimum of all other methods is 112 Kwh, therefore we got 8.5 % reduction. If we consider average value, MMSD_FS consumed 136.5 Kwh and all other methods consumed 169 Kwh on average resulting 19 % energy saving. Therefore, we can infer that the basic objective of this research is achieved by saving energy consumption.

B. SLA Violation

SLA violation is one of the key indicators of QoS. SLA Violation is calculated by keeping two scenarios into consideration, i) if any virtual machine got overloaded, and ii) The SLA violation incurred while migration. A method having low SLA violation ensures the desired QoS. From Fig. 4, SLA violation is decreased significantly which is clearly visible. Minimum SLAV by proposed method is 0.0004 % whereas the minimum of all other method is 0.00279 %, resulting 84 % reduction. If we consider average value, MMSD_FS incurred 0.0005 % SLAV and all other methods incurred 0.00617 % on average, resulting 91 % reduction in SLA violation. This is main achievement of this research. It means that the overload detection method we have used, predicted the overloaded host efficiently and as an outcome, SLA violation was dropped significantly. If host overload is predicted successfully then there will be less number of migration which will reduce SLA violation as well.

C. ESV

As energy consumption has been successfully reduced by the proposed method, now energy-QoS trade off needs to be checked. ESV is the metric which is a product of Energy consumption and SLA violation; hence, provides a tradeoff picture of the proposed method with other existing methods. From previous two sub-sections, we have observed that both energy and SLA violation reduced, so it is inevitable that ESV will also be reduced significantly. From the Fig. 5, ESV is found to be reduced significantly which is clearly visible. If ESV reduces it means that this approach saves energy and at the same time SLA violation is controlled. As ESV is reduced significantly, it means that Energy and SLA tradeoff has been achieved. Minimum ESV by proposed method is 0.04 whereas the minimum all other method is 0.49, resulting 91 % reduction. If we consider average value, MMSD_FS incurred 0.07 ESV and all other method incurred 1.09 on average, resulting 93 % reduction in ESV. D. Number of virtual machine migration Less number of virtual machine migrations means efficient consolidation, less traffic in cloud network and less SLA violation for virtual machine migration. Reduction in Number of virtual machine migration is also clearly visible.

D. Number of virtual machine migration

Less number of virtual machine migrations means efficient consolidation, less traffic in cloud network and less SLA violation for virtual machine migration. Reduction in Number of virtual machine migration is also clearly visible from Fig. 6. To quantify, minimum number of virtual machine migration caused by MMSD_FS is 5185 whereas the minimum all other method is 16,317, resulting 68 % reduction. If we consider average value, MMSD_FS incurred 7943 migrations on an average and average of all other methods is 24,929, resulting 68 % reduction in migration. From this percentage it is evident that the proposed method provides most optimum virtual machine consolidation compared to the existing virtual machine consolidation approaches.

E. OTF and PDM

From Fig. 7 to Fig. 8, it can be inferred that OTF and PDM is significantly reduced.

The proposed method reduced both SLA violation due to overload and SLA violation due to virtual machine migration. Minimum OTF is reduced up to 60 % by the proposed method and Minimum PDM is reduced up to 64 %. On an average OTF is decreased by 67 % and PDM is decreased by 74 % compared to the existing methods. Finally, we have performed a statistical test namely two-tailed students' t-test on the performance of the proposed method MMSD_FS and IQR_MMT method (the best method in CloudSim as per the ESV, Fig. 4). Our null hypothesis is: "There is no significant difference in the performance between two techniques". Table 5 reports p-values for six performance metrics between MMSD_FS and IQR_MMT generated from 10 days' experimental data. If the p-value is greater than 0.05, then we must accept the null hypothesis, otherwise we must reject the null hypothesis.

Table 5: P-values for performance metrics

Metric	p-value
Energy consumption	0.004
ESV	2E-09
Number of migration	1.4E-08
SLAV	6.78E-12
OTF	5.38E-19
PDM	1.44E-12

From Table 5 we find that the p-value is significantly smaller than 0.05 for every performance metric. Therefore, we must reject the null hypothesis and we could conclude that there is significantly difference in performance found. Fig. 4 SLA violation Fig. 5From all the performance metrics it can be inferred that the proposed method outperforms all other methods.

F. A Deep dive

From the above result analysis, we have found that the proposed method improved significantly. Most of the improvement came from the SLA violation part. This phenomenon indicates that the proposed method identifies the host overload efficiently. To visualize the performance in easier way we have generated two heat maps of MMSD_FS method and another is IQR_MMT method which are given in Fig. 9 and Fig. respectively.

For this experiment, we have used 50 hosts and 50 virtual machines and random load. In the heat map, if a host is in sleeping mode i.e., 0 % utilization then it is marked by blue color and red color for high utilization. In the X-axis the time is portrayed. As iteration duration is 5 min, so there are total 288(starting from 0 to 8600 s) iterations as the simulation is done for 1-day data. Y-axis represents 50 hosts. From Fig. 9, it is visible that hosts are experiencing ON-OFF frequently and the map seems scattered and the total number of overload occurs 685 times. So this method will invoke virtual machine migration at least 685 times. On the other hand, Fig. 10 shows the heat map for MMSD_FS where we can observe less fluctuation (ON-OFF) of the host. It is easily perceivable that the host is put to sleeping mode and stays in sleeping mode for long. Total number of overload incident is 92, which indicates the efficiency of the algorithm. The main reason behind the performance of MMSD_FS is the prediction done by this algorithm helped to reduce the number overloaded hosts.

6. Conclusion

In this research we have devised algorithm for fuzzy virtual machine selection method and introduced migration control in the selection method. Fuzzy virtual machine selection methods take intelligent decision to select a virtual machine to be migrated from one host to the other. Then we designed mean, median and standard deviation based overload detection algorithm. After simulation and making comparison against existing methods, it has been found that the proposed method outperformed other previous methods in both perspectives, i.e., more energy saving and less SLA violation….

References

1. Beloglazov A, Abawajy J, Buyya R (2011) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Futur Gener Comput Syst (FGCS) 28(5):755–768

2. 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: Practice and Experience (CCPE) 24(13):1397–1420

3. Ferreto TC, Netto MAS, Calheiros RN, De Rose CAF (2011) Server consolidation with migration control for virtualized data centers". Futur Gener Comput Syst 27(8):1027–1034

4. Beloglazov A, (2013) PhD Thesis: "Energy-Efficient Management of Virtual machines in Data Centers for Cloud Computing". Link: http://beloglazov.info/ thesis.pdf. Accessed 04 Jul 2016

5. Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 41(1):23–50

6. CloudSim link: http://www.cloudbus.org/cloudsim/. Accessed 04 July 2016

7. Monil MAH, Qasim R, Rahman RM (2014) Incorporating migration control in virtual machine selection strategies to enhance performance". Int J Web Appl (IJWA) 6(4):135–151

8. Monil MAH, Qasim R, Rahman RM (2014) "Energy-aware virtual machine consolidation approach using combination of heuristics and migration control", 9th IEEE International Conference on Digital Information Management, pp. 74–79.

9. Farahnakian F, Ashraf A, Liljeberg P, Pahikkala T, Plosila J, Porres I, Tenhunen H "Energy-Aware Dynamic virtual machine Consolidation in Cloud Data Centers Using Ant Colony System" 2014 IEEE 7th International Conference on Cloud Computing (CLOUD), pp. 104–111

10. Prevost J, Nagothu K, Kelley B, Jamshidi M (2011) "Prediction of cloud data center networks loads using stochastic and neural models," in Proc. IEEE System of Systems Engineering (SoSE) Conf., pp. 276–281, 27–30

11. Di S, Kondo D, Cirne W (2012) "Host load prediction in a Google compute cloud with a Bayesian model", in Proc. Of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Salt Lake City, UT, Nov. 10–16

12. Farahnakian F, Liljeberg P, Plosila J (2013) "LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers," 2013 39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 357–364.

13. Cingolani P, Alcala-Fdez J (2012) "jFuzzyLogic: A robust and flexible Fuzzy-Logic inference system language implementation", Proc. Intâl Conf. Fuzzy Systems, pp. 1–8

14. Verma A, Dasgupta G, Nayak TK, De P, Kothari R (2009) "Server workload analysis for power minimization using consolidation," in Proceedings of the 2009 USENIX Annual Technical Conference, pp. 28–28.

15. Cao Z, Dong S (2013) "Energy-Aware framework for virtual machine consolidation in cloud computing" High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference, pp. pp. 1890–1895

16. Akula GS, Potluri A (2014) "Heuristics for migration with consolidation of ensembles of Virtual machines" Sixth International Conference on Communication Systems and

Networks Communication Systems and Networks (COMSNETS), pp. 1–4
17. Madani N, Lebbat A, Tallal S, Medromi H (2013) "New cloud consolidation architecture for electrical energy consumption management", IEEE Africon 2013 Conference, pp. 1–3.
18. Farahnakian F, Liljeberg P, Plosila J (2014) "Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning", 22nd Euromicro International Conference on Parallel, Distributed and Networkbased Processing (PDP), pp. 500–507
19. Mastroianni C, Meo M, Papuzzo G (2013) "Probabilistic Consolidation of Virtual machines in Self-Organizing Cloud Data Centers", IEEE Transaction of Cloud Computing, vol.1, pp. 215–228.
20. Barroso LA, Holzle U (2007) The case for energy-proportional computing. Computer 40(12):33–37
21. Duy TVT, Sato Y, Inoguchi Y (2010) "Performance evaluation of a green scheduling algorithm for energy savings in cloud computing", 2010 IEEE international symposium on parallel & distributed processing. Workshops and Phd Forum (IPDPSW), Atlanta, pp. 1–8
22. Feller E, Rilling L, Morin C (2011) "Energy-aware ant colony based workload placement in clouds", Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing, pp. 26–33.
23. Mills K, Filliben J, Dabrowski C (2011) "Comparing virtual machine-placement algorithms for on-demand clouds, IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), Athens, 2011, pp. 91–98.
24. Srikantaiah S, Kansal A, Zhao F (2008) Energy aware consolidation for cloud computing, Proceedings of the 2008 conference on Power aware computing and systems, pp. 10–10.
25. Park KS, Pai VS (2006) CoMon: a mostly-scalable monitoring system for planet- Lab. ACM SIGOPS Operating Syst Rev 40(1):65–74
26. Monil MAH, Rahman RM (2015) "Fuzzy Logic Based Energy Aware virtual machine Consolidation", 8th International Conference on Internet and Distributed Computing Systems (IDCS 2015), Windsor, U.K., September 2–4, pp. 223–227
27. Fuzzy Trapezoidal membership function. Link: http://www.mathworks.com/ help/fuzzy/trapmf.html?requestedDomain=www.mathworks.com. Accessed 04 Jul 2016
28. Beloglazov A, Buyya R, Lee YC, Zomaya A (2011) "A taxonomy and survey of energy-efficient data centers and Cloud computing systems," Advances in Computers, M. Zelkowitz (ed.), vol. 82, Elsevier, pp. 47–111.