JACET Journal of Advances in Computer Engineering and Technology, 1(3) 2015

A Near Optimal Approach in Choosing the Appropriate Physical Machines for Live Virtual Machines Migration in Cloud Computing

Seyedeh Roudabeh Hosseini[1], Sepideh Adabi[2], Reza Tavoli[3]

Received (2015-08-5)
Accepted (2015-10-30)

Abstract - **Migration of Virtual Machine (VM) is a critical challenge in cloud computing. The process to move VMs or applications from one Physical Machine (PM) to another is known as VM migration. In VM migration several issues should be considered. One of the major issues in VM migration problem is selecting an appropriate PM as a destination for a migrating VM. To face this issue, several approaches are proposed that focus on ranking potential destination PMs by addressing migration objectives. In this paper we propose a new hierarchal fuzzy logic system for ranking potential destination PMs for a migrating VM by considering following parameters: Performance efficiency, Communication cost between VMs, Power consumption, Workload, Temperature efficiency and Availability. Using hierarchal fuzzy logic systems which consider the mentioned six parameters which have great role in ranking of potential destination PMs for a migrating VM together, the accuracy of PMs ranking approach is increased, furthermore the number of fuzzy rules in the system are reduced, thereby reducing the computational time (which is critical in cloud environment). In our experiments, we compare our proposed approach that is named as (HFLSRPM: Hierarchal Fuzzy Logic Structure for Ranking potential destination PMs for a migrating VM) with AppAware algorithm in terms of communication cost and performance efficiency. The results demonstrate that by considering more effective parameters in the proposed PMs ranking approach, HFLSRPM outperforms AppAware algorithm.**

Index Terms - **Cloud computing, Hierarchal fuzzy logic structure, Virtual machine Migration.**

1- Department of Electronic & Computer Engineering, Institute for Higher Education Pouyandegan Danesh, Chalous, IRAN (roudabeh.hosseini@gmail.com)
2- Department of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, IRAN (Adabi.sepideh@ieee.org)
3- Department of Mathematics, Chalous Branch, Islamic Azad University, Chalous, IRAN (r.tavoli@gmail.com)

I. INTRODUCTION

Developments of virtualization and communication technologies have changed data center's design and operation in recent years [1] and there has been a strong tendency in development of data centers which applications have low dependencies to underlying infrastructure and can easily share resources. Virtual machine migration is one of the famous approaches to fulfill the cloud computing objectives.

Generally there are two types of migrations including: 1) live migration and 2) non-live migration. In VM live migration, VMs are moved from one PM to another PM while VMs are running [2]. In non-live migration, VM is stopped working in source PM, and when received all processor state, memory pages and disk data VM starts working in destination PM, from the last sates before migration [2].

Although there are many approaches in selecting an appropriate Physical Machine (PM) as a destination for a migrating VM, they have paid little attention to consider the combination affect of the PM ranking parameters include: Performance efficiency, Communication cost between VMs, Power consumption, Workload, Temperature efficiency, Availability together for increasing the accuracy of PM ranking approach. This motivating consideration provides the impetus for proposing a new approach named as HFLSRPM (Hierarchal Fuzzy Logic Structure for Ranking potential destination PMs for a migrating VM). HFLSRPM focuses on considering the most important PM ranking parameters including: performance efficiency, communication cost between VMs, power consumption, workload, temperature efficiency and availability together. Notions about PM ranking parameters that make numerical value of PM rank (i.e., PM_rank) are vague and uncertain to be expressed by crisp mathematical models. It is, however, often possible to describe the PM_rank by means of building fuzzy models. Two common sources of information for building fuzzy models are prior knowledge and data (process measurements). Real data in the field of cloud computing is rare and not available; hence it is prudent to construct fuzzy logic system for determining PM_rank by using knowledge of experts.

There is a direct relation between the number of fuzzy sets of input parameters of the system and the size of the fuzzy knowledge base [3]. As the number of fuzzy sets of input parameters increase, the number of rules increases exponentially. Obviously by considering six PM ranking parameters as the number of inputs into the fuzzy system we will face the mentioned problem. In this case limiting the number of inputs that the system use, is recommended. However, this may sacrifices the accuracy of the system. Another way is trimming the number of rues in the fuzzy knowledge base if it is known that some rules are never used. This may be time consuming (which cannot be tolerated in cloud computing) or even impossible. To face this problem Raju and ZHOU [4] suggested using a hierarchal fuzzy logic structure for such fuzzy logic systems. Their idea leads to reduce computational time and maintain systems robustness and efficiency. As a result, designing a hierarchal fuzzy logic structure is a suitable choice for our problem.

The remainder of the paper is structured as follows: Section 2 presents a brief overview of previous works. The proposed Hierarchal Fuzzy Logic Structure for Ranking potential destination PMs for a migrating VM (HFLSRPM) is described in details in section 3. The experimental results to study the performance of HFLSRPM are given in Section 4. Finally, conclusions and future works are described in section 5.

II. LITERATURE REVIEW

In this section we review the state-of-the-art VM migration approaches in cloud computing.

Many researches in context of VM migration has been done for managing resource usage in data centers to decrease costs and improve performance, efficiency and flexibility. For eliminating hotspots in data center, Sandpiper algorithm [5] was proposed. Sandpiper provides two monitoring strategies for collecting statistics including black-box and grey-box strategy. Black-box strategy, collect statistic from outside the VM and grey-box approach that access to OS-level statistics, resource usage of VMs and application resident within each VM and migrate overloaded VMs to less loaded servers that can satisfy VMs need.

Unbalanced temperature in data centers results higher cooling cost [1]. In [6] a multi-objective approach virtual machine management in data centers was proposed that improves

VM performance and temperature efficiency and reduce power consuming [6]. To detect overloaded server, a method has been proposed in [7] in which TOPSIS algorithm has been used to relocate VMs between clusters. The proposed method consists a control unit which receives PM information and sorts PMs from the highest rank to the lowest rank. The control unit checks the ranks and if it is higher than a predefined threshold, this means that the server is saturated and migration must be done. In next step, hotspot VMs is determined by some parameters.to avoid transferring large data and reduce cost, the VMs which have the lowest RAM utilization are chosen for migration. By migrating hotspot from overloaded PM to under loaded PM, the load distribution is done and response time is improved [7]. In [8] control architecture for VM migration to trade-off between performance and cost and power is proposed.

Optimization bandwidth usage is a primary goal in the data centers [9]. In this context AppAware is an evaluating approach for selecting the most appropriate PM to host VM in terms of minimizing the traffic of data center network. The main aim of AppAware is to put dependent VMs in close proximity to reduce total traffic in data center physical network. This algorithm takes into account inter-VM dependencies and underlying network topology into host selection. AppAware migrates an overloaded VM to a PM based on a migration impact factor and required resources [10].

In [11] the authors study offline and online versions of the four versions of the Virtual Machine Assignment problem. In proposed model VM assignment is based on a CPU requirement and shows that the optimal load of a given PM is a function only of the fixed cost of being active and the exponential rate of power increases on the load. The goal of model is optimizing the power consumed by all the PMs.

Tao et al. [12] proposed triple-objective comprehensive model for solving dynamic migration of VMs which uses a binary graph matching-based bucket-code learning algorithm (BGM-BLA) for evaluating the candidate solutions. The model goal is reducing the energy consumption and communication cost while reducing migration cost.

However, although there are a large number of works in the field of VM migration in cloud computing, to the best of authors' knowledge they were not considered the effect of important parameters in VM migration approach in names performance efficiency, communication cost between VMs, power consumption, workload, temperature efficiency and availability together.

III. HFLSRPM: HIERARCHAL FUZZY LOGIC STRUCTURE FOR RANKING POTENTIAL DESTINATION PMS FOR A MIGRATING VM

The proposed HFLSRPM has two layers. In the first layer of HFLSRPM three aspects for calculating PM_rank are defined: (1) calculating PM_rank based on serving conditions, (2) calculating PM_rank based on communication cost and (3) calculating PM_rank based on power consuming. As shown in Fig.1 the first layer of HFLSRPM composes of two types of fuzzy decision controller: Fuzzy PM_Serving_Condition determinator and Fuzzy PM_Power_Consuming determinator which are designed to determine the numerical values of PM_rank based on serving conditions and PM_rank based on power consuming respectively.

The second layer of HFLSRPM is composed of a fuzzy decision controller, Total_PM_Rank, determinator which is designed to determine the total values of PM_rank based on a) the output of Fuzzy PM_Serving_Condition determinator, b) the output of Fuzzy PM_Power_Consuming determinator and c) communication cost.

A fuzzy decision controller is composed of (1) input and output variables, which are determined based on knowledge of experts; (2) a fuzzification interface (FI), which has the effect of transforming crisp data into fuzzy sets; (3) a fuzzy rule base (RB), in which a set of fuzzy rules is determined; (4) a fuzzy negotiation decision making logic (DML), that uses them together with the RB to make inference by means of a reasoning method; and (5) a defuzzification interface (DFI), that translates the fuzzy rule action thus obtained to a real action using a defuzzification method.

Following the five components of each part of PM_Serving_Condition determinator and PM_Power_Consuming determinator of the first layer of HFLSRPM and Total_PM_Rank determinator of the second layer of HFLSRPM are discussed.

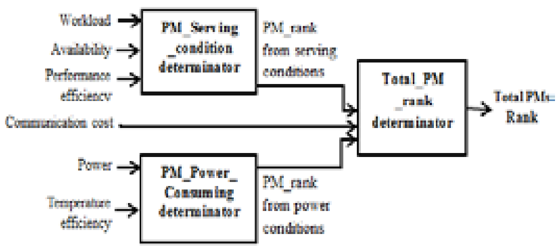

Fig1. An abstract view of HFLSRPM.

III-1. PM_Serving_Condition determinator

The PM_Serving_Condition determinator ranks each PM in terms of PMs servicing condition. The PM with the highest rank in terms of servicing condition is the best destination for a migrating VM.

The inputs of PM_Serving_Condition determinator are : i) Availability, ii) Workload, and iii) Performance efficiency.

Availability: is a percentage of time that a customer can access to the service [13]. For selecting a PM as a destination of a migrating VM, it is important to know if the PM is available on that time or not. In other words, investigate if a candidate destination PM has sufficient capacity for supporting new VM and can satisfy its requirements [1]. Availability of a i'th PM is determined as (1) [1]:

$$PM_i\_\text{Availability} = \frac{T_t - T_n}{T_t} \qquad (1)$$

where Tt is total service time and Tn is total time for which service was not available. According to (1), when tends to 1 the availability of a PM increased. Obviously a PM with the highest value is the best destination for a migrating VM in term of PM availability.

Workload: is a discrete capability or amount of work you'd like to run on a Cloud instance [14]. A key issue in cloud computing environments is to maximize profit by accepting all incoming requests and to minimize SLA (Service Level Agreement) violation. Achieving these goals highly depends on how available resources are used. The violation of SLA is likely to increase if the workload of a PM increases. So in choosing a destination PM for a migrating VM the current workload of the candidate destination PMs should be considered. In practice three criteria include Memory utilization, CPU utilization and network utilization capture the load of a physical server. Workload of a i'th PM is determined as (2) [15]:

$$PM_i\_Workload = \frac{1}{1-U_{mem}} * \frac{1}{1-U_{CPU}} * \frac{1}{1-U_{net}} \qquad (2)$$

where Umem is memory utilization, Ucpu is CPU utilization, and Unet is network utilization [15]. According to (2), when tends to 1 the workload of a PM increased. Obviously a PM with the lowest value is the best destination for a migrating VM in term of PM workload.

Performance efficiency: represents the amount of use of resource of different types. To avoid resource contention, the efficiency decreases rapidly when the usage of one or more of resources increased the maximum allowed [6]. VMs should migrate to the PMs that have better performance efficiency. Equation (3) defines the performance efficiency of i'th PM [6].

$$\begin{aligned}
&PM_i_Eff_i(C) = \min\left(PM_i_Eff_i(CPU), PM_i_Eff_i(IO), PM_i_Eff_i(Net)\right)\\
&PM_i_Eff_i(CPU) = 1-\left(\frac{CPU_i - CPU_{low}}{CPU_{high} - CPU_{low}}\right)^m\\
&PM_i_Eff(IO) = 1-\left(\frac{IO_i - IO_{high}}{IO_{high} - IO_{low}}\right)^m\\
&PM_i_Eff(Net) = 1-\left(\frac{Net_i - Net_{high}}{Net_{high} - Net_{low}}\right)^m
\end{aligned} \qquad (3)$$

where CPUi is CPU usage (%), CPUlow is CPU usage of idle PM (0%), CPUhigh is CPU usage of overloaded PM (100%), IOi is disk utilization (%), Olow is IO usage of idle PM (0%), IOhigh is IO usage of overloaded PM (100%), Neti is the network IO usage of PMi, Nethigh is the highest network IO usage (20 M bytes/sec), Netlow is the lowest network IO usage (0) and m is exponent (set to 3 in implementation) [6].

According to (3), when tends to 1 the performance efficiency of a PM decreased. Obviously a PM with the lowest value is the best destination for a migrating VM in term of PM performance efficiency.

The PM_Serving_Condition determinator's input and output variables have three fuzzy values: {L(low), M(moderate), H(high)}. The membership functions of PM_Serving_

Condition determinator are shown in Fig.2, Fig.3, Fig.4 and, Fig.5 .For the sake of less computational complexity we apply linear (triangular) membership function instead of non-linear shapes for each PM_Serving_Condition determinator's input and output parameter. In addition, the weighted average method [16] is used for defuzzification. If the PM_rank from serving conditions perspective tends to become 1, the chance of selecting a PM as a destination for a migrating VM from servicing condition perspective should be increased.

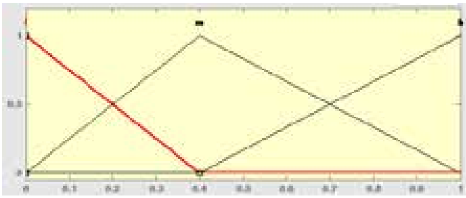

Fig 2. Availability membership function

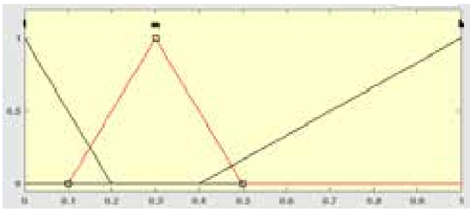

Fig 3. Performance efficiency membership function

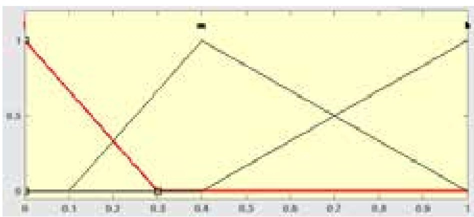

Fig 4. Workload membership function

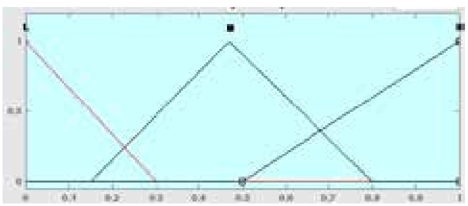

Fig 5. Output membership function

TABLE I illustrates the rule set for PM_Serving_Condition determinator which is determined based on knowledge of experts. For more clarification, an example is provided. The rule in the 6 th row of TABLE I interpreted as:

If *(Performance efficiency is Low)* ***And*** *(Availability is High)* ***And*** *(Workload is Low)* ***Then*** *(Output is High)*

The rank of PM from PM_Servicing_Condition perspective is set to High due to workload is Low thus PM being capable to fulfill the migrating VM's requirements with low chance of violating other serving VMs' SLA and availability is High thus the chance of readiness of PM to serve the migrated VM is high and finally performance efficiency is Low thus according to details of (3) the PM has the best situation from performance efficiency perspective.

TABLE I
THE RULE SET OF PM_SERVING_CONDITION DETERMINATOR

Rule #	Input metrics			Output
	Performance efficiency	Availability	Workload	
1	H ∨ M	L	L	L
2	L	L	L	M
3	M ∨ L	M	L	M
4	H	H ∨ M	L	L
5	M	H	L	M
6	L	H	L	H
7	M ∨ H	L	M	L
8	H	M	M	L
9	M	M	M	M
10	L	M ∨ H	M	M
11	H	H	M	L
12	M	H	M	L
13	H	L	H	L
14	L	L	H ∨ M	L
15	H ∨ M	M ∨ L	H	L
16	M ∨ L	M	H	M
17	L ∨ M ∨ H	H	H	L

III-2. PM_Power_Consuming determinator

The PM_Power_Consuming determinator ranks each PM in terms of PMs power consuming. The PM with the highest rank in terms of power consumingis a better destination for a migrating VM.

The inputs of PM_Power_Consuming determinator are: i) *Power consumption* and ii) *Temperature efficiency*.

i) Power consumption: The power utility is a function of resource utilization by a PM in a time interval [17]. A PM with Lower power utilization should be selected as a destination for a migrating VM. Equation (4) shows the Power utility function [17].

$$PM_i\_power = P_{idle} + P_{cpu}\frac{U_{cpu}}{C_{cpu}} + P_{disk}\frac{U_{disk}}{C_{disk}} \quad (4)$$

where P_{cpu} is the maximum dynamic power usage of the CPU, U_{CPU} is CPU consumption as a percentage of the total CPU capacity [%], U_{disk} is disk usage as a percentage of the total bandwidth capacity [%], C_{disk} is total disk bandwidth capacity and P_{idle} is power utilization by a PM when it is idle [17]. According to (4) if PM_i_power value tends to 1, the power utilization of the PM increased. Obviously a PM with the highest value is the best destination for a migrating VM in term of power utilization.

ii) Temperature: by increasing CPU temperature, the cost of cooling data center will be increased furthermore the performance of PM is affected. To reduce data center cooling cost and power consuming, it is essential to select a PM with higher temperature efficiency as a destination for a migrating VM. The Temperature efficiency of i'th PM is determined as (5) [6]:

$$PM_i\_Eff_i(T) = 1 - \left(\frac{T_i - T_{low}}{T_{high} - T_{low}}\right)^m \quad (5)$$

Where T_i is the temperature of PM_i, T_{Low} is the temperature for an idle PM (15˚C), T_{high} is the temperature for overloaded PM (55˚C) and *m* is degree which is set to 3 in implementation [6]. According to (5) if value PM_i_Eff tends to 1, the temperature efficiency of the PM decreased. Obviously a PM with the highest PM_i_Eff value is the best destination for a migrating VM in term of temperature efficiency.

The linguistic terms for the inputs and output of PM_Power_Consuming determinator are as same as the linguistic terms for the inputs and output of PM_Serving_Condition determinator. The membership functions of Temperature efficiency and output are as same as Fig.3 and Fig.5 respectively and membership functions of power is shown in Fig.6.

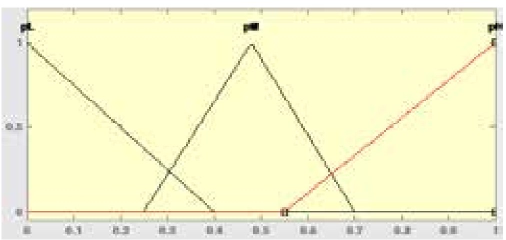

Fig 6. Power membership function

TABLE II illustrates the rule set for PM_Power_Consuming determinator which is determined based on knowledge of experts. For more clarification, an example is provided. The second rule of TABLE II interpreted as:

***If** (Temperature efficiency is Moderate or High) **And** (Power consuming is Low) **Then** (Output is High)*

In this example the rank of PM from PM_Power_Consuming determinator perspective is set to High cause temperature efficiency is High or Moderate thus cooling cost will be decreased and power consuming is Low thus the PM has the best situation from power consuming perspective.

TABLE II
THE RULE SET OF PM_POWER_CONSUMING DETERMINATOR

Rule #	Input metrics		Output
	Temperature efficiency	Power consuming	
1	L	L	M
2	M ∨ H	L	H
3	L	M	L
4	M ∨ H	M	M
5	L ∨ M	H	L
6	H	H	M

III-3. Total_PM_Rank determinator

III-4. The Total_PM_Rank determinator ranks each PM to find a near_optimaldestination for a migrating VM. PM with the highest PM_rank will be chosen as a destination for a migrating VM.

The inputs of Total_PM_Rank determinator are: i) *output of Fuzzy PM_Serving_condition determinator,* ii) *output of Fuzzy PM_Power_Consuming determinator,* and iii) *communication cost.* Follows the communication cost parameter is discussed.

i) *Communication cost:* the network communication cost is the time that is taken to communicate and swap data between VM_i and VM_j [18]. By moving dependent VMs which exchange a large volume of network traffic closer to each other, the network communication cost will be reduce. The VM communication cost in VL2 and Tree topology are determined in (6) and (7) [20] respectively:

$$C_{ij}^{VL2} = \begin{cases} 0 & if\ i = j \\ 1 & if \left\lfloor \frac{i}{p_0} \right\rfloor = \left\lfloor \frac{j}{p_0} \right\rfloor \\ 5 & if \left\lfloor \frac{i}{p_0} \right\rfloor \neq \left\lfloor \frac{j}{p_0} \right\rfloor \end{cases} \quad (6)$$

$$C_{ij}^{Tree} = \begin{cases} 0 & if\ i = j \\ 1 & if \left\lfloor \frac{i}{p_0} \right\rfloor = \left\lfloor \frac{j}{p_0} \right\rfloor \\ 3 & if \left\lfloor \frac{i}{p_0} \right\rfloor \neq \left\lfloor \frac{j}{p_0} \right\rfloor \cap \left\lfloor \frac{i}{p_0 p_1} \right\rfloor = \left\lfloor \frac{j}{p_0 p_1} \right\rfloor \\ 5 & if \left\lfloor \frac{i}{p_0 p_1} \right\rfloor \neq \left\lfloor \frac{j}{p_0 p_1} \right\rfloor \end{cases} \quad (7)$$

Where P_0 is the fan-out of the access switch and P_1 is the fan-out of the aggregation switch [20]. One of VL2 advantage is that VL2 can be easily implemented with low cost [19]. In VL2 Topology, the cost is a function of fan-out of the access switch (P_0) and can be calculated as (6) [20]. In (7), the cost between two VMs is a function of access switches (P_0) fan-out as well as the fan-out of the aggregation ones (P_1) [20].

If the result value is closer to 1, PM has more communication cost and if the result value is closer to zero it means that the PM is more suitable and has less communication cost. When C_{ij}^{VL2} (respectively, C_{ij}^{Tree}) tends to 1 the PM has more communication cost. Obviously a PM with the lowest C_{ij}^{VL2} (respectively, C_{ij}^{Tree}) value is the best destination for a migrating VM in term of communication cost.

The linguistic terms for the inputs and output of Total_PM_Rank determinator are as same as the linguistic terms for the inputs and output of PM_Serving_Condition determinator. Membership functions of communication cost is as same as Fig.3. The membership functions of the two inputs include: output of Fuzzy PM_Serving_condition determinator and output of Fuzzy PM_Power_Consuming determinator are same as Fig.6. The membership function of output of Total_PM_Rank determinator is shown in Fig.7.

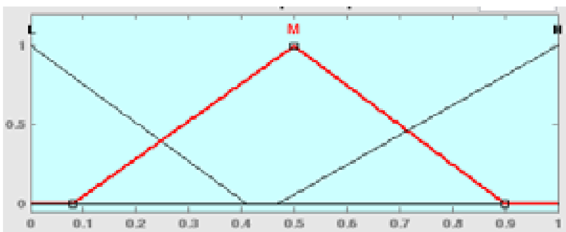

Fig 7. The membership functions of output of Total_PM_Rank determinator

TABLE III illustrates the rule set for Total_PM_Rank determinator which is determined based on knowledge of experts. For more clarification, an example is provided. The rule in the 10th row of TABLE III interpreted as:

If *(Power consuming state is Moderate or High)* ***And*** *(Communication cost is Low)* ***And*** *(PM service state is High)* ***Then*** *(Output is High)*

considering the VM migration goals which are shown in our paper (i.e., reducing cost and improving performance efficiency), one can understand that PM is an ideal host for the migrating VM if power consuming state of a PM is High, PM's servicing state is High and finally communication cost is Low.

TABLE III
The Rule Set of Total_PM_Rank Determinator

Rule #	Input metrics			Output
	PM power consuming state	Communication cost	PM service state	
1	L ∨ M	L	L	L
2	H	L ∨ M	L	M
3	L ∨ M	M ∨ H	L	L
4	H	H ∨ L	L ∨ M	L
5	L ∨ M	L	M	M
6	L ∨ M	M	M	L
7	H	M	M ∨ H	M
8	L	H	M	L
9	L	L	H	M
10	M ∨ H	L	H	H
11	M	M	H	M
12	L ∨ M ∨ H	H ∨ M	H	L

IV. EXPERIMENTAL RESULTS

A) Experimental setting

A testbed is developed in Matlab program to evaluate the performance of the proposed HFLSRPM. Two data center topologies in names VL2 and Tree are considered. All the input parameters required for setting simulation testbed and their possible values are shown in TABLE IV. In our simulation we run 100 scenarios for small topologies and 288 scenarios for large ones.

TABLE IV
Simulation Parameters

Simulation parameters	Quantity domain	
VMs dependencies	Uniforms distribution 1-3	
Fraction of overloaded VMs	0.8,0.4.0.2	
Architecture(PM)	[10]Tree, VL2	
# of VMs	Large topology	Small topology
	[20]-240[10]	5-12[10]
# of PMs	100[10]	7-10[10]

A) Performance measure

A series of experiments were carried out to compare the performance of HFLSRPM with AppAware algorithm [10] in terms of *communication cost* and *performance efficiency.*

Fig.8 and Fig.11 show the average Performance efficiency in large and small topology respectively. In these figures we can notice that HFLSRPM outperforms AppAware algorithm in terms of average Performance efficiency in both VL2 and Tree topologies because considering Workload, Availability, Performance and Temperature efficiency criteria, affect Performance efficiency. Fig.9 and Fig.10 show average communication cost in large and small topology respectively. From Fig.9 and Fig.10 it can be observed that HFLSRPM outperforms AppAware algorithm in terms of average communication cost in both VL2 and Tree topologies because considering Workload, Temperature efficiency, Power and Communication cost criteria affect migration cost.

These results confirm that the consideration of the average combinatorial effect of the six parameters (Performance efficiency, Communication cost between VMs, Power consumption, Workload, Temperature efficiency and Availability) by the proposed system, the accuracy of HFLSRPM increases which leads to better results in both VL2 and Tree topologies.

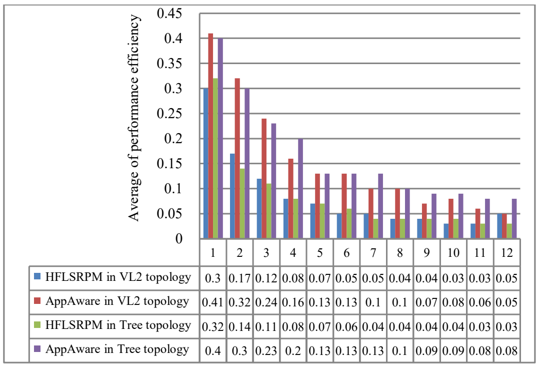

Fig8. Comparison of HFLSRPM and AppAware in term of average of performance efficiency in large topology

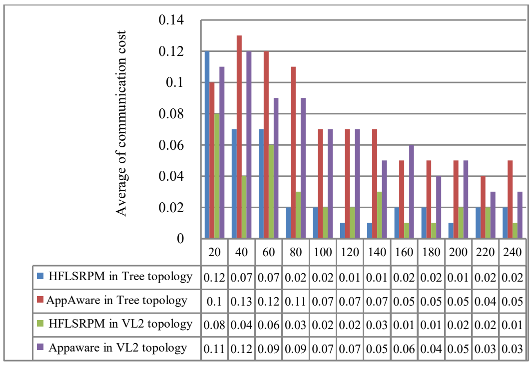

Fig9. Comparison of HFLSRPM approach and AppAware in term of communication cost in large topology

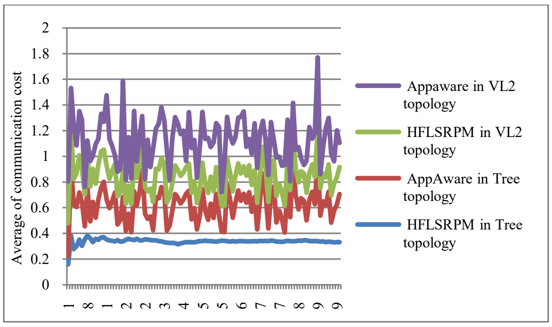

Fig 10.Comparison of HFLSRPM approach and AppAware in term of communication cost in small topology

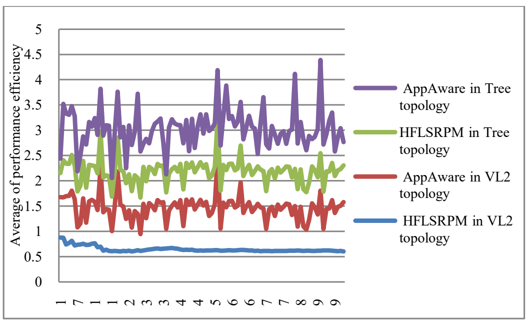

Fig11. Comparison of HFLSRPM and AppAware in term of average of performance efficiency in small topology

V. CONCLUSION

In this paper, we proposed a new Hierarchal Fuzzy Logic Structure to select an appropriate destination PM for a migrating VM (HFLSRPM) by considering two objectives: 1) reduce Communication cost and 2) improve the Performance efficiency of PMs. The HFLSRPM has two layers. In the first layer of HFLSRPM two fuzzy inference systems, PM_Serving_Condition determinator and PM_Power_Consuming determinator, are designed for ranking potential destination PMs for a migrating VM from PM's servicing condition and power consuming perspectives respectively. Availability, workload and Performance efficiency are considered as inputs for PM_Serving_Condition determinator whereas power consumption and Temperature efficiency are considered as inputs for PM_ Power_Consuming determinator. In the second layer of HFLSRPM the total rank of potential destination PMs for a migrating VM is determined by designing fuzzy inference systems named Total_PM_Rank determinator, which consider the outputs of PM_Power_Consuming determinator and PM_Serving_Condition determinator besides Communication cost as inputs.

Experimental results obtained from the simulations show that HFLSRPM has lower Communication cost and achieves higher average performance efficiency than AppAware algorithm by increasing the accuracy of PM ranking process due to combine the effect of the mentioned six parameters.

In future works we will work on two challenges: (1) re-design HFLSRPM that can be applied in intercloud migration problem and 2) develop HFLSRPM by combining fuzzy logic and genetic algorithm.

References

[1] R. Boutaba, Q. Zhang and M. F. Zhani, "Virtual Machine Migration in Cloud Computing Environments. Benefits, Challenges, and Approaches", IGI Global, 2013, pp. 383-408.

[2] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, and A. Warfield, "Live migration of virtual machines", In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, vol.2 , 2005, pp. 273-286.

[3] M. Mohammadian., "Designing Customized Hierarchical Fuzzy Logic Systems For Modelling and Prediction", 4thAsian-Pacific Conference on Simulated Evolution and Learning, pp.18-22, 2002, Singapore.

[4] G. V. S.Raju, J. Zhou., "Adaptive Hierarchical Fuzzy Controller", IEEE Transactions on Systems, Man & Cybernetics, vol 23, no.4,1993. pp. 973-980.

[5] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper. Black-box and gray-box resource management for virtual machines", Computer Networks, vol.53, no.17, 2009, pp. 2923-2938.

[6] J.Xu, andJ. Fortes, "A multi-objective approach to virtual machine management in datacenters", In Proceedings of the 8th ACM international conference on Autonomic computing, 2010, pp. 225–234.

[7] M.Tarighi, S. A. Motamedi and S.Sharifian, "A new model for virtual machine migration in virtualized cluster server based on Fuzzy Decision Making", arXiv preprint arXiv.1002.33292010.

[8] G. Jung, M. A.Hiltunen, K. R. Joshi, R. D.Schlichting, and C.Pu, "Mistral. Dynamically managing power, performance, and adaptation cost in cloud infrastructures", In Proceedings of the IEEE International Conference on Distributed Computing Systems ICDCS, 2010, pp.62–73.

[9] H. Liu, H. Jin, C. Z.Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines", Cluster computing, vol.16, no.2, 2013, pp.249-264.

[10] V. Shrivastava, P.Zerfos, K.W.Lee, H.Jamjoom, Y.H. Liu, and S. Banerjee, "Application-aware virtual machine migration in data centers", In Proceedings of IEEE INFOCOM, 2011, pp. 66 -70.

[11] C.Thraves and L. Wang, "Power-Efficient Assignment of Virtual Machines to Physical Machines. In Adaptive Resource Management and Scheduling for Cloud Computing", 2014, In first International Workshop, ARMS-CC 2014, held in Conjunction with ACM Symposium on Principles of Distributed Computing, PODC 2014, Paris, France, July 15, 2014, Revised Selected Papers, vol. 8907, pp. 71.

[12] F. Tao, C.Li,T. Liao and Y.Laili, "BGM-BLA: a new algorithm for dynamic migration of virtual machines in cloud computing", 2015.

[13] Garg, S. K., Versteeg, S., and Buyya, R., "A framework for ranking of cloud computing services", Future Generation Computer Systems, vol.29, no.4, 2013, pp.1012-1023.

[14] A. Burkimsher, I. Bate and L. S. Indrusiak, “A survey of scheduling metrics and an improved ordering policy for list schedulers operanking on workloads with dependencies and a wide variation in execution times”, Future Generation Computer Systems,vol. 29, no.8, 2013, pp. 2009-2025.

[15] A. Sallam, K. Li, A. Ouyang and Zh. Li, “Proactive workload management in dynamic virtualized environments”, In Journal of Computer and system Science, 2014, vol.80, pp.1504-1517.

[16] T. J. Ross, “Fuzzy logic with engineering applications”. McGraw-Hill, New York, 1995.

[17] D. Minarolli, and B. Freisleben, “Distributed Resource Allocation to Virtual Machines via Artificial Neural Networks”, In Parallel, Distributed and Network-Based Processing PDP, 2014 22nd Euromicro International Conference on, 2014, pp. 490-499.

[18] J. Sonnek, J.Greensky, R.Reutiman, andA. Chandra, “Starling. Minimizing communication overhead in virtualized computing platforms using decentralized affinity-aware migration”, In Parallel Processing ICPP, 2010 39th International Conference on, 2010, September, pp. 228-237.

[19] A. Hammadi, and L. Mhamdi, “A survey on architectures and energy efficiency in Data Center Networks”, Computer Communications, vol. 40, 2014, pp.1-21.

[20] X. Meng, V. Pappas and L. Zhang, “Improving the scalability of data center networks with traffic-aware virtual machine placement”, In INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1-9