International Journal of Advanced Research in Engineering and Applied Sciences

ISSN: 2278-6252
Impact Factor: 5.795

# A NOVEL APPROACH TO IMPROVE THE SLA AND ENERGY CONSUMPTION OF GRID NETWORKS

**Fatemeh Hourali***

**Samira Hourali****

**Abstract:** *In recent years, Modern data centers in grid computing are hosting a variety of advanced applications and the IT infrastructure due to the demand for computational power infrastructure that is used by applications, is growing rapidly. One of the most important objectives of the VM placement algorithm is locating optimized virtual machines on physical servers, So that the minimum number of physical servers to increase the overall performance of the network environment to be turned on. Putting efficient VMs in PMs (Physical Machines) in grid environment improves resources utilization and energy consumption. In this paper, we apply TOPSIS method to design an integrated VM placement algorithm, called TOPSIS VM Placement (TVMP) which can reduce the number of running PMs and energy consumption. Simulation results in GridSim environment show that the proposed algorithm is superior to existing algorithms in terms of migration, traffic cost, SLA and energy.*

***Keywords:*** *Energy Consumption, TOPSIS, Migration, VM Placement, Migrations, Service Level agreement (SLA), Energy Consumption.*

*Department of Electrical and Computer Engineering, Esfarayen University of Technology, Esfarayen, Iran

**Department of Electrical and Computer Engineering, Non Benefit University of Shahrood, Shahrood, Iran

## 1. INTRODUCTION

Grid computing is a new technology in academic world [1]. In a grid 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 [2]. On the other hand, dynamically balancing the load among the servers improves resource utility and the overall grid 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 grid environments are called as virtual machines (VMs).

Resource management of non-homogeneous hardware resources has been extensively studied [3]. Typically, a resource management system receives, queues, and finally matches user job requirements with the characteristics of the offered hardware. For instance, scheduling jobs in the Grid requires choosing an appropriate Grid site that complies with the user requirements. The advent of the grids has introduced very strict abstractions over the physical resources. IaaS grids restrict users from specifying the exact physical resources to be consumed when instantiating virtual machines (VMs). Grid consumers remain agnostic of the underlying physical infrastructure. Only high-level resource requirements such as CPU and RAM are stated in user requests.

Grid architectures exploit virtualization technologies to provide multiple VMs, possibly belonging to different service providers, on the same host. power-efficient management infrastructures need to carefully address the placement problem since the overall DC power consumption is strictly related with the number of powered-on hosts, networking elements state, and so on [4]. VM placement has to consider many heterogeneous constraints, spanning from service SLAs to limited computational resources, and to enforce anti-co-location VM constraints aimed to avoid that the failure of a single host deeply impacts service reliability (for instance, when all the web servers giving access to the same service are on one host.

From a more formal perspective, given a set of hosts with limited resources and a set of VMs with well-defined resource requirements, the grid management infrastructure has to

decide both final VM-to-host mappings and alternative VM relocation plan. In fact, VM placement is a continuous process where the management infrastructure periodically checks if a better placement exists and, if that is the case, reconfigures the current configuration through the definition of a new relocation plan. Modifying a pre-existing VM configuration may involve several VM migrations and networks reconfiguration and that, in its turn, introduces an improvement problem, namely, the design of a VM relocation plan useful to bring the DC from an initial state to new desired one. For the sake of clarity, and in order to highlight main optimization goals and constrains, in the remainder we specifically focus on the placement function only.

The VM placement function has to model and consider all resource constraints because each host has limited resources in terms of CPU, memory, and I/O operations, and the placement process has to consider them to prevent solutions that would require more resources than available ones. In addition, more complex constraints associated with the whole DC have to be considered for the sake of placement feasibility. On the one hand, it could be impossible to keep powered-on all the hosts due to limitations on aggregated power consumption and cooling systems. On the other hand, the DC network infrastructure can introduce tight constraints on the placement due to limited networking capacities; in particular, different aspects, including time-varying traffic demand, network topology, and dynamic multipath routing schema, make the definition and the enforcement of such constraints very difficult in the general case [5, 6].

## 2. RELATED WORKS

In the last few years, many research works addressed the problem of power-aware VM placement in grid systems and several industrial are available for energy-efficient dynamic VM provisioning. At the same time, to the best of our knowledge, these solutions do not exploit full awareness of (i) running services, (ii) service providers' SLA, and (iii) DC status to select the optimal power-efficient placement and to guide the VM reallocation process.

In [13], authors focus on power-efficient VM placement problem by proposing a two-phase approach. The first phase finds the optimal number of VMs to meet service providers' SLA, while the second on details VM-to-host mapping with the main goal of reducing powered-on hosts. The management infrastructure exploits service-supplied utility functions to

evaluate the proposed VM placement; hence, different types of services can be supported by properly adjusting supplied utility functions.

Mistral is a novel solution to optimize VM placement while reducing power consumption [14]. Mistral considers transient costs associated with runtime reconfigurations and exploits dynamically updated indicators on placement stability to carefully tailor DC reconfiguration; toward that goal, authors introduce filtering and forecasting techniques to better weight management decisions. In addition, Mistral exploits A*-search techniques to find suitable plans of reconfiguration actions needed to reach the new placement. Considering the notion of SLA adopted by Mistral, i.e., a target mean response time for each service, we can safely assume that authors target online services.

In [6], authors study the problem of VM placement with goal of reducing the aggregate traffic into Dc. In [9], authors consider applications made by computation and a storage part and propose a new optimization problem to place them while reducing runtime DC network traffic. In [5], authors propose a new network-aware placement problem that places VMs so to reduce the worst-case load ratio over all network cuts with the main goal of increasing placement stability with time-varying traffic demands; some seminal works are starting to use equivalent capacity notions to co-locate VMs with uncorrelated traffic demands.

In [7] authors consider the placement of VMs and applications in grid environments, while accounting for SLA violations and migration costs; the propose several strategies, based on host power efficiency and First Fit decreasing heuristics, to reach proper trade-offs between performance and power consumption. A large set of shown experimental results confirms that there are interfaces between VM live migrations and supports the technical validity of pMapper. However to the best of our knowledge, authors do not focus on specific service type, but rather assume the existence of a generic function to evaluate that the computed placement is feasible and compiles with agreed SLAs.

In [8], authors focus on CPU-bound batch services: VM placement considers computation only and exploits CPU requirements, while it does not consider any constraint on user interactions. In [12], authors focus on CPU-bound online services by presenting a new optimization model to find the minimum number of physical hosts required to execute a specific workload. The workloads consist of different jobs, each one characterized by a number of requests/second to be processes. Authors assume that it is possible to relate

each request with a number of instructions to be executed and use that indicator to handle the placement phase. It is worth noting that, apart from retrieving the minimum number of powered-on hosts, the model proposed by authors also finds the optimal value of the host CPU frequencies. As a possible drawback, authors make the assumption that grid jobs can be arbitrarily split between hosts to meet the required number of requests/second; although that can be viable in some scenarios, it does not fit well in the general VM allocation problem.

## 3. TOPSIS VM PLACEMENT ALGORITHM (TVMP):

### 3.1 TOPSIS Method

The TOPSIS (Technique for Order Preference by Similarity to the Ideal Solution) method, at the first stage, consists of the composition of the Decision Matrix A with the values of attributes (criteria), and the construction of the normalised Decision Matrix R based upon matrix A. The elements of matrix R are computed as $r_{ij} = x_{ij} / (\sum_{i=1}^{M} x_{ij}^2)^{1/2}$, where xij is the value of the jth criterion for the ith alternative, and is, as in equation (1), an element of Decision Matrix A. The weighted normalised decision matrix is obtained by using the normalised decision matrix R and weights assigned to criteria as $V[v_{ij}] = [w_j * r_{ij}]$.

At the second stage, the ideal (fictitious best) solution $A^+$ and the negative-ideal (fictitious worst) solution $A^-$, are determined, respectively, as follows:

$$A^+ = \{(\max_i v_{ij} \mid j \in J_1), (\min_i v_{ij} \mid j \in J_2) \mid i = 1,2,\ldots,N\} \quad (1)$$
$$= \{v_1^+, v_2^+, \ldots, v_j^+, \ldots, v_M^+\}$$

$$A^- = \{(\min_i v_{ij} \mid j \in J_1), (\max_i v_{ij} \mid j \in J_2) \mid i = 1,2,\ldots,N\} \quad (2)$$
$$= \{v_1^-, v_2^-, \ldots, v_j^-, \ldots, v_M^-\}$$

Where $J_1$ is associated with the benefit and $J_2$ with the cost criteria. Consequently, the Euclidean distance of each alternative from the overall ideal and negative deal solution is determined, respectively, as follows:

$$S_i^+ = [\sum_{j=1}^{M} (v_{ij} - v_j^*)]^{1/2} \quad and \quad S_i^- = [\sum_{j=1}^{M} (v_{ij} - v_j^-)]^{1/2} \quad for\ i = 1,2,\ldots,N \quad (3)$$

Where all symbols are as above. The relative closeness of each alternative to the ideal solution is computed as ratio $C_i^+ = S_i^- / (S_i^+ + S_i^-)$ for i= 1,2,…,n . Finally, the alternative with the highest value of $Ci^+$ is selected as the preferable (best) one [10-11].

### 3.2 A Model for Grid Environment

The defined space of grid computing consists of K clusters for processing (service) .m is the number of physical machines that is such as $P = \{P_1, P_2, \dots, P_n\}$.n is the number of virtual machine per physical server that is shown as $VM\ (R_f) = \{VM_1, VM_2, \dots, VM_n\}$, $f \in [1, m]$. Each physical machine in the grid computing environment has four characteristics that include$(CPU_i, Mem_i, IO_i, BW_i)$, where $CPU_i$ which is based on MI represents the CPU capacity of $PM_i$, 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 B/Sec, and bandwidth of $PM_i$ which is based on (MB/Sec).

For each virtual machine four characteristics are defined as $VM_j = (cpu_j, m_j, io_j, b_j)$. $cpu_j$ is processing power of each virtual machine that is the number of instructions executed by each processing elements of source in terms of million per second (MIPS). $m_j$ and $io_j$, respectively represents the rate of utilization of memory and input/output, which is calculated based megabyte per second (B /S), $b_j$ represent amount of bandwidth requirement for $VM_j$. Weights of VM's characteristics are calculated as bellow by eq. 4,5,6 and 7, sum of this weight must be equal to 1:

$$\text{CPU\_weight} = \frac{cpu_j}{cpu_j + m_j + io_j + b_j} \quad (4)$$

$$\text{Memory\_weight} = \frac{m_j}{cpu_j + m_j + io_j + b_j} \quad (5)$$

$$IO\_\text{weight} = \frac{io_j}{cpu_j + m_j + io_j + b_j} \quad (6)$$

$$\text{Bandwidth\_weight} = \frac{b_j}{cpu_j + m_j + io_j + b_j} \quad (7)$$

### 3.3 TVMP Algorithm

On each PM, the virtual infrastructure manager create server table, in this table, information of servers providing a service is existing. With addition of each physical server to grid, this table is updated. Servers in the grid can provide one or more services simultaneously. If the server can provide only one service this means that all existing VMs on that server are located in a cluster and if a server simultaneously provides more than one service this means that as the number of services provided by this server, groups of VMs are existing. The proposed algorithm is as follows:

ISSN: 2278-6252
Impact Factor: 5.795

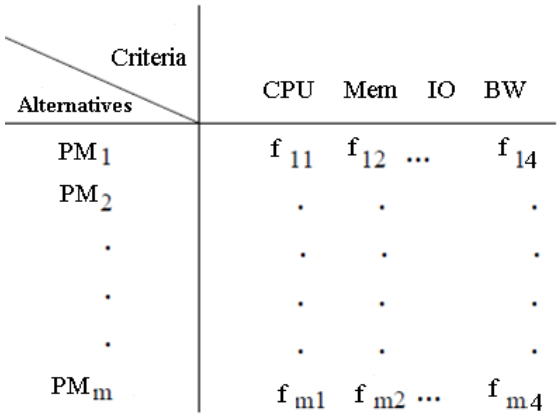


**Fig 1: PM's Decision Matrix**

First the overloaded servers are removed from the table, then by considering the characteristics of each PM (Fig.1), weights of VM's requirements, using TOPSIS 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.

For example if we have 6 PMs with various characteristics as shown in table 1, and the weights of desired VM's criteria is W={0.411, 0.29, 0.179, 0.12}, it means the values of CPU, Memory, I/O, bandwidth respectively is 0.411, 0.29, 0.179 and 0.12 and sum of this weights is equal to 1. Suitable PM (PM2) for this VM is obtained as follows (Figures 2, 3 and 4) from TOPSIS method:

**Table 1: PM's characteristics**

| **Criteria** / **PMs** | CPU (MPIS) | Memory (MB) | I/O (B/sec) | Bandwidth (MB/sec) |
|---|---|---|---|---|
| PM1 | 10000 | 2048 | 200 | 300 |
| PM2 | 20000 | 1024 | 300 | 100 |
| PM3 | 10000 | 2048 | 100 | 200 |
| PM4 | 10000 | 2048 | 400 | 100 |
| PM5 | 20000 | 1024 | 200 | 100 |
| PM6 | 10000 | 1024 | 100 | 200 |

ISSN: 2278-6252
Impact Factor: 5.795

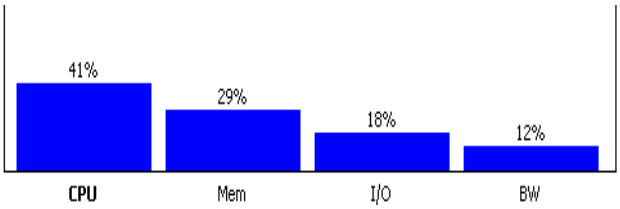


**Fig 2: Weights of VM's criteria**

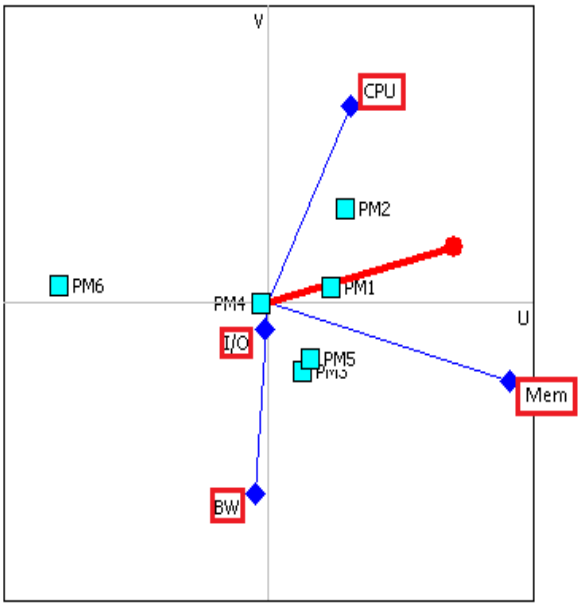


**Fig 3: Distance of each PM form each criterion**

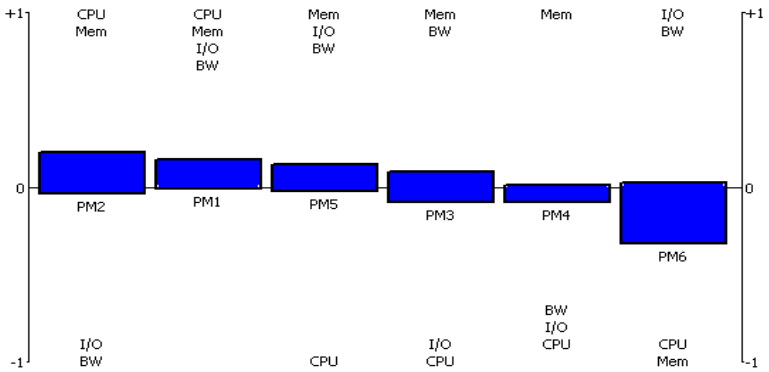


**Fig 4: PM's status**

As shown in Fig.5 According to TOPSIS method the best PM for allocate considered VM is PM2.

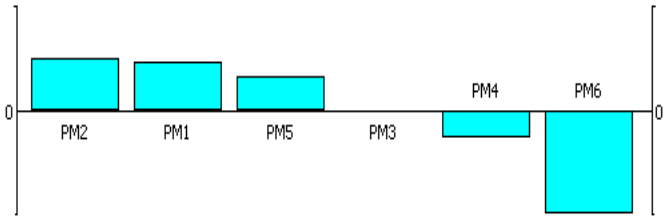


**Fig 5: Final PM's ranking with TOPSIS**

## 4. EXPERIMENTAL RESULTS

We developed a simulator to evaluate the effectiveness of TVMP algorithm. It simulates a grid network topology with VMs, hosts, broker and links as a discrete event simulation. Each server is assumed to host at most one VM. VMs run applications that generate network traffic to other VMs, and VMs can migrate from one node to the other. At each time step, network traffic generated by VMs (denoted by an entry in the input traffic matrix) is read and corresponding VMs are mapped to hosts. The corresponding route between the hosts is also calculated based on the consumed network power, the network topology and available link capacities. At each time step of the simulation, we compute power consumption. A new input matrix, that represents the network traffic at that time stamp, is used at each time step of the simulation.

We evaluate the efficacy of our algorithm in a simulation environment using GridSim [15]. The simulated data center 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 [19]. The running period is 24 hours to simulate the diurnal pattern of a communication network [17]. The VM and PM configurations are as same as [2], plus that all PMs in a partition have the same configuration.

As in [16, 19] we also use FNSS 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 ($\sigma$) equals to 4 to form an environment where some VMs are linked with high traffic [7].

These static volumes then added a zero-mean normal fluctuation value. According to [18], the relation between the standard deviation of this fluctuation ($\sigma'$) and the mean traffic volumes is:

$$\overline{x_{ij}(t)} = \psi \sigma_{ij}^{'\gamma} \quad (8)$$

As in [19] we also chose = 0.8 and log $\psi$ = -0.33 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, VMs are classifying into three categories: network- intensive, CPU-network balance and CPU-intensive servers [19]. The CPU utilization of each VM is then generated correspond to which category it belongs to [19]. In the simulation, the experiment results when TVMP, TPVMP, Traffic-only and Energy-only algorithms are applied are compared (Figures.6, 7, 8 and 9).According to Fig.9, TVMP respectively saves about 9% and 7% of traffic cost compared to Energy-only and TPVMP algorithms. Proposed method also saves about 35% SLA violation when the number of VM is not so high. When the number of VM is high, there are not many available positions for VM migration, thus cause high energy consumption and SLA violation, but reduce the number of migrations for all algorithms [19]. The number of migrations and energy consumption of proposed method is less than TPVMP, Traffic-only and Energy-only algorithms. As shown in figure 6, the Traffic-only and TVMP algorithms respectively save about 33% and 30% of traffic cost, also as shown in figures 8 and 9, Traffic-only, TPVMP and TVMP algorithms respectively has 20%, 15%, 9% number of migrations and 11%, 10%, 9% energy consumption.

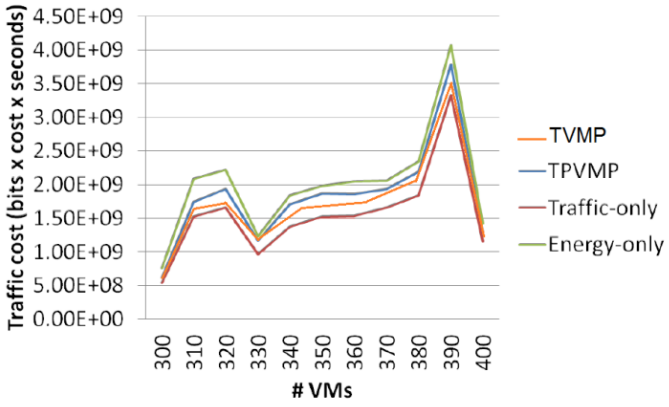


**Fig 6. Traffic cost**

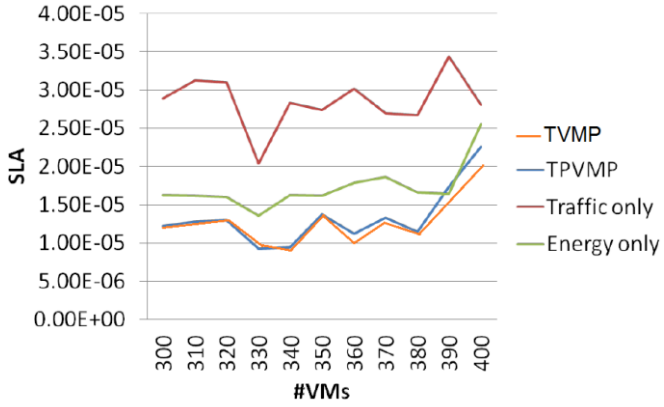


**Fig 7. SLA**

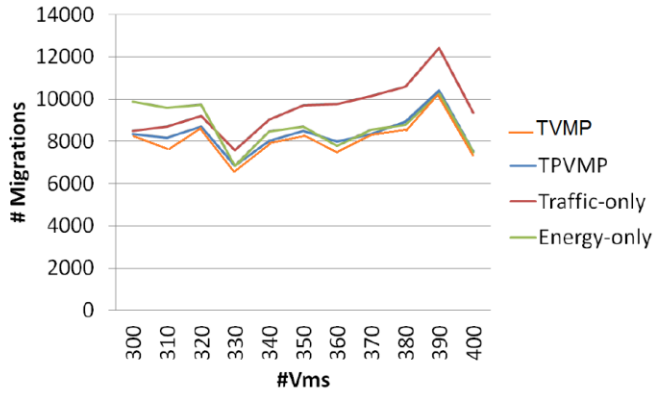


**Fig 8. Migrations**

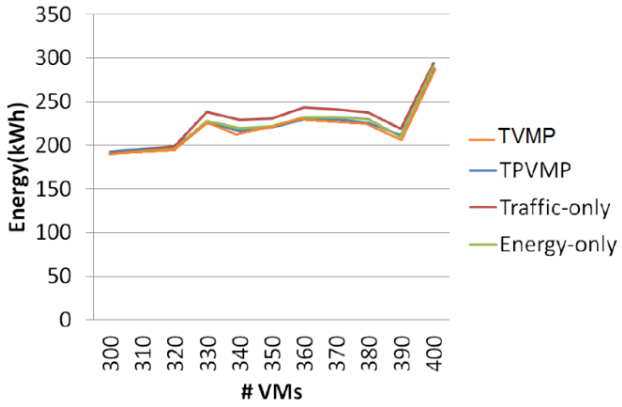


**Fig 9. Energy**

## 5. CONCLUSION

This paper presents a VM placement algorithm with using TOPSIS method. This allocation is a choice between existing PMs for considered VM, based on the weights of VM's criteria and all PM's characteristics. Then, the virtual machines follow a certain sequence to be placed at their preferable physical machines to achieve the profit maximization and heavy communicated virtual machines are hosted by physical machines that located close

together. In proposed algorithm virtual machines are consolidated on physical machines with high CPU usage per energy consumption. Result shows that our algorithm is superior in various performance metrics and produces better allocation result considering VM communication, energy consumption, VM's migration, SLA violation and cost.

## 6. REFERENCES

[1] Cisco Data Center Infrastructure 2.5 Design Guide. May 2008.

[2] Anton Beloglazov and Rajkumar Buyya.. 2012 ."Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in grid data centers". Concurr. Comput: Pract. Exper., 24(13):1397-1420, September.

[3] Gierniak, M., Zaki, M.J., Li, W. 1997. Compile-Time scheduling Algorithms for a Heterogeneous Network of Workstations. The Computer Journal 40(6), 356-372.

[4] Beloglazov, A. et al. 2010. A taxonomy and survey of energy-efficient data centers and Grid computing systems. Proceeding of CoRR.

[5] Biran, O. et al. 2012. A stable network-aware VM placement for Grid Systems. Proceedings of the IEE CCGride'12, Ottawa.

[6] Meng, X. et al. 2010. Improving the scalability of data center networks with traffic-aware virtual machine placement. Proceedings of the 29 th Conference on Information Communications (INFOCOM'10).

[7] Ankit Anand, 2013. Adaptive Virtual Machine Placement supporting performance SLAs. A Project Report submitted in partial ful_lment of the requirements for the Degree of Master of Technology In Computational Science, Super Computer Education and Research Centre Indian Institute of Science Bangalore-560 012 (INDIA), 10-23.

[8] Aaron Carroll, Gernot Heiser, 2010. An Analysis of Power Consumption in a Smartphone, USENIXATC'10 Proceedings of the 2010 USENIX conference on USENIX annual technical conference ,21-25.

[9] Ching-Chi,Pangffeng,Jan-jan Wu, 2011. Energy-Aware Virtual machine Dynamic provision and scheduling for grid computing. In Proc. of the 2011 IEEE 4th International Conference on grid computing.

[10] Zeleny, M, Multiple Criteria Decision Making. McGraw-Hill, New York , 1982.

[11] 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), pages 1-8.

[12] Calcavecchia, N.M. ; Dipt. di Elettron. e Inf., Politec. di Milano, Milan, Italy ; Biran, O. ; Hadad, E. ; Moatti, Y. 2012. VM Placement Strategies for Grid Scenarios. Grid Computing (GRID), 2012 IEEE 5th International Conference on, 852 – 859.

[13] Hwang, L.C., Yoon K. 1981. Multi Attribute Decision-Making: A Methods and Applications, Lecture Series in Economics and Mathematical Systems, Springer-Verlag, Berlin, Germany.

[14] G. Jung, M.A. Hiltunen, K.R. Joshi, R. D. Schlichting, C.Pu, 2010. Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Grid Infrastructures, In Proc. Of the IEEE 30th International Conference on Distributed Computing Systems (ICDCS'10), IEEE Press, Jun, 62-73.

[15] 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):23-50.

[16] A. 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, pages 1-12.

[17] Atiq Rehman and M. Hussain. 2011. Efficient cloud data condentiality for daas. International Journal of Advanced Science and Technology, 35:1-10, October.

[18] Ismael Solis Moreno and Jie Xu. 2011. Energy-efficiency in cloud computing environments: Towards energy savings without performance degradation. IJCAC, 1(1):17-33.

[19] Hieu Trong Vu, Soonwook Hwang. 2014. A Traffic and Power-aware Algorithm for Virtual Machine Placement in Cloud Data Center. International Journal of Grid & Distributed Computing, Vol. 7 Issue 1, 350-355.