

INTERNATIONAL JOURNAL OF
RESEARCH IN COMPUTER
APPLICATIONS AND ROBOTICS

ISSN 2320-7345

A NOVEL METHOD TO IMPROVE THE PERFORMANCE OF DYNAMIC DISTRIBUTED NETWORKS

Shahram Jamali¹, Khadije Hourali²

¹Department of Computer Engineering
University of Mohaghegh Ardabili, Ardabil, Iran
Jamali@iust.ac.ir

²Department of Computer Engineering
University of Mohaghegh Ardabili, Ardabil, Iran
hourali68@gmail.com

Abstract

Virtual machine placement is the process of mapping virtual machines to physical machines. In other words, virtual machine placement is the process of selecting the most suitable host for the virtual machine. The process involves categorising the virtual machines hardware and resources requirements and the anticipated usage of resources and the placement goal. The placement goal can either be maximizing the usage of available resources or it can be saving of power by being able to shut down some servers. The autonomic virtual machine placement algorithms are designed keeping in mind the above goals. In this paper, we employ ORESTE method to design an integrated virtual machine placement algorithm, called ORESTE VM Placement (OVMP) which can reduce the number of running physical machines and lower the energy consumption. Simulation results in CloudSim environment show that the proposed algorithm outperforms existing algorithms in terms of traffic cost, migration, SLA and energy.

Keywords: ORESTE, Energy Consumption, Migration; PM.

1. Introduction

Cloud computing is a new technology in academic world. In a cloud platform, resources are provided as service under a predefined Service Level Agreement (SLA). But, since the resources are shared, subscribers' requirements have big dynamic heterogeneity and sometimes platform is irrelevant, the resource may be wasted if they cannot be assigned properly (Beloglazov & Buyya, 2012). On the other hand, dynamically balancing the load among the servers improves resource utility and the overall cloud performance. Therefore, an important problem to be solved is how to dynamically and efficiently manage resources to meet the subscribers' requirements and to maximize the overall performance. The customer is interested in reducing the overall execution time of tasks on the machines. The processing units in cloud environments are called as virtual machines (VMs).

VM placement is the process of mapping VMs to PMs. In other words, virtual machine placement is the process of selecting the most suitable host for the virtual machine. The process involves categorizing the virtual machines hardware and resources requirements and the anticipated usage of resources and the placement goal. The placement goal can either be maximizing the usage of available resources or it can be saving of power by being able to shut down some servers. The autonomic virtual machine placement algorithms are designed keeping in mind the above goals.

Cloud computing utilizes virtualization to enable cloud users to hire computing resources from cloud data center as a service instead of owning it. Applications are executed in isolated virtual machines (VMs) which are running on a shared physical infrastructure. Cloud users create an image for an application and initialize a number of VMs on demand. The number of VMs is adjustable to deal with workload spike. These workloads can be scientific or commercial applications with different resource requirement. They can be CPU intensive, CPU network balance, or network intensive applications. Because of the differences in VM resource requirement, consolidating them to maximize utilization of overall system is a complicated work. In order to host a VM, a PM must provide all resources the VM requires, including CPU utilization, memory, storage and bandwidth. Among those resources, CPU utilization is the only one provided dynamically according to performance requirement while other resources are provided with fixed size. Due to that reason, most of current researches migrate VMs based on CPU utilization (Meng et. al, 2010; Soule et.al, 2004). However, for many applications, the performance is not only relied on CPU utilization. For ones that require communication among services, the communication cost can also influence the overall performance. For example, for a 3 tier web application, migrating an application server to a section far from the front end web server and the database server will increase the communication latency, thus reduce the overall throughput. Another example is non-overlap MPI applications which wait for messages before continuing. Research (Wood et. al, 2007) shows that service fragment can affect the data center network performance. In paper (Nucci et. al, 2005), the placement of virtual machines that execute the reduce phase of a MapReduce application can reduce total job runtime by 4 times. As the demand for resource provided by cloud computing increase, the energy consumption of data centers becomes a pressing issue. According to (Gul & Hussain, 2012), between 2000 and 2006 the amount of energy consumed by data centers around the world has doubled and today datacenter electricity consumption is almost 2% of world production. The energy consumed by cloud data centers not only influent provider electricity bill, but also CO2 emission and global warming (Bianchini & Rajamony, 2004). Due to the energy consumption of components such as hard disk, memory, main board, a server at idle state still consumes about 70% of the energy it consumes at full CPU speed (Nucci et. al, 2005). In order to save energy, VMs are consolidated to reduce the number of physical machines (PMs). Unused PMs are turned on and will be turned on using techniques such as Wake on LAN when the demand for resource increases. In this paper, we propose a virtual machine placement mechanism that considers traffic as well as power among VMs within a cloud data center. The goal of this paper is to minimize the communication cost and also save energy.

The rest of this paper is organized as follow. Firstly, we introduce some related works in Section II. In Section III, we explain the proposed algorithm in detail and verify that effectiveness of our algorithm through evaluations which are simulation based experiments using realistic workloads in section IV. The last Section Vends with conclusion and future works.

2. Related Works

Much recent research has been devoted to investigating algorithms for allocating virtual machines (VMs) to physical machines (PMs) in infrastructure clouds. Many such algorithms address distinct problems, such as initial placement, consolidation, or tradeoffs between honouring service-level agreements and constraining provider operating costs. Even where similar problems are addressed, each individual research team evaluates proposed algorithms under distinct conditions, using various techniques, often targeted to a small collection of VMs and PMs. Some of the approaches for virtual machine placement are explained in the subsequent paragraphs. The placement problem is a non deterministic problem. Following are some of the algorithms that have been used to solve the virtual machine placement problem. The literature identifies that VM-placement

decisions can be made under any of at least three different regimes (Shang et. al, 2011): (1) reservations (Fujiwara et. al, 2010) (2) on-demand access (Amazon EC2, 2010) and (3) spot markets (Fujiwara et. al, 2010; Andrzejak et. al, 2010). In one reservation regime (Amazon EC2, 2010), a user pays a fee per instance per VM type for a period (e.g., one year) during which the specified VMs may be acquired at a discount from published usage charges. In on-demand access regimes, a user simply requests a specified number of one or more VM types needed immediately, and pays for VM usage according to a fixed schedule of fees. In spot markets, a provider's prices fluctuate over time and a user specifies the usage rates they are willing to pay for requested VMs. When the provider price falls to or below the user's willingness to pay, then the user's requested VMs are launched. Should the provider price subsequently rise above the user's willingness to pay, then the user's VMs are terminated, and can only be restarted when the price falls to the level the user is willing to pay. In the grand scheme of resource-allocation decision making, one can envision PMs migrating back and forth among three pools, each assigned to one of the three regimes, as demand for VMs varies. Consideration of how best to allocate PMs to each pool would seem a ripe area for research (Shang et. al, 2011). We restrict our study to consider only on-demand access. In on-demand clouds, there are potentially two types of VM placement decisions to be made: (1) initial placement (Cardosa et. al, 2010) and (2) migration (and/or resizing) of VMs over time (Bobroff et. al, 2007; Malet & Pietzuch, 2010), as PM availability changes, as consolidation is needed to conserve power and in response to the degree to which service-level agreements (SLAs) are being achieved.

Most previous research on initial VM placement considered only PMs within a single cloud, but in one case (Mark et. al, 2011) placement decisions considered which of several clouds to choose. In the existing literature, initial placement and VM migration are usually considered as separate topics, though in some cases similar algorithms may be adopted. Future research might consider interaction between initial placement and migration decisions, especially under situations where tradeoffs are needed among power conservation, SLAs, revenue maximization and reliability. We restrict our study to consider only initial VM placement.

One could consider initial VM placement in on-demand clouds at two levels: (1) cluster and (2) node (i.e., PM). When VMs communicate, placing them on the same cluster makes good sense because communication among the VMs will be local to a cluster switch. Most existing research (Meng et. al, 2010; Cardosa et. al, 2010; Van & Tran, 2010; Machida et. al, 2010) considers PMs as an unstructured pool, where restricting VMs to a shared cluster would be accomplished by designating a Boolean attribute, one of potentially many attributes over which some optimization algorithm or bin packing heuristic would be executed. In our study, guided by the open-source code in Eucalyptus (v1.6) (Nurmi, et al, 2009), we adopt explicit use of two distinct decisions levels: (1) choosing a cluster for all VMs in a given request and then (2) choosing specific PMs within the selected cluster. Taking this course is the same as assuming that all VMs within a single request will communicate. VMs that need not communicate would then be included in separate requests. In most VM placement algorithms, PMs are partitioned into two sets: (1) those that meet some criteria and (2) those that do not. Subsequently, the set of PMs that meet the criteria are ordered, and VM placement attempts are made starting with the first PM on the list, and continuing until all

VMs have been placed or until the set of qualified PMs is exhausted. Various criteria have been used to order qualified PMs. For example, many researchers (Cardosa et al, 2010; Xu & Fortes, 2010; Bellur et al, 2010; Machida et al, 2010) adopt ordering heuristics based on the literature associated with online bin packing (Meng et al, 2010). Other schemes extend those heuristics by adding specific attributes (e.g., CPU usage, network and disk controller usage, and memory usage), summarized into a weighted value used to order PMs or to assign categories that can be used to order PMs. In some schemes, attributes used to order PMs are determined by individual VM users (Machida et al, 2010; Malet & Pietzuch, 2010), while in other schemes attributes are determined by the provider (Cardosa et al, 2010; Meng et al, 2010; Fontan, 2010), or user and provider attributes are combined (Van & Tran, 2010; Mark et al, 2011; Machida et al, 2010, Das et al, 2010). To limit our study, we elected to use heuristics based on those found in online bin-packing literature. The method we use to compare placement heuristics should be applicable to any specific set of VM-placement algorithms that one wishes to compare

3. ORESTE VM Placement algorithm (OVMP):

3.1 ORESTE Method

If we consider A as a limited set, these alternative shall be analyzed by the set including k. in this method, the relative importance of each index is not specified by their weight, but it is stated by a superiority structure on the index, which is described under a weak level. The so called weak level is stated in a full and transition Equation of S, which is consisted of P and I Equations. P or superiority show discrepancy and I shows incuriosity, which the representative of superiority coordination among the criteria. Also for each of the criteria of $j=1 \dots k$, a superiority structure in the set A is described, which is similar to C criteria of the superiority structure is transitional and consisting of a set of P and I relationships (Jafari et al., 2013). Thus, the 1st superiority structure is established based on criteria' relative importance to each other and the 2nd superiority structure also created on the optional set and according to each one of them individually. After formation of the abovementioned 2 superiority structures, we should pay attention to the preliminary ranking of these structures. To do so, we may use Besson average ranking method. In such a way to refer to the superiority structure 1st and according to its rank in comparison to all other criteria, dedicate numbers 1-K (k index) and for all alternative numbers 1-m (m criteria). Then we obtain the mean from the maximum or the minimum dedicated number which is constructed based on the superiority structure enjoys similar superiority or I (Equation 1). In other words, instead of dedicating grades 1 and 2 to the so called two criteria (alternative), we shall grant it to both ranks (1/5); therefore, with Besson average ranking, the priorities shall turn to ranks. The obtained rank for criteria shall be represented by r_k and the gained rank for each option in each index shall be represented by $r_k(m)$ (Brans et al., 1986).

$$\frac{X_1 + X_2}{2} = \bar{x} \quad (1)$$

X_1 is the maximum amount while X_2 is the minimum amount and is regarded the average distance. ORESTE Method to perform the ranking process has 3 phases as the following. Projection of alternative intervals $d(o, m_k)$: Estimating in ORESTE method is based on using the hypothetical matrix called position- matrix that in all its columns the decision alternative are organized from the best to the worst and accordingly the columns are arranged based on the criteria ranks. By scanning matrix's members eventuating from the main diameter, the best situation are listed on the left side of the diameter and the worst are at the right side. Then a zero offset is located at the very end of the left side of the diameter and all the formed pictures are considered and their intervals are determined from the zero offset which is shown by $d(o, m_k)$ as it is shown below (Brans and Mareschal, 2005).

The interval estimation $d(o, m_k)$, which was explained above is executed for different modes including:

Direct linear estimation: In this mode to perform the interval estimation $d(o, m_k)$ from r_k and for option m in k index we shall comply to Equation (2).

$$d(o, m_k) = \frac{1}{2} [r_k + r_{k(m)}] \quad (2)$$

Indirect linear estimation: In this mode, pictures' intervals from the offset point are computed as the following using Equation (3):

$$d'(d, a_k) = ar_k + (1-a)r_{k(m)} \quad (3)$$

Non-linear estimation: In non- linear scanning mode to determine the pictures distances from the desired origin we use Equation (4):

$$d''(o, m_k) = \sqrt{r_k + r_{k(m)}^2} \quad (4)$$

To achieve more general conditions, Equation (5) shall change as follows.

$$d'''(o, m_k) = \sqrt[r_k + r_{k(m)}^R]{} \quad (5)$$

And finally if we add the normal weights of, Equation (6) shall be gained.

$$d''(o, m_k) = \sqrt[R]{(a, r_k^R + (1-a)r_k(m)^R)} \quad (6)$$

In this regard, with respect to some amounts, the R distance of d shall be illustrated.

Mean of balanced arithmetic.

Mean of balanced arithmetic	$R = 1 \rightarrow d''$	$R = -1 \rightarrow d''$	Geometry mean
Mean of squares	$R = 2 \rightarrow d''$	$R = -\infty \rightarrow d''$	$\min(r_k, r_{k(m)})$
		$R = +\infty \rightarrow d''$	$\max(r_k, r_{k(m)})$

Global ranking of the alternative interval $R(m_k)$: By determining the interval of the scans pertaining to matrixes' members, the sources' position and the global ranking shall be implemented by one of the abovementioned styles. Generally speaking, selecting every mode or different R amounts for scanning and determining intervals $d(o, m_k)$ with the solemn intention of creating an impact on their position in comparison to each other which in progress, the intervals with the assistance of Besson average ranking method and consequently the issue shall revert to its original sequential essence. The result of this ranking equals to the obtained rank by Besson method and the intervals of is in a way that we shall have the following e.g (Goumas and Lygerou, 2000)

$$R(a_1) \leq R(a_2) \quad \text{if} \quad d(o, a_1) \leq d \quad (7)$$

The obtained ranks are called the total ranks and all exist in the following scope:

$$1 \leq R(m_k) \leq m, k$$

Thus an incremental sequential structure is modified based on and with regard to the following Equations:

$$\text{if } R(a) < R(b) \quad \text{then} \quad aPb$$

$$\text{if } R(a) = R(b) \quad \text{then} \quad aqb \quad (8)$$

An option that the relative is smaller is more appropriate and a better rank shall be awarded to it; in other words, it is the top option in which the total sum of all its criteria is less than the others.

3.2 OVMP Algorithm

The defined space of Cloud computing consists of K clusters for processing (service) .m is the number of PMs that is such as $P = \{P_1, P_2, \dots, P_m\}$. n is the number of VM per physical server that is shown as $VM(R_f) = \{VM_1, VM_2, \dots, VM_n\}, f \in [1, m]$. Each PM in the cloud computing environment has four characteristics that include (C_i, M_i, I_i, B_i) , where C_i which is based on MI Represents the CPU capacity of PM_i , Mem_i which is based on MB Represents the memory capacity of PM_i , Other specifications include IO allocated to the considered VM which is based on MB/Sec, and bandwidth of PM_i which is based on (MB/Sec).

For each VM four characteristics are defined as $VM_j = (c_j, m_j, i_j, b_j)$. c_j is processing power of each VM that is the number of instructions executed by each processing elements of source in terms of million per second (MIPS). m_j and i_j respectively represents the rate of utilization of memory and input/output, which is

calculated based megabyte per second (MB /S), b_j represent amount of bandwidth requirement for VM_j . Weights of VM's characteristics are calculated as bellow by eq.9, sum of this weight must be equal to 1:

In this paper, the weight vector is calculated by the following method:

$$if \sqrt{c^2 + m^2 + I^2} = x \tag{9}$$

$$weight \ C = \frac{\frac{c}{x}}{\frac{c}{x} + \frac{m}{x} + \frac{I}{x} + 2k}$$

The parameter 'k' represents the weight of bandwidth. Weight of memory, I/O and maturity (or cost) respectively with replacement values of M, I and K instead of $\frac{c}{x}$ in the numerator are obtained. Weights obtained are in the range of [0 -1] and will be their sum is equal to one.

According to the following table, Three Scenarios are considered for setting up cloud environment, that results in the below charts

Scenarios for Set cloud environment

The amount of elapsed time for each VM on each iteration (t)	0.0004					0.0006					0.0008				
K' value	0.001	0.2	1	2	4	0.001	0.2	1	2	4	0.001	0.2	1	2	4

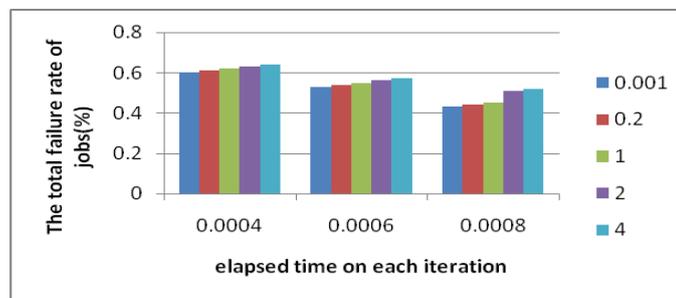


Figure1. The total failure rate of jobs in various services

As shown in Fig.1 with Increasing the repetition time, VMs workload ends faster, this means the tasks failure rate is decreased and the amount of completed tasks is increased. So in less replication time, more jobs are lost, because the VM workload is high. Given the low value of K also means less attention to two issues of bandwidth. In other words, with low K-value weighted amount of CPU, memory, and input/output is increased, that means better allocation of tasks to the processing resources reduces the amount of failed tasks. to show the quality of service the following equation was used.

$$Qos = \left(\frac{rc}{b}\right) + \left(\frac{m}{d}\right) \tag{10}$$

b and d respectively, are the maturity and budgeted amount for each task and rc and rn respectively represent the bandwidth of processing nodes that the task assigned to it. And the maximum amount of QOS will be equal which is most desirable. Increasing the value of QOS is meant to increase the quality of service.

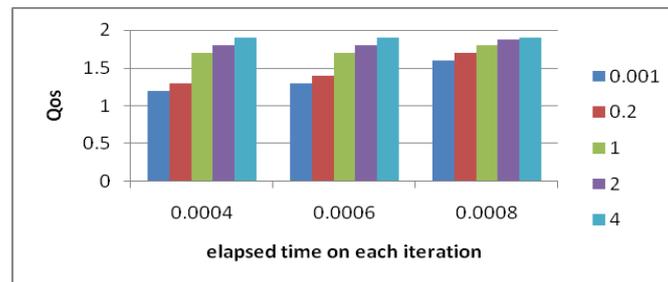


Figure 2. Quality of service in various services.

In the figure above, with increasing k, due to increasing weight of bandwidth, quality of service is increased and with reducing K value, Suitability of task with processing resources of nodes is more considered, so the quality of service is reduced.

According to Fig.2 increasing the K value increase it is desirable because it increases the quality of service. And according to Fig.1 reducing the k value is desirable because it leads to Reduce unsuccessful tasks.

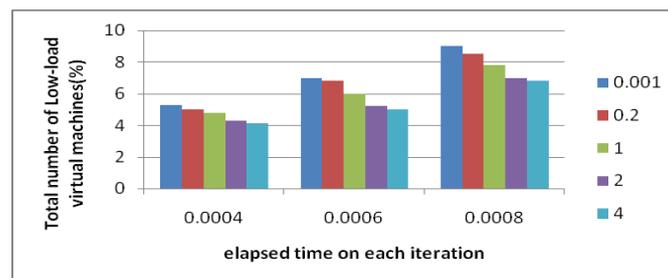


Figure 3. Total number of Low-load virtual machines in various services

According to Fig.3 the average of each of the three times in Figure is 4.76, 6.04, 7.88.

When k value is small, weights of resources are increased so tasks are assigned to nodes that are more suitable. As a result, virtual machines quickly unfilled and remains Low-loaded.

However, with increasing the K value, the importance of budget and maturity is increased. And tasks may be sent to the VM that not have a large proportion with the task. This has led to faster full VMs capacity and therefore low-load VMs are decrease.

According to Fig 1,2 and 3 reduction of K value would be desirable and according to Figure 4.2 The ideal case is increasing the k value, so should be consider a moderate state. here the replication time is assumed equal to 0.0005 and K is assumed equal to 1, According to this state, the best mode is selected. The proposed algorithm is as follows:

First the overloaded servers are removed from the table, then by considering the characteristics of each PM (figure 3), weights of VM's requirements, using ORESTE method and taking server table, the best server for placement target VM is determined. Server information is constantly updated. The advantage of this approach is that the VM allocation process is dynamically done based on current condition of environment. By using this algorithm the VM migration is minimized.

4. Experimental Results

We evaluate the efficacy of the traffic and power aware virtual machine placement in a simulation environment using Cloudsim toolkit (Rodrigo et. al, 2011). The simulated datacenter has 1 core switch which connected to 3 aggregation switches. Each aggregation switch in turn is connected to 5 edge switches. Finally each edge switch is connected to ten PMs to form a partition. Totally the data center contains 150 PMs. It is a worth notice that since our algorithm is based on the concept of distance and cost matrix, it can be applied for any topology. The running period is 24 hours to simulate the diurnal pattern of a communication network (Singh et. al, 2008). The VM and PM configurations are as same as (Beloglazov & Buyya, 2012), plus that all PMs in a partition have the same configuration.

We use FNSS (Rehman & Hussain, 2011) to generate a cyclo-stationary traffic map which updated every hour. First, the static mean traffic volumes is generated follow a lognormal distribution with standard deviation (σ) equals to 3 to form an environment where some VMs are linked with high traffic (Dias & Costa, 2012). These static volumes then added a zero-mean normal fluctuation value. According to (Moreno & Xu, 2011), the relation between the standard deviation of this fluctuation (σ') and the mean traffic volumes is

$$\overline{x_{ij}(t)} = \psi \sigma_{ij}^{\gamma} \quad (11)$$

We chose $\psi = 0.6$ and $\log \psi = -0.21$ as same as Sprint Europe network. Finally, traffic volumes are multiplied by a sin function with unitary mean to model the daily fluctuation. Based on the mean traffic volumes, we classify VMs into three categories: network- intensive, CPU-network balance and CPU-intensive servers. The CPU utilization of each VM is then generated correspond to which category it belongs to. In (Beloglazov & Buyya, 2012), the author presents some statistical policy to determine whether a PM is over- utilized and which VM should be migrated.

Among the dedicated heuristics, the combination of Local regression and Minimum migration time (LrMmt) with safety parameter 1.2 produce the best energy consumption with acceptable SLA. We define this algorithm as Knapsack. In the simulation, the experiment results when OVMP, Greedy Worst-Fit and Knapsack algorithms are applied are compared (figures 4, 5, 6 and 7). Originally all vertices in the traffic graph are connected to the others. Every round, a portion of edges is removed from each subgraph until no edge remains and all the vertices are absolutely isolated.

According to figure 4, OVMP saves about 13% of traffic cost compared to Knapsack algorithm. Moreover, OVMP also save about 31% SLA violation when the number of VM is not so high. When the number of VM reach high value, there are not many available position for VM migration, thus cause high SLA violation and energy consumption, but reduce the number of migrations for all algorithms. The number of migrations and energy consumption cause by OVMP and Knapsack are nearly the same. The Greedy Worst-Fit algorithm saves about 35% of traffic cost, but double the value of SLA violation, 22% number of migrations and 13% energy consumption (Fig.5, 6 and 7).

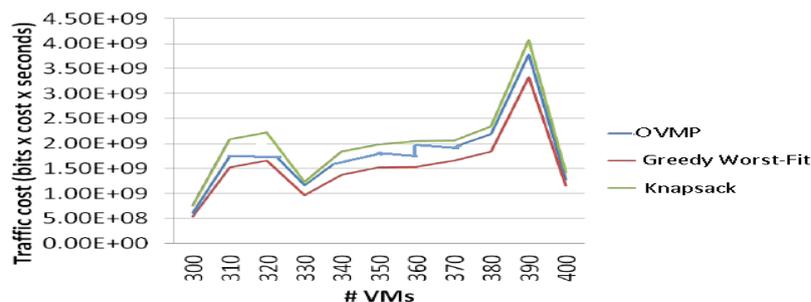


Figure 4: Traffic cost

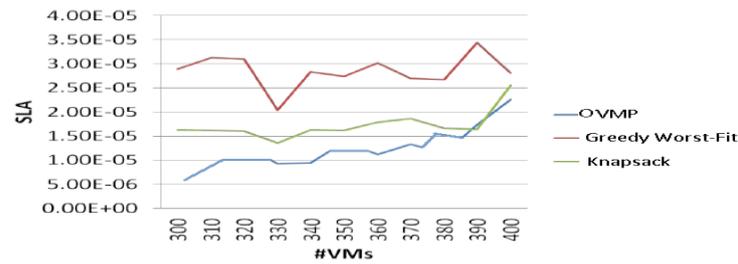


Figure 5: SLA

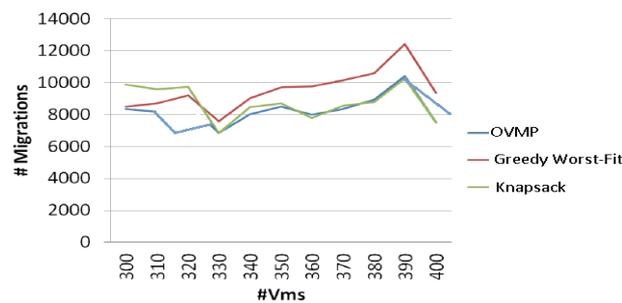


Figure 6: Migrations

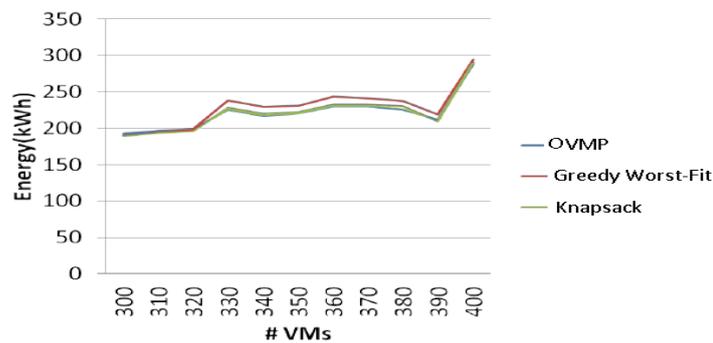


Figure 7: Energy

4. Conclusion

This paper presents a virtual machine placement algorithm with using ORESTE method in cloud data centers that minimize network congestion while energy consumption is unchanged. This allocation is a choice between existing physical machines for considered virtual machine, based on the weights of virtual machine's criteria and all physical machines' characteristics. At the end of the simulation, virtual machines are consolidated on physical machines with high CPU usage per energy consumption and heavy communicated virtual machines are hosted by physical machines that located close together. The result shows that our algorithm produce better balance result considering virtual machine communication cost, SLA violation and energy consumption.

REFERENCES

- [1] Anton Beloglazov and Rajkumar Buyya, 2012, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, *Concurr. Comput: Pract. Exper.*, 24(13), pp.1397-1420.
- [2] R. Bianchini and R. Rajamony, 2004, "Power and energy management for server systems", *Computer*, 37(11), pp.68-76.
- [3] Irfan Gul and M. Hussain, 2012, Distributed cloud intrusion detection model, *International Journal of Advanced Science and Technology*, 34, pp.71-82.
- [4] Xiaoqiao Meng, Vasileios Pappas, and Li Zhang, 2010, Improving the scalability of data center networks with traffic-aware virtual machine placement, In *INFOCOM, 2010 Proceedings IEEE*, 1-9.
- [5] Antonio Nucci, Ashwin Sridharan, and Nina Taft, 2005, The problem of synthetically generating ip tra_matrices: initial recommendations, *SIGCOMM Comput. Commun. Rev.*, 35(3), pp.19-32.
- [6] Augustin Soule, Antonio Nucci, Rene Cruz, Emilio Leonardi, and Nina Taft, 2004, How to identify and estimate the largest traffic matrix elements in a dynamic environment. In *Proceedings of the joint international conference on Measurement and modeling of computer systems, SIGMETRICS '04/Performance '04*, New York, NY, USA., ACM, 73-84.
- [7] Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif, 2007, Black-box and gray-box strategies for virtual machine migration, In *Proceedings of the 4th USENIX conference on Networked systems design & implementation, NSDI'07*, Berkeley, CA, USA USENIX Association, 17-17.
- [8] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajku- mar Buyya, 2011, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Softw. Pract. Exper.*, 41(1), pp.23-50.
- [9] D.S. Dias and L.H.M.K. Costa, 2012 , Online traffic-aware virtual machine placement in data center networks, In *Global Information Infrastructure and Networking Symposium (GIIS)*, pp.1-8.
- [10] Ismael Solis Moreno and Jie Xu, 2011, Energy-efficiency in cloud computing environments: Towards energy savings without performance degradation, *IJCAC*, 1(1), pp.17-33.
- [11] Atiq Rehman and M. Hussain, 2011, Efficient cloud data confidentiality for daas, *International Journal of Advanced Science and Technology*, 35, pp.1-10.
- [12] Singh, M. Korupolu, and D. Mohapatra, 2008, Server-storage virtualization: Integration and load balancing in data centers, In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008*, International Conference, pp.1-12.
- [13] S. S. Seiden, R. V. Stee, and L. Epstein, 2003, New Bounds for Variable-Sized Online Bin Packing, *SIAM Journal on Computing*, Vol. 32, No. 2, pp.455-469.
- [14] E. Coffman, M. Garey, and D. Johnson, 1997, *Approximation Algorithms for Bin Packing: A Survey*, PWS Publishers, Boston.
- [15] S. Shang, Y. Wu, J. Jiang and W. Zheng, 2011, An Intelligent Capacity Planning Model for Cloud Market, *Journal of Internet Services and Information Security*, 1:1, pp. 37-45.
- [16] Fujiwara, K. Aida, and I. Ono, 2010, Applying Double-sided Combinational Auctions to Resource Allocation in Cloud Computing, *Proceedings of the 10th Annual International Symposium on Applications and the Internet*, IEEE, July 19- 23, pp.7-14.
- [17] Amazon Elastic Compute Cloud (Amazon EC2) <http://aws.amazon.com/ec2/>, 2010.
- [18] Andrzejak, D. Kondo, and S. Yi, 2010, Decision Model for Cloud Computing under SLA Constraints, INRIA Technical Report-004/4849, Version 1.
- [19] M. Cardosa, A. Singh, H. Pucha and A. Chandra, 2010, Exploiting Spatio-Temporal Tradeoffs for Energy Efficient MapReduce in the Cloud, Dept. of Comp. Sci. and Eng., University of Minnesota, TR-10-008.
- [20] X. Meng, V. Pappas and L. Zhang, 2010, Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement, *Proceedings of IEEE 2010 INFOCOM*.
- [21] J. Xu and J. Fortes, 2010, Multi-objective Virtual Machine Placement in Virtualized Data Center Environments *Proceedings of the 2010 IEEE/ACM Conference on Green Computing and Communications*, pp.179-188.
- [22] U. Bellur, C. Rao and M. Kumar, 2010, Optimal Placement Algorithms for Virtual Machines, *Proceedings of CoRR*.
- [23] J. Fontan, 2010, Session 6: Advance Usage of OpenNebula, *OpenNebula Technology Days*.
- [24] H. Van and F. Tran, 2010, Autonomic resource management for service host platforms, *Proceedings of Workshop on Software Engineering Challenges in Cloud Computing*, Vancouver, Canada.

- [25] C. Mark, D. Niyato and T. Chen-Khong, 2011, Evolutionary Optimal Virtual Machine Placement and Demand Forecaster for Cloud Computing, Proceedings of the 2011 IEE International Conference on Advanced Information Networking and Applications, pp.348-355.
- [26] F. Machida, M. Kawato and Y. Maeno, 2010, Redundant Virtual Machine Placement for Fault-tolerant Consolidated Server Clusters, Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium, Osaka, Japan, 32-39.
- [27] N. Bobroff, A. Kochut and K. Beaty, 2007, Dynamic Placement of Virtual Machines for Managing SLA Violations, Proceedings of the 10th IFIP/IEEE Symposium on Integrated Network Management, pp.119-128.
- [28] S. Das, M. Kagan and D. Crupnicoff, 2010, Faster and Efficient VM Migrations for Improving SLA and ROI in Cloud Infrastructures, DC CAVES 2010.
- [29] B. Malet and P. Pietzuch, 2010, Resource Allocation across Multiple Cloud Data Centres, Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, ACM.
- [30] C. Hyser, B. McKee, R. Gardner and B. Watson, 2007, Autonomic Virtual Machine Placement in the Data Center, HP Laboratories, HPL-2007-189.
- [31] D. Nurmi, et al., 2009, The Eucalyptus Open-Source Cloud- Computing System, Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, May 18-21, pp.124-131.
- [32] Yu, P.L., 1973, A Class of Solutions for Group Decision Problems, Management Science 19, pp. 936-946.
- [33] Jafari, H. and dadkhah, A. 2013. Risks Analysis of Container Handling Operation Using ELECTRE and Shannon's Entropy Methods. International Journal of Basic Sciences & Applied Research. 2(4): 484-493.
- [34] Brans, J. P. and Mareschal, B. 2005. ORESTE methods, in Figueira, J., Greco, S. and Ehrgott, M. (Eds), Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, New York. NY. Pp: 163-198.
- [35] Goumas, M. and Lygerou, V. 2000. An extension of the ORESTE method for decision making in fuzzy environment: ranking of alternative energy exploitation. European Journal of Operational Research. 1(23): 606-613.

Authors Biography

1 Shahram Jamali- He is an Associate Professor leading the Autonomic Networking Group at the Department of Engineering, University of Mohaghegh Ardabili. Dr. Jamali received his M.Sc. and Ph.D. degree from the Dept. of Computer Engineering, Iran University of Science and Technology in 2001 and 2007, respectively. Since 2008, he is with Department of Computer Engineering, University of Mohaghegh Ardabil and has published more than 50 conference and journal papers. He is currently associate professor of computer engineering at University of Mohaghegh Ardabili (UMA), His research interests include Congestion control in computer networks, UPFC Controller design, Load balancing in Grid environment. Dr.Shahram jamali is Head of Computer Engineering and Information Technology Department for 2008-2014.

2 Khadije Hourali – She is received her B.Sc. in computer engineering from Shahrood University, Shahrood, Iran, in 2011 and her M.Sc. in computer engineering from Mohaghegh Ardabili University, Ardabil, Iran, in 2013. Her research interests Cloud Computing and Wireless Sensor Networks.