International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 07, July-2019

EEMFRA:Energy Efficient Multi-Decisive Fuzzy-based Resource Allocation Mechanism in Cloud

Asha N, Dr. G.R. Rao
CS & E Department
National Institute of Engineering
MYSORE, India,

***Abstract*—The emergence of cloud computing has lead to an increase in the growing rate of datacenters so as to serve the huge needs of the users. These datacenters housing the physical nodes emit large volume of CO_2 to process the requests thereby increasing the operational costs. Virtualization technology with the capabilities as VM consolidation and migration techniques plays a major role in the replacement of actual physical machines by virtual machine(VM)s, thus enabling the reduction of a large amount of energy consumed by the active servers. In continuation of this, with the sole aim of improving the energy efficiency, we have employed Fuzzy based scheme in this paper that involves VM consolidation followed by shifting of VMs from an Over-Provisioned(OP) or Under-Provisioned(UP) host onto other host nodes. These techniques try to bring a balance between the workloads of the machines and try to improve SLA to a certain extent. This paper also demonstrates through simulation study, an outperforming behavior of our proposed algorithm with those of the existing techniques.**

Keywords—Energy efficiency; virtualization; overprovisioning; underprovisioning; consolidation

I. INTRODUCTION

As cloud computing[1] is gaining primary focus across researchers of computing and network area datacenters have occupied prominent place of accommodating scalable services to the users in terms of computing and data storage. As these datacenters are consuming lots of electricity through its equipments[2] housed within it, it is incurring high operational cost. Some of the components contributing to high energy consumption are processing units, servers, cooling units, networking systems and lighting components[3]. There are few existing methods that are already available which are meant for minimization of the power consumed[4-6]. Unfortunately, they are not ideal for latest datacenters. However, direct measurement of energy consumed by various components is really a challenging problem. This can be made happen with a proper RA and management algorithms so that resource utilization is maximized and energy consumption can be minimized. Virtualization[7], which plays a major role in cloud computing provides an abstraction of logical resources from their corresponding actual resources. Virtual Machine Monitor(VMM), the software used in this technology divides the computer system resources into multiple VMs and provides an abstraction from the host Operating System(OS). Each of these multiple VMs hosing guest OSs will get an impression themselves that they are the sole user of the entire system, being ignorant of other guest OSs sharing the same system. This technology offers multiplexing of VMs running on UP machines into fewer machines by switching off these UP machines thereby saving the energy considerably. VM multiplexing is essentially a hot topic of the researchers that involves aggregation of VMs into fewer hosts designated as target hosts based on the host utilization information. Performance degradation and resource contention may result from an OP server state, which is another crucial step in VM aggregation, also contributes to high power consumption. Hence, there is a need to monitor such hosts helps in improving resource utilization and prevents exhausting of VMs. Hence, there is an urgent need to identify these two Host Over-Provision(HOP) and Host Under-Provison(HUP) states, failing which will also lead to Service Level Agreement(SLA)[9] violation(a contractual engagement between user and provider). Hence, this paper introduces an algorithm of an energy efficient Multidecisive Fuzzy-based Resource Allocation(EEMFRA) for both HUP and HOP state detection based on a threshold value strategy[10], which is nothing but a maximum or minimum CPU utilization percentage value which can be maintained as either static or dynamic. Our proposed work exhibits superiority over existing algorithms to be discussed in the next section.

Thus this paper is organized as follows: Related work discusses the various forms o existing works with some of the gaps identified and covered in our work. Proposed work section introduces the relevant algorithm of our work along with the methodologies used. Experimental setup section depicts the results of simulation and lastly conclusion section follows with scope for future work.

II. RELATED WORK

Most of the works on VM multiplexing are divided into two parts: one is threshold-aware and the other is threshold-less types. Again threshold-aware methods are divided into static, adaptive and regression based thresholds[10]. In this section, we set our focus on significant contribution on these. In fact, these plays a prima facie role in the detection of HUP and HOP states.

In the case of threshold-aware type, host machine’s total CPU usage is matched against a value. In our work, we have called it as a Ceiling Value(CV), which are of two types: fixed and adaptive. If the host CPU usage value exceeds this CV, HOP is detected else if the same CPU usage value falls below this CV, HUP is detected. However, these static CV are not ideal for dynamic workloads. Thus adaptive[11] CV based algorithms sets automatically CPU usages to a maximum or minimum value based on an analysis of past-history of data collected[12]. This is actually a good approach, but fails to infer HOP state. During VM to Physical Machine(PM) mapping, fuzzy approach is used for VM placement in our work.

Regression based algorithms are explained in [10]. Basic idea of this work is to develop a model to approximate the actual data. In some of the works, threshold-less types are suggested that chooses the source host machines either randomly or based on functions that tests maximum or minimum resource usage[13-14]. But threshold-aware work types are popular.

Authors in [15] have proposed multiple regression algorithms that produced results have used random and PlanetLab workload datasets. But it is only the preliminary stage. In one of the reports in[16], authors have addressed VM placement problem which is one of the stages in VM multiplexing approach. Some of the advance energy-efficient VM migration techniques are discussed in [17]. Works in [18] have discussed VM placement algorithm based on genetic theory. Our work has incorporated algorithms for CV computation, HOP and HUP detection in a single implementation.

III. PROPOSED WORK

Our proposed system architectural diagram is represented in Fig 1. We have concentrated only on IaaS of cloud computing to provide better services through on-demand resource provisioning. One or more host machines are connected to switches, which can be again connected mutually to form a network communication. It is assumed that size of each of the components of multiple VMs created within each Physical Machine(PM) will not exceed that of maximum capacity of all the resources of the PM.

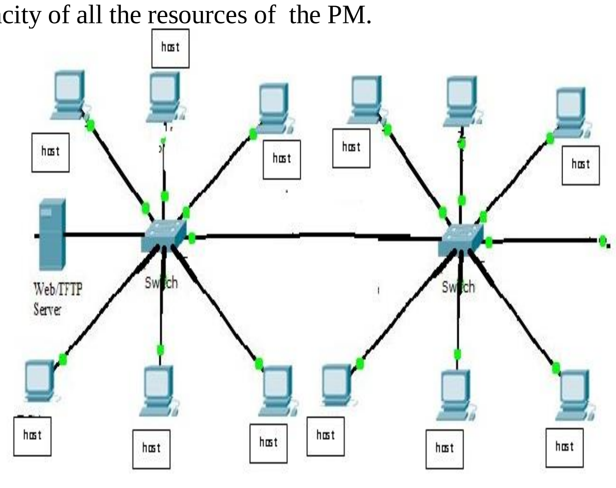

Fig. 1. Architectural Diagram

Following are the steps involved in our proposed work:

a. Setting up of suitable CVs
b. HOP/HUP detection
c. VM aggregation
d. Fuzzy based host discovery for VM to PM mapping

Two types of CVs are maintained in our work: Upper CV(UCV) and Lower CV(LCV). These values are not fixed instead they are computed and discussed in the next section. HOP state is detected if the total energy consumption by CPU falls beyond UCV. HUP state is detected if the same CPU usage falls below LCV. After these detections, either few or all of the VMs are marked and shifted on to other hosts. This process is called VM migration. It is to note that during this process, we have ensured that destination hosts launching these migrated VMs are not entering into HOP state which in turn should not get locked in continuous VM migration process and hence may cause increased energy consumption. Another point is also observed that the destination host is having enough resources to match with those of the migrated VM's.

For this reason, we have employed a technique of computing a Rank_Score(RS) for each of the host based on CPU usage. Initially, before we start allocating this request to VM, a test is made about its existence. If so and has sufficient resources to serve the request, it is assigned else, a new VM satisfying the requirements is created. For this machine, based on Fuzzy logic theory[19], RS is calculated. More details about the implementation of these steps are explained in upcoming sections.

IV. METHODOLOGY EMPLOYED

A. Fuzzy logic

In recent years, fuzzy logic is like an Artificial Intelligence(AI) technique used for employing intelligent methods to get the desired result. This approach is used in our work for both allocation and migration of VMs in which a Rank_Score(RS) is calculated for all the hosts in the list. Originally, Fuzzy logic was hosted by Prof. L.A Zadeh, from California University at Berkley in 1965[19]. This method is somewhat similar to human psychology for making decisions faster. It is a superset of Boolean logic that handles the concept of true and false, but more than it. This logic routes us in a simple way to come to certain conclusions based on ambiguous, imprecise information. Propositions are represented with degrees of truth. It is a decision-making Rule-based Support system. It uses simple IF-THEN rules by means of linguistic values as expert knowledge and uses the inference mechanism to perform appropriate action control. Fuzzy-logic-controller [20-24] is comprised of (i) a Knowledge Base that is composed of the information in the form of linguistic rules; (ii) Fuzzification Interfaces that maps input variables values into fuzzy variables using membership functions that is used in the fuzzy interface; (iii) Inference System that takes values from fuzzification phase and knowledge base to apply reasoning; (iv) Defuzzification done at the end translates the fuzzy action from the influence into control variable.

In our fuzzy Rank_Score calculation system, following are the input variables used EU(Energy Usage), CR(Communication Rate) and output variable is RS(Rank Score).

If we have a fuzzy set F, and if an element i is a member of this set, then the mapping can be

$$\mu_F(i) \in [0,1] \exists F = (i, \mu_F(i)) | i \in U \quad (1)$$

where U is the universal set and μ_F is a discrete membership function.

Linguistic partitions with uniformly distributed piecewise linear membership functions are as shown in Fig. 2. Low(L), High(H), Medium(M) represents linguistic values, and EU, CR represents input values from the analysis and RS is the output for n number of samples

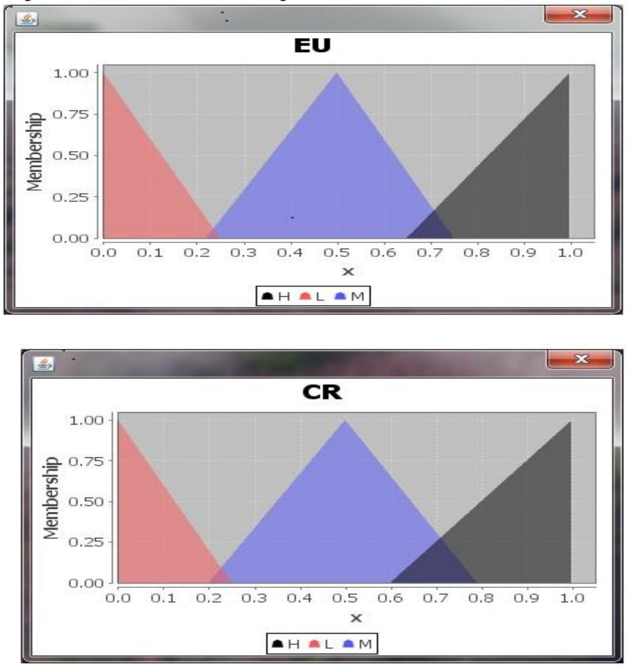

Fig. 2. Uniformly Distributed functions for Input variables

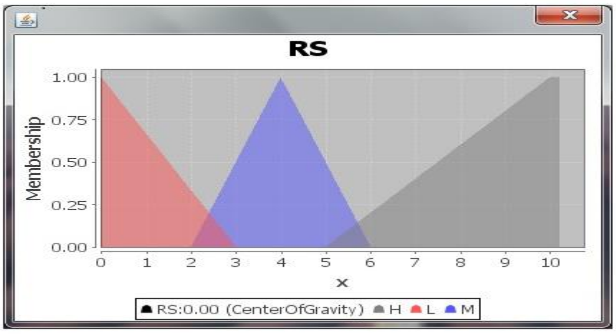

Fig. 3. Uniformly Distributed function for Output variable

Finally an optimization API is developed to fine-tune Fuzzy Inference System (FIS) parameter. jFuzzylogic treats each variable double datatype (REAL here) in java. After global optimization, we can observe membership functions differing from the originals. To represent the RS high to allocate the VM, Centre of Gravity (COG) defuzzification [21] method is used.. For example, to represent the Rank_Score to allocate the VM, COG output can be

$$CG = \frac{\sum_{i=1}^{n} d_i \mu_{ps}(d_i)}{\sum_{i=1}^{n} \mu_{ps}(d_i)} \quad (2)$$

Where d_i is a sample fuzzy set member of RS and μ_{ps} is discrete membership function of the corresponding member in the fuzzy set.

B. VM Selection

Selection of VMs is based on double valued CVs. We choose the migrating VMs to those hosts whose CPU usage is low and aim to reduce the migration rate depending on whether this value is violating the rule of LCV <= CPU_{util} <= UCV. These values are calculated statistically depending on the past history during lifetime of VMs[25]. For each of the host, CPU usage is calculated to determine whether it comes under HOP or HUP state. Based on this, either few(based on a random policy) or all VMs within the host machine are marked for migration. Now, it is the time to choose the new host machines for the placement of the marked VMs towards migration. If more number of such hosts are available, they are all assigned with a RS value and those nodes with the highest RS is chosen for VM allocation. Fig. 4 shows the pictorial representation of the approach.

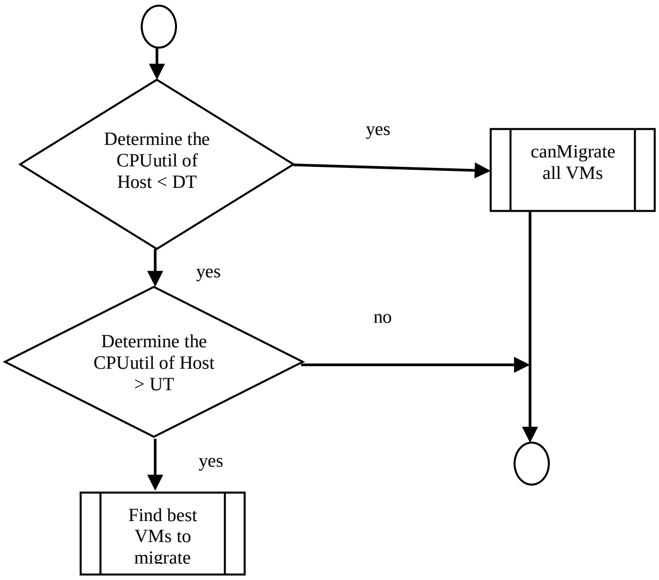

Fig. 4. Selection of VMs Approach for Migration

C. Ceiling Value

In our approach, we have considered variable TVs for the determination of OP and LP hosts. Towards the calculations of the same, we have used following computations to arrive at Upper Ceiling Value(UCV) and Low Ceiling Values(LCV). Approximate CPU usage of each of the VM can be calculated as

$$CPU_{util} = \frac{requesed\_MIPS}{VM\_MIPS\_assigned} \quad (3)$$

Total CPU usage by all the VMs is

$$T_{CPU} = \sum_{i=1}^{u} CPU_i \quad (4)$$

$$ST_{CPU} = \left(\sqrt{\sum_{i=1}^{u} (CPU_i)^2} \right) \quad (5)$$

For u VMs, UCV can be computed as

$$UCV = 1 - ((LP_{max} * ST_{CPU}) + T_{CPU}) - (LP_{min} * ST_{CPU}) + T_{CPU} \quad (6)$$

where LP_{max} and LP_{min} are the assumed maximum and the minimum limits of probability for the amount of CPU size.

Similarly for LCV computation, we have considered the CPU usage is above 30%, single maximum limit of probability for the LT as 0.3 and host machine is hosting u

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 07, July-2019

VMs. $$T_{CPU} = \frac{\sum_{i=1}^{u} CPU_i}{u} \quad (7)$$

$$ST_{CPU} = \sqrt{\sum_{i=1}^{u} (CPU_{util} - T_{CPU})^2} \quad (8)$$

If CPU usage is greater than 30%

LCV=0.3 (9)

else if less than 30%

$LCV = T_{CPU} - (LCV * ST_{CPU})$ (10)

V. EXPERIMENTAL ANALYSIS

A. Algorithmic Approach

Algorithm 1: Fuzzy Based Host Discovery

Input: VMs demands from user U_r

Output: HS_{out}

1. **Begin**
2. **foreach** host node in Hostlist, **do**
   a. Extract current EU,CR values
   b. Computes RS by applying COG method
   c. RS: $RS_{host}(i) \leftarrow FIS(EU,CR)$
   endfor
3. Select the highest-RS host
   $HS_{out} \leftarrow MAX(RS_{host})$
4. Go to step 2 and repeat for next VM allocation based on resource demand
5. **end**

Fig. 5 shows selection of host nodes for the allocation of VMs.

Algorithm 2: Enhanced HOP state detection

Input: Host node under test

Output: Boolean Value indicating EP or not

1. select VMs within this host and prepare a list VMList
2. **for** each i of VMList **do**
3. **Begin**
4. demCPU=TotReqCPU + presReq(i)
5. Varc=EvalVarc(i)
6. Avrg=EvalAvrg(i)
7. Midval=EvalMid(i)
8. **if** var < HCV **then**
9. PredictVal=Predictval + Avg + Var
10. **else**
11. PredictVal=PredictVal + Midvalue + Var
12. **end**
13. presUsage=ReqCPU / CPUutil
14. expected=PredictVal / CPUUtil
15. **if** either of presUsage or expected value is greater than MAX(CPUHost) **then**
16. **return** true
17. **else**

return false

B. Simulation Setup

For our simulation, we have used CloudSim 3.0 toolkit[26] as the simulator as it is very easy to realize the cloud environment. It is a modern simulation tool that supports the real world realization of datacenter with virtualized resources and management. Initial setup is initialized with different configurations of hosts, VMs, switches and cloudlets and separate input files are used to fetch these files to observe the varied outputs. Each of the HostList contains host machine-id, number of cores, CPU Length in MIPS(Million Instructions per Second), RAM Size, Storage and Bandwidth. Each of VMList contains VM-id, instruction length in MI(Million Instructions) required, Number of Cores, RAM Size, VM Image Size, and Network Bandwidth.

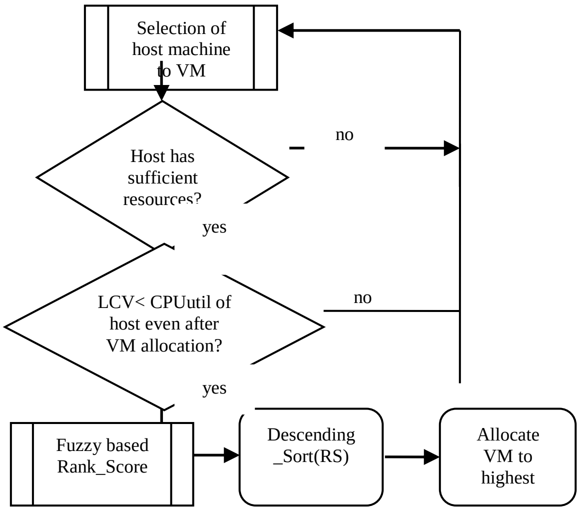

Fig. 5. Choosing of Host nodes for VM Allocation

Each of the cloudlet contains cloudlet-id, length of instructions, processor cores required, input file-size, and output file-size and CPU-I/O ratio. Lastly, each of switches contains switch-id and host-machine-list as a part of the input configuration file.

Fig. 6 to Fig. 9 shows the graphical analysis of the various parameters as a measure of performance in comparison with existing algorithm during multiple runs. It is also observed from the results, through rigorous simulation runs using various configuration files, it is proven that our proposed work stands high compared to the existing approach.

To define them, Resource Utilization is a measure of maximum resource exploration where any change in usage depends on the determination of the obtainable resource capacity within the host node. This node monitors regularly the availability of the resources that is housing and their job requirements.

Deadline-miss Ratio is the measure of the percentage of tasks when they miss their deadline of completion

Response time is the time taken from the time of request arrival to the time of start of execution.

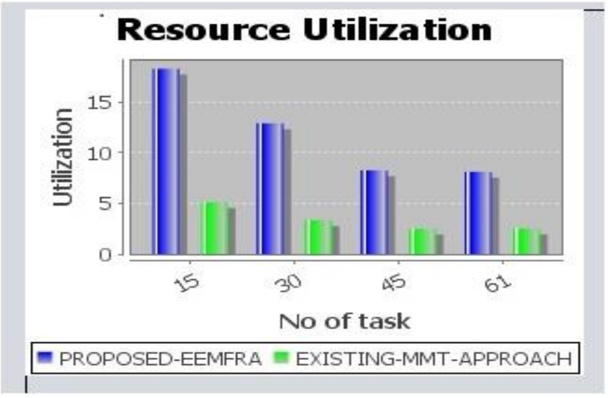

Fig. 6. Graphical Analysis of Resource Utilization

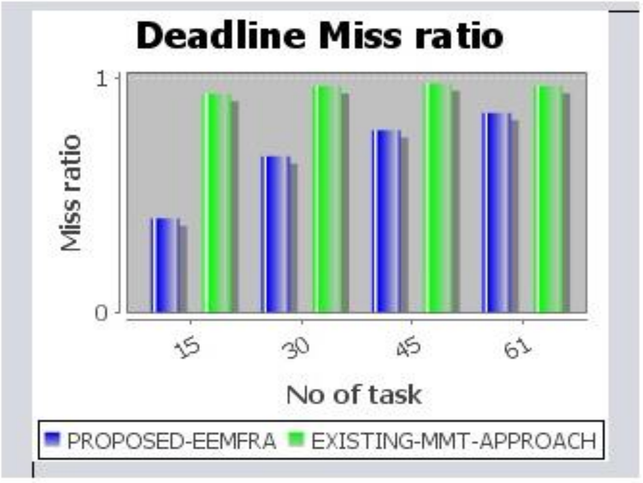

Fig. 7. Graphical Analysis of Deadline Miss Ratio

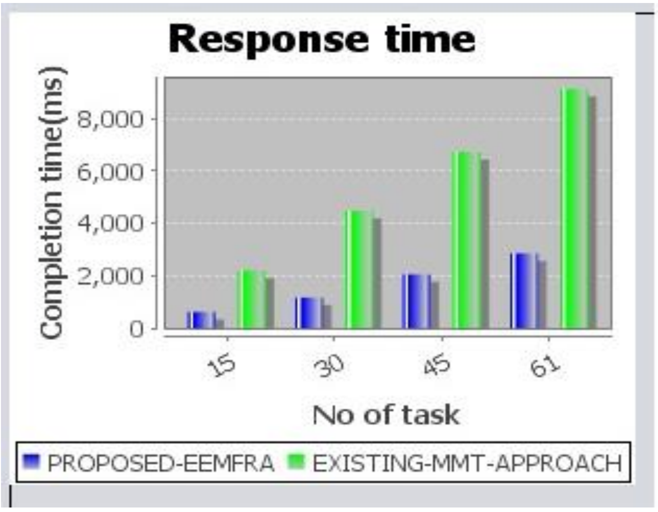

Fig. 8. Graphical Analysis of Response Time

Thus overall Host node energy consumption is the minimum energy consumed with maximum resource utilization.

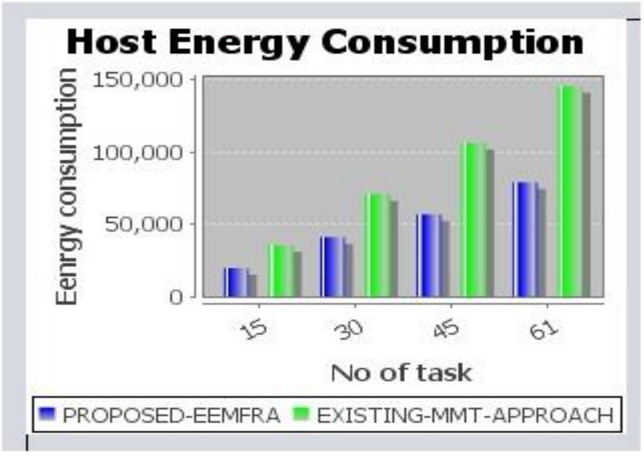

Fig. 9. Graphical Analysis of Host Energy Conumption

CONCLUSION

In this paper, an energy conscious RA is done based considering those only those hosts as the candidates for launching VMs to be migrated. Thus, resource utilization is improved by allocating the tasks to only a fewer nodes and controlling of hosts entering into OP or LP states with respect to dynamic CVs. The process of choosing the hosts during consolidation is modeled as using Fuzzy logic technique. It is also observed from the graphs that our proposed system has shown better performances when compared with the existing approach. We have used simulation with limited infrastructure for the work. In future, we would get improved results if advance technologies like machine learning and artificial intelligence are applied. Also, the anticipation of requests can be done which influences maximum resources usage with a minimized energy consumption.

REFERENCES

[1] Chen Q, Deng QN (2009) Cloud computing and its key techniques. Jr. Comput Appl 4:25–62

[2] Kaushar H, Ricchariya P, Motwani A (2014) Comparison of sla based energy efficient dynamic virtual machine consolidation algorithms. Int J Comput Appl 102:0975–8887

[3] Barroso, L.A., and U. Hölzle. 2009. The Datacenter as a Computer—An Introduction to the Design of Warehouse-Scale Machines. Synthesis Series on Computer Architecture. San Rafael, Calif.: Morgan & Claypool Publishers

[4] Prakash K, Edwin B (2013) Survey about power management techniques for high performance data centers in cloud environment. Int J Eng ResTechnol (IJERT) 2:3403–3407

[5] Wang L, Von Laszewski G, Dayal J, Wang F (2010) Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. In: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. IEEE Computer Society. pp 368–377

[6] Von Laszewski G, Wang L, Younge AJ, He X (2009) Power-aware scheduling of virtual machines in dvfs-enabled clusters. In: 2009 IEEE International Conference on Cluster Computing and Workshops. IEEE. pp 1–10

[7] Thakral D, Singh M (2014) Virtualization in cloud computing. Int J Comput Sci Mob Comput 3:1262–1273

[8] P. Xiaoa, Z. Hub, D. Liua, X. Zhanga, and X. Qua. Energy-efficiency enhanced virtual machine scheduling policy for mixed workloads in cloud environments. Computers & Electrical Engineering, 40(5):1650{1665, 2014

[9] Prakash K, Edwin B (2013) Survey about power management techniques for high performance data centers in cloud environment. Int J Eng Res Technol (IJERT) 2:3403–3407

[10] Beloglazov, A.; Buyya, R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr. Comput. Pract. Exp. 2012, 24, 1397–1420
[11] Beloglazov A, Buyya R (2010) Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. MGC, Bangalore, India. Copyright 2010 ACM 978-1-4503-0453-5/10/11. T. R. V
[12] Monil MAH, Rahman RM (2016) VM consolidation approach based on heuristics, fuzzy logic, and migration control. J Cloud Comput 5:8
[13] Farahnakian, F., Ashraf, A., Pahikkala, T., Liljeberg, P., Plosila, J., Porres, I., Tenhunen, H.: Using ant colony system to consolidate VMs for green cloud computing. IEEE Trans. Serv. Comput. 8(2), 187–198 (2015)
[14] A. Marcel, et al.: 'Thermal aware workload consolidation in cloud data centers', 'Book Thermal aware workload consolidation in cloud data centers' (2016, edn.), pp. 377-384
[15] Abdelsamea A, El-Moursy AA, Hemayed EE, Eldeeb H (2017) Virtual machine consolidation enhancement using hybrid regression algorithms.Egypt Inform J 18(3):161–170
[16] Moges FF, Abebe SL (2019) J Cloud Comput 8(1):2. https://doi.org/10.1186/s13677-019-0126-y
[17] Wang H, Tianfield H (2018) Energy-aware dynamic virtual machineconsolidation for cloud datacenters. IEEE Access 6:15259–15273. https://doi.org/10.1109/ACCESS.2018.2813541
[18] Yousefipour A, Rahmani AM, Jahanshahi M, Energy and cost-aware virtual machine consolidation in cloud computing. Softw: Pract Experience 48(10):1758–1774. https://doi.org/10.1002/spe.2585, https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2585
[19] Zadeh. L.A(1965) Fuzzy sets. International information and Control, 8, 338-353
[20] C. Lee, "Fuzzy Logic Controller- Parts i and ii": IEEE Trans. on Syst. Man Cybern., vol. 20, pp. 404-435, March/April 1990.
[21] D. Driankov, H. Hellendoorn and M. Reinfrank, "An Introduction to Fuzzy Control," Springer-Verlag, Berlin, 1993
[22] R.R. Yager and D.P. Filev, Essentials of Fuzzy Modeling and Control, John Wiley & Sons, New York, 1994
[23] P. Bonnisone, "Fuzzy logic Controllers: An Industrial reality", in Computational Intelligence:Initiating Life, IEEE press, 1994, pp.316-327.
[24] Bai Y, Wang D (2006) Fundamentals of fuzzy logic control - fuzzy sets, fuzzy rules and defuzzifications. Advanced fuzzy logic technologies in industrial applications. Springer, Berlin, pp 17–36.
[25] Beloglazov A., & Buyya R. (2010, May 17-20). Energy efficient allocation of virtual machines in cloud data centers. In Proceedings of the 10th IEEE/ ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid'2010), Melbourne, Victoria, Australia (pp. 577-578). doi:10.1109/ CCGRID. 2010
[26] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Journal of Software Practice & Experience, Vol. 41, No. 1, 2011, pp. 23-50. doi:10.1002/spe.995