Libra: A Load Balancing System for Virtualized Data Center

Mya Sithu
University of Computer Studies, Yangon
ms.myasithu@gmail.com

Abstract

In the virtualized data center, load imbalances and the deviation of service level agreements of the application can occur when the load changes dynamically over time. This paper presents Libra, four- level resource management system which consists of application level controllers (ALC), virtual machine level controllers (VLC), host level controllers (HLC) and a center level controller (CLC). ALC, VLC and HLC automates the task of monitoring, detecting load unbalancing, resource managing.CU determines a new mapping of physical and virtual resources and initiating the necessary migrations. The four-level resource controllers use rule-based fuzzy logic for resource management and migration decisions.

Keywords: *Virtualized Data Center, Service Level Agreement, Load Balancing, Fuzzy Modeling*

1. Introduction

The virtualized data center contains physical and virtual servers which serve a variety of services including web services, file services etc. Virtualized data center provides numerous benefits. It enables application isolation since malicious or greedy applications cannot impact other applications co-located on the same physical server. It enables server consolidation and provides better multiplexing of data center resources across applications. Perhaps the biggest advantage of employing virtualization is the ability to flexibly remaps physical resources to virtual servers in order to handle workload dynamics. A workload increase can be handled by increasing the resources allocated to a virtual server, if idle resources are available on the physical server, or by simply migrating the virtual server to a less loaded physical server. Migration is transparent to the applications and all modern virtual machines support this capability [2, 9].

Data center, as resource providers are expected to deliver on performance guarantees while optimizing resource utilization to reduce cost [6]. Each application in an I/T environment is usually associated with a service level agreement (SLA), which in the simplest case, consists of response time and throughput requirements. During run time, if the SLA of an application is violated, it is often because of factors such as high CPU utilization and high memory usage of the server where it is hosted. This paper proposes these two problem issues, load unbalancing and SLA violation.

Manually- initiated migration cannot respond to the sudden changes of workloads. When the needs of considering the mix of CPU, network and memory intensive OS appears, the manual migrations of virtual machines are complex jobs for the system administrator. To perform required migrations efficiently and effectively, the auto-migration policies are needed for the virtualized data center. In that case, the auto-controller has to choose which virtual machine is overloaded and which one is the destination server for the overloaded machine.

In this paper, the proposed system, Libra is presented. The Libra is an auto-migration controller system for the virtualized data center. It comprises with four main components. The first part is application level controllers (ALC) which monitor and detect the SLA agreements of the running applications on virtual machines. According to the responses of ALC, virtual machine level controllers (VLC) try to manage resources that are allocated to applications. If it has not enough resources that can be allocated to running applications, it asks for additional resources from the physical machine which it is hosted. HLC can give more resources if it has ideal resources. But ideal resources are running out in the host, HLC requests center level controller to migrate loads from the overloaded physical machines to underutilized ones. As the data center contains various heterogeneous servers, many uncertainty parameters occur when the center level controller tries to make migration decisions. The best theory for those varies and uncertainties are fuzzy logic control theory. In this system, ALC, VLC, HLC uses same fuzzy sets and membership functions to manage resource management and the center level controller is also based on the fuzzy inference engine with different fuzzy sets.

The rest of the paper is structured as follows. Section 2 presents the related work of this paper. Section 3 is about the background theory. Section 4 will explain about the operation of Libra and how Libra is comprised of. It will be followed by conclusion and future work in section 7.

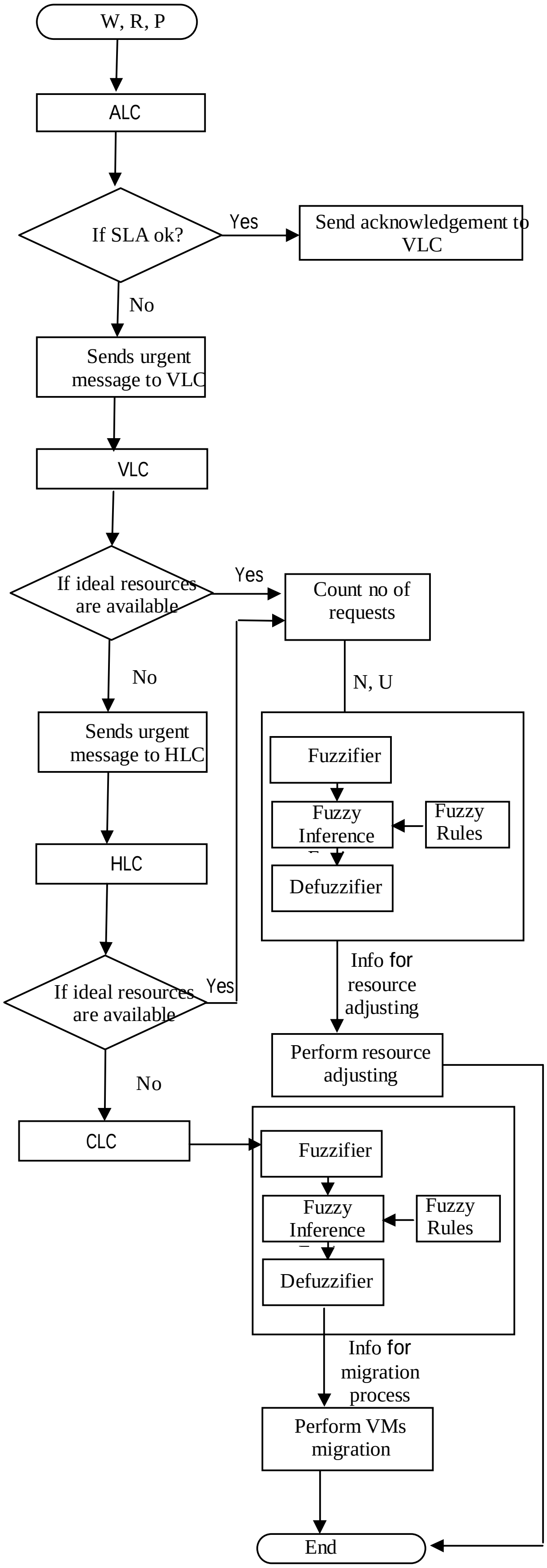

Figure 1. System flowchart of four-level controllers

2. Related Work

This section presents the related work that is considered for resource management of virtualized data center and virtual machine migrations. Sandpiper [12] automates the task of monitoring and detecting hotspots, determining a new mapping of physical to virtual resources and initiating the necessary migrations. Sandpiper implements a black-box approach that is fully OS- and application-agnostic and a gray-box approach that exploits OS- and application-level statistics. But Sandpiper did not take into account the complicated and uncertain relationship between the system's parameters. In [5], they implemented Xen API. Using the metrics of each VM, the system give a calculation of redundancy which tells how many of the physical hosts can go down before there is no more free resources to keep all the VMs up. In their system, they did not consider the complicated relationships between the system parameters. P.Paddala presents their system as automated control of multiple virtualized resources. Auto control is a resource control system that automatically adapts to dynamic workload changes to achieve application service level objectives (SLO). The model estimator captures the complex relationship between application performance and resource allocations, while the MIMO controller allocates the right amount of multiple virtualized resources to achieve application SLOs. In online model estimator, they used adaptive modeling approach to capture the complex behavior of enterprise applications. The first layer of MIMO controller consists of a set of application controllers and it uses the estimating models and a feedback based approach to achieve application SLO. The second layer of MIMO is comprised of a set of node controllers to detect resource bottlenecks on the shared nodes. In their system, they only used CPU and disk I/O as the resources, and application throughput or average response time as the performance metric [11].

Jing Xu proposed a two level resource management system to dynamically allocate resources to individual virtual containers. It uses local controllers at the virtual container level and a global controller at the resource pool level. The local controller is based on fuzzy modeling approach and fuzzy prediction to deal with the complexity and uncertainties of dynamically changing workloads and the global controller is based on the profit model to maximize the data center's profit. In their implementation, they only chose the CPU intensive applications and response time for the input and output pairs of the fuzzy rules. Although they consider the quantity of physical resources allocated to each virtual container, they did not take into account which physical server is assigned to host the virtual container at each allocation period [6]. VMware's Distributed Resource Scheduler solves the CPU and memory pressure by performing load balancing dynamically. But VMware's DRS cannot utilize application logs to have better placement decisions. Moreover, DRS is only efficient for homogeneous virtualized environment [13].

In [10], M.Tarighi presents a method to migrate VMs between cluster nodes using TOPSIS algorithm to find the most loaded server. TOPSIS algorithm is based on positive ideal solution and negative ideal solution. The positive ideal solution is a solution that maximizes the benefit criteria/attributes and minimizes the cost criteria/attributes, whereas the negative ideal solution maximizes the cost criteria/attributes and minimizes the benefit criteria/attributes. In their system, they combine the TOPSIS algorithm and fuzzy theory. In their first level of implementation, they order the physical servers and decide which the most overloaded ones are. Then the second level of operation starts by ordering the virtual machines. After completing the ordering of two steps, then migration decisions are made to move the most overloaded virtual machines from the most overloaded physical machines to the least overloaded machine. But they did not consider the historic trends for the future estimation and application service level agreement at run time.

3. Background Theory

Fuzzy logic is a tool to deal with uncertain, imprecise, or qualitative decision-making problems. Unlike Boolean logic, where an element x either belongs or does not belong to a set A, in fuzzy logic the membership of x in a fuzzy set F has a degree value (called fuzzy value) in a continuous interval between 0 and 1 representing the extent to which x belongs to F. Fuzzy sets are defined by membership functions that map set elements into the interval [0, 1]. One of the most important applications of fuzzy logic is the design of fuzzy rule-based systems. These systems use "IF-THEN" rules (also called fuzzy rules) whose antecedents and consequents use fuzzy-logic statements to represent the knowledge or control strategies of the system. The collection of fuzzy rules is called a rule base. There are many approaches to the construction of fuzzy rules, for example, by capturing expert experience or system operator's control actions. The approach taken for the design of our system is to learn fuzzy rules using online monitoring information, making it a so-called self-organizing fuzzy system.

The process of formulating the mapping from inputs to outputs using fuzzy rules is called the fuzzy inference (FIS) mechanism. Since fuzzy rules use fuzzy sets and their associated membership functions to describe system variables, two functions are necessary for translating between numeric values and fuzzy values. The process of translating input values into one or more fuzzy sets is called fuzzification. Defuzzification is the inverse transformation which derives a single numeric value that best represents the inferred fuzzy values of the output variable.

4. Resource Management for Load Balancing

Figure 1 show how Libra runs in the virtualized data center. ALCs sit in every application and monitor the workload performance. They send additional resource requests to upper level VLCs if their current resource usage is not sufficient for the incoming workloads. VLCs solve the requests from the ALCs by adjusting the resources it own. HLCs perform their operation by managing the physical resources if VLCs cannot handle the requests of ALCs. CLC performs the final level operation by giving information for migration process.

4.1 Four Level Controllers

The four-level controllers of Libra perform their operations cooperatively to achieve effective migration decisions.

4.1.1 Application Level Controller

As shown in the above figure, at every specified period, ALC calculates its state info according to the predefined metric. The predefined metric consists of three vectors (*W*,*R*,*P*). In this three vectors notation, W stands for current workload, R stands for the current resource usage and P means required performance. If the current workload and current available resource is enough to get the required performance, ALC will send positive acknowledge message to VLC. If there is something difference between the available resource and required performance, ALC will inform VLC that how many additional resources are needed. There may be a case that the application gets more resources than enough to perform service. In that occasion, ALC also considers the recent history. If there may be a service peak to use all the resources that it got, it sends acknowledge to VLC. If there may not be a condition for such a peak, ALC sends a message of how many resources can be reduced. Based on the information collected from the ALC, VLC carries on its operation.

4.1.2 Virtual Machine Level Controller

As we have seen in section 4.1.1, VLC performs according to the collected information at

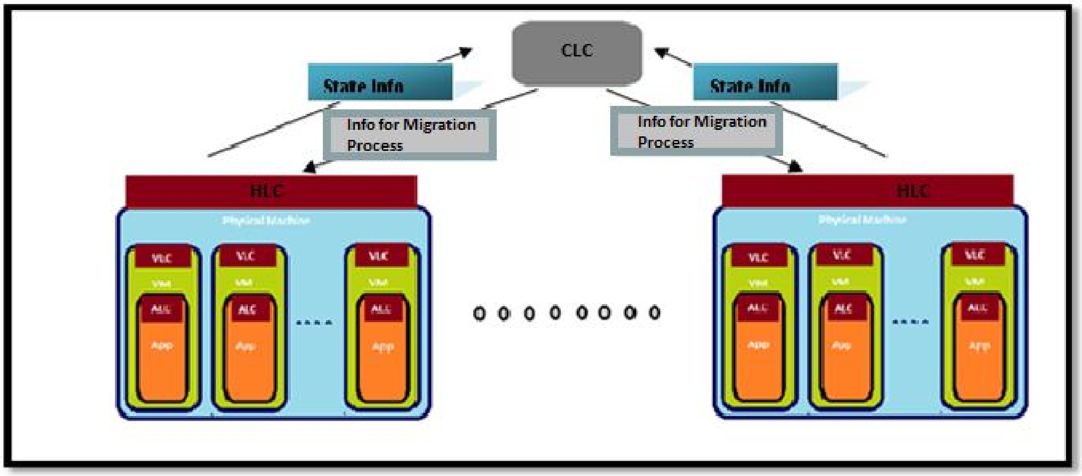

Figure 2. An architecture of four-level controllers

every specified period. In case there have messages informing the application is running on more than enough resources, VLC reallocates the resources for that application and takes back the extra resources. But, VLC can reallocate for the application by giving additional resources when VLC receives messages like they need more resources. However, this solution is available when VLC has idle resources. If it has not, it has to inform immediately to HLC.

4.1.3 Host Level Controller

When the VLC informs their status, HLC performs its operation. If HLC sees the acknowledgement, it will do nothing special for that VM as this VM is fit with its quota. If the message is about extra resources, HLC will try to reallocate the given resources. In this way, these extra resources can give to another virtual machine which requests for more resources. If the message is from the VM which needs more resources, HLC will give more by reallocating. But this condition is available when HLC has enough resources to give it. Unless the physical machine itself has enough resources, HLC knows exactly it is overloaded. To solve such this condition immediately, HLC has to send urgent message to CLC. CLC will give efficient migration decisions for a stable condition of the data center.

HLC sorts the loads of the virtual machines of the host which it is also existed on. Therefore, the ascending order of VMs can also be announced when HLC sends urgent message. In this way, CLC can know easily which VM is the most overloaded and which one is the under loaded machine of the host in the urgent message. The system parameters which are considered for the VMs are described in Table 2. The utilization of each VM can be computed according to the equations described below.

The vector R stands for the resources and it is a four dimensional vector.

$$R = \{CPU\%, RAM\%, NET\%, DISK\%\} \quad (1)$$

$$Weight = \{WR_1, WR_2, WR_3, WR_4\} \quad (2)$$

$$LVM_k = \sum_{i=1}^{4} Weight_i \times R_i \quad (3)$$

where i = 1,2,.,4 as the four main resources are considered and k = 1,2,….,n as each host can contain n virtual resources.

The overall load of each host can be obtained from the equation:

$$LH_j = \sum_{k=1}^{n} LVM_k \quad (4)$$

where j = 1, 2,…., m as the data center contains the maximum number of **‘m’** physical machines at a time.

Let matrix R(t) denote the residual capacity of the system

$$R(t) = [r_1(t), r_2(t), \ldots\ldots, r_m(t)] \quad (5)$$

Where R(t) is the vector of residual capacity of host at time t and $r_i(t)$ is the residual capacity of the i^{th} resource at time t.

$$RC = \sqrt{(r_1(t)^2 + r_2(t)^2 + \cdots + r_m(t)^2)} \quad (6)$$

4.2 Fuzzy Logic for Resource Management

This section describes the fuzzy membership functions and fuzzy rules for the four-level controllers.

4.2.1 Fuzzy Logic for ALC, VLC and HLC

If idle resources are possible, both of the ALC and HLC uses the fuzzy logic controller to decide how much resource is needed to add. The logic controller has two input variables N and U and one output variable ΔV. N is the number of requests in the observation window and U is the current resource usage. ΔV is the change of resource capacity. Whenever each window expires, the fuzzy controller is invoked to generate a suitable change of the resource capacity according to the current values of N and U. Both N and U are a series of fuzzy sets as Low , Medium and High. As shown in Figure 3, their membership functions $f(N)$ and $f(U)$ are respectively chosen as trapezoidal and triangular ones where N_L, N_M, N_H are critical points of $f(N)$.

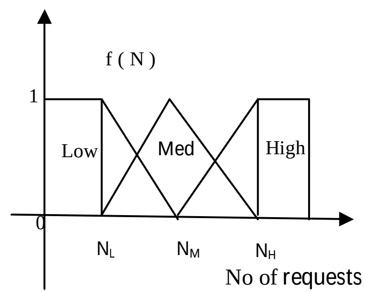

Figure 3. Membership functions of number of requests f(N)

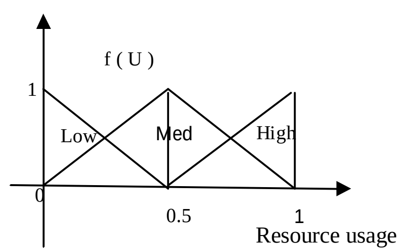

Figure 4. Membership functions of resource usage f(U)

The fuzzy sets of the output variable ΔV are defined as shown in Table 1. Their membership functions are selected as triangular ones. It should be stressed that this type of membership functions is easy for computation and is commonly used in practical applications of fuzzy control. Intuitively, the utility capacity should be reduced when the usage ratio U goes low, and vice versa. The fuzzy rules included in Table 3 reflect this intuitive description of the control policy.

To derive the activation levels of the output fuzzy sets. The fuzzy inference engine adopts the Max-Min fuzzy inference method. All these weights are used to derive ΔV during defuzzification.

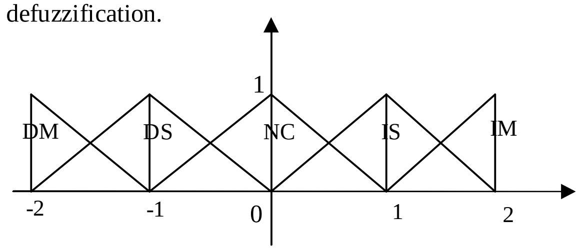

Figure 5. Membership functions of change of resource ΔV

Table 1. Fuzzy Rules for ALC, VLC and HLC

	IF		THEN
No	N	U	ΔV
1	Low	Low	DM
2	Low	Med	DS
3	Low	High	IS
4	Med	Low	DS
5	Med	Med	DS
6	Med	High	IS
7	High	Low	DS
8	High	Med	NC
9	High	High	IM

NC = No change, DS = Decrease Slightly
DM = Decrease Moderately
IS = Increase Slightly
IM = Increase Moderately

4.2.2 Fuzzy Logic for Central Level Controller

As for CLC, the first step it is needed to operate is to find which host can cause overloading. The main task of CLC is to decide which one is the sender physical machine, which one is the receiver machine and which one is the intended guest virtual machine to migrate. In some cases, migrating overloaded VM is the best choice for the host machine. However, sometimes removing the VMs who have less loaded to other host machine and leaving the overloaded one with enough resources is a better decision. The most important thing is to get the transparency in the sight of VM users. For these reasons, CLC scores each VM on the overloaded host based on the cost function. As a first step, hosts are grouped into sender, receiver and no change group according to their loads and residual capacity. This grouping problem is solved by fuzzy logic controller again. After examining the groups, the hosts are ranked according to their loads in the sender groups. The hosts in the receiver groups are ranked with their associated residual capacity. When the most overloaded machine is chosen in the sender group, the virtual machine with the least cost is decided to migrate to the host with most residual capacity in the receiver group. This process is looped again all the hosts of sender group are migrated. In that way, Libra could manage a load balanced data center. Actually, all of the HLC send state data at predefined time. But when one of the HLC sends its state data as an urgent message, other hosts' state data. The fuzzy logic controller of the control unit has two fuzzy sets as input domains, namely, f (LH_j) and f (RC). The output fuzzy sets are shown in figure 6. The VM which uses the least memory pages is regarded as the VM of least cost machine.

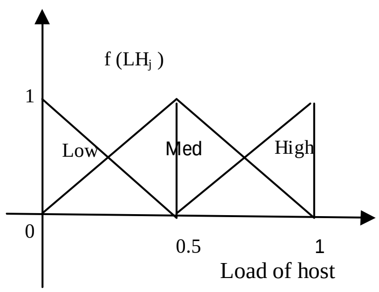

Figure 6. Membership functions of load of host f(LHj)

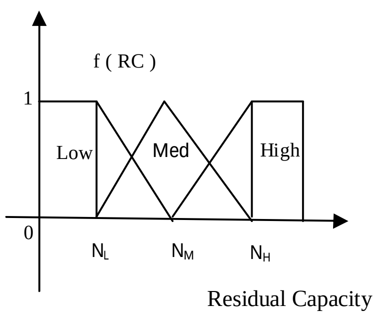

Figure 7. Membership functions of residual capacity f(RC)

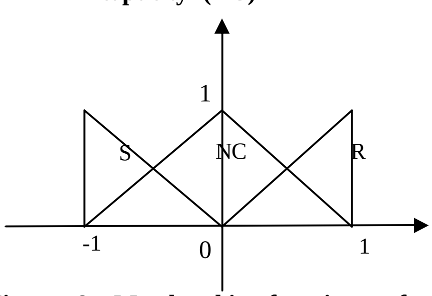

Figure 8. Membership functions of output fuzzy sets

Table 2. Fuzzy Rules for CLC

	IF		THEN
No	LH_j	RC	G
1	Low	Low	NC
2	Low	Med	NC
3	Low	High	R
4	Med	Low	NC
5	Med	Med	NC
6	Med	High	R
7	High	Low	S
8	High	Med	S
9	High	High	NC

NC = no change, S = sender, R = receiver

5 Conclusion and Future Work

In this paper, we have discussed the usage of fuzzy logic control system in the environment of our test bed datacenter. As an ongoing research, we have just build the fuzzy inference engine with efficient rules and currently testing with the sample data sets. In this paper, we could not present our experimental results as we are now building infrastructure and testing. Steps of detailed implementation and rich experimental results will be our future work. According to our manual testing with some sample data, we are sure that fuzzy logic decision making has the capability of controlling the nonlinear relations, uncertainty and dynamic nature. It is also able to perform well in situations with unpredictable and abrupt peak loads. In the future, we would like to extend our algorithm for the historical trends of the service types in the data center. In this way, Libra can limit the unnecessary migrations of virtual machines.

References

[1] B.Kosko. Neural Networks and Fuzzy Systems. First Edition, Prentice Hall of India, 1997.

[2] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach,I. Pratt, and A. Warfield. Live migration of virtual machines. In Proc . NSDI '05, May 2005.

[3] D.Gmach, J.Rolia, L.Cherkasova,A. Kemper.Resource pool management : Reactive versus proactive or let's be friends. In Journal of Elsevier , 2009

[4] E.Cox. Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration. First Edition, Elsevier,2000.

[5] I.Mevag.Towards Automatic Management and Live Migration of Virtual Machines. Thesis of Oslo University, May 2007.

[6] J.Xu, M.Zhao, J.Fortes, R.Carpenter and M.Yousif. Autonomic Resource Management in virtualized data centers using fuzzy logic-based approaches. In Proc. IEEE, July 2008.

[7] K. Appleby, S. Fakhouri, L. Fong, M. Goldszmidt, S. Krishnaku-mar, D. Pazel, J. Pershing, and B.Rochwerger. Oceano sla ,oc. IFIP/IEEE Symposium on Integrated Management, May 2001.

[8] K.Begnum and M.Disney. Decision Support for Virtual Machine Re-Provisioning in Production Environments. 21st Large Installation System Administration Conference, LISA' 07.

[9] M. Nelson, B. Lim, and G. Hutchins. Fast Transparent Migration for Virtual Machines. In Proc. USENIX 2005.

[10] M.Tarighi, S.A.Motamedi, S.Sharifian. A New Model for Virtual Machine Migration in Virtualized Cluster Server Based on Fuzzy Decision Making. In Journal of Telecommunications, Volume 1, Issue 1, February 2010.

[11] P.Padala, X.Zhu, M.Uysal and Z. Wang. Automated Control of Multiple Virtualized Resources. In Proc. EuroSys'09 ACM, April 2009.

[12] T.Wood, P.Shenoy,A.Venkataramai, and M.Yousif.Black Box and Gray Box Strategies for Virtual Machine Migration . Computer Networks Journal Special Issue on Virtualized Data Center 2009.

[13] VMWare,Inc.DRS Performance and Best Practices, 2008.

[14] VMWare Web Site, Server Consolidation and Containment Solutions Brief, http : // www. vmware. com/ pdf / server consolidation.pdf

[15] Z.Chen, Y.Zhu and M.Yuan. Towards Self-Optimization in Utility Computing using Fuzzy Logic Controller. In Proc.IEEE, May 2005.