Computer Engineering and Applications Vol. 4, No. 2, June 2015

Energy Efficient Virtual Machine Placement Algorithm With Balanced Resource Utilization Based on Priority of Resources

Amin Rahimi*, Lelli Mohammad Khanli, Saeid Pashazadeh

Department of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran
**amin.rahimi.1365@gmail.com, l-khanli@tabrizu.ac.ir, pashazadeh@tabrizu.ac.ir*

ABSTRACT

The increasing energy consumption has become a major concern in cloud computing due to its cost and environmental damage. Virtual Machine placement algorithms have been proven to be very effective in increasing energy efficiency and thus reducing the costs. In this paper we have introduced a new priority routing VM placement algorithm and have compared it with PABFD (power-aware best fit decreasing) on CoMon dataset using CloudSim for simulation. Our experiments show the superiority of our new method with regards to energy consumption and level of SLA violations measures and prove that priority routing VM placement algorithms can be effectively utilized to increase energy efficiency in the clouds.

Keywords: Cloud computing, VM placement, Energy efficient

1. INTRODUCTION

The rapid growth in the need for ubiquitous computing utilities has resulted in the realization of a new computing model called *cloud computing* in which consumers from all over the world access the shared computing resources in an on-demand *pay-as-yougo* model. The computing resources are usually available in the form of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The could computing model requires an adaptive resource allocation management to adopt with the workload of several applications with different resource allocation requirements to provide quality of service for each application and to utilize data center resources efficiently. One solution is to use Virtual Machines (VMs) over Physical Machines (PMs) for resource consolidation and environment isolation. The focus of cloud research is shifting from performance optimization to energy efficiency due to the rapid increase in energy bills and environmental damage. The total energy bill for data centers was $11.5 billion in 2010 and this cost is estimated to double every five year [1]. Existing works [2, 3, 4] show that energy efficiency is highly dependent on the number of running PMs. [5] reports that server utilization is usually about 10-50% resulting in lots of idle machines in the cloud. An idle machine consumes as much as 70% power as its peak so minimizing the number of active PMs by powering off or low the idle machines and utilizing the active ones more efficiently can reduce the power consumption [6]. Live migration of VMs partitions PMs into a highly utilized group and an idle group that can be put in low power mode (e.g. sleep, power off, hibernate) to eliminate idle power consumption . This results in huge savings in power consumption, cooling requirements, energy bills, Carbon footprint and in the same time assures quality of service for applications. The workload of applications highly varies in time so the dynamic resource consolidation should migrate VMs in order to adopt to this change. Excessive migration of VMs results in delays in response time of applications which damages the quality of service. To overcome this, a VM placement algorithm should maximize utilization on the least possible number of PMs in order to reduce energy consumption and in the same time should limit the number of VM migrations to assure quality of service which makes the task very challenging yet very rewarding [2]. VM consolidation problem divide into four sub-probblems:

a) Host overload detection
b) Host underload detection
c) VM selection
d) VM placement

When host overload or underload detected, some VMs should be migrated from that host(s). If host overload detected, VM selection algorithm choose some VMs to migrated from that host. If host underload detected all the VMs should be migrated from that host. Process of placing VMs from migration list to PMs is called VM placement.

In this paper we propose a new VM placement method based on priority routing. We compare our method with PABFD [3] in terms of power consumption, number of migrations, SLA (Service Level Agreement), load balancing and ESV metric described in 4.4. Our main contributions in this paper are: a) The introduction of a new priority routing based VM placement algorithm. b) to show that priority routing VM placement method outperforms PABFD in terms of power consumption, load balancing, number of migrations, SLA violations and ESV performance metric. Empirical evidence that PRVM are generally superior and can be used to achieve energy efficiency in the clouds.

2. RELATED WORK

Several methods have been proposed for VM placement. [7] proposed an architecture design for automatic VM placement lead to get better resource utilization and hostsget less overloaded.

[8] proposed a network-aware VM placement to avoid network latency and congestion situations caused by long data transfer time to maintain application performance. The main idea is to place VMs in PMs that minimize data access time. to achieve this purpose they calculate data access time of PMs for each data and create a data access time matrix and choose PM, which has minimum data access time to host that application.

[9] proposed a traffic-aware VM placement by considering traffic patterns between VMs. VMs which has large mutual bandwidth usage are placed in PMs that close together.

[10] proposed a VM placement algorithm to optimize problem of joint VM placement and routing to minimize the network congestion by exploiting multi path routing capability and combine it with VM placement.

[3] has developed a power-aware VM placement named PABFD (power-aware best fit decreasing). PABFD place VM in host that will increase least power consumption after allocation. [4] investigate the VM placement algorithm for HPC

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

data centers. Proposed algorithm is like PABFD with considering memory in power model.

Our proposed VM placement algorithm considering reducing resource usage variances to achieve better load balancing by placing VMs in PMs with regard to their need of resources and reduce the overlapping of resources by placing VMs with same resource usage in separate PMs.

3. PRIORITY ROUTING VM PLACEMENT

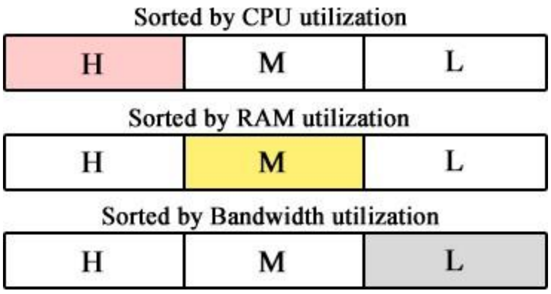
Priority routing VM palcement (PRVP) algorithm have proved to be very effective in energy efficient cloud computing. The main idea is to categorize VMs with regard to their resource utilization and categorize PMs based on their resource availability then place VMs to the PMs with the most similar category.

3.1 CATEGORIZE VMS AND PMS

One of the weaknesses of previous VM placement algorithms is that not considering priority of resources. For example CPU has higher priority compare to RAM so a good VM placement algorithm should consider load balancing of resources based on their priority. another problem of VM placement algorithms is overlapping of resources so we should not put VMs with same resource usage in same PM to minimize resource overlapping of resource usage by VMs so we introduce a VM placement algorithm To addressing this problems. To solve this problem we should first categorize VMs and PMs. For categorizing VMs we sort them in decreasing order based on their last request of resources(CPU, RAM, Bandwidth), Then partition the lists into three equal parts (High, Medium, Low). According to position of a VM in each three lists we labeled that with three character string CRB (CPU jRAM jBandwidth):

$$CRB \in \{HHH, HHM, HHL, HMH, \ldots, LLL\}.$$

For example VMHML has a high CPU utilization, medium RAM utilization and low bandwidth utilization compare to other VMs in the migration list (Figure 1). We do the same procedure for PMs based on their resource availability, so PMHML has a high CPU availability, Medium RAM availability and low bandwidth availability compare to other hosts.



ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

FIGURE 1. Position of VM_{HML} in sorted lists

3.2 CREATING THE ROUTING TABLE

With regard to each VM category (label), we create a path of PMs for placement of VMs. We create the routing paths from the PMs with the most similar categories to least one with considering the priority of resources. CPU has the highest priority and bandwidth has the lowest. To changing from one category to other, we change the bandwidth category character first, so bandwidth has the lowest priority. For example the routing path for VMHLM is:

$$PM_{HLM}, PM_{HLH}, PM_{HLL}, PM_{HMM}, \ldots, PM_{LHL}.$$

3.2.1 DATA STRUCTURE OF THE ROUTING TABLE

For creating the routing table, we use a HashMap. The keys are the categories and value consist of a list of PMs (routing path) according to VM category and a counter. Counter indicates next PM for placement of the VM, because of not to place VMs with same category in same PM.We do this for better load balancing and avoid performance degradation due to overlapping of resource usage by VMs. Also load balancing lead to avoid performance degradation due to high utilization.

3.3 VM PLACEMENT

When host overloaded or host underloaded detected, some VMs should be migrated from the detected hosts. First we create path of hosts for each VM category from the most similar category to least similar category as mentioned in Section 3.2. Then for each VM we calculate the VM category based on the VM resource utilization described in Section 3.1. After that we get the routing path regard to VM category and read the next-host-counter for placement of the VM (all counters initiated to zero at first). Then check if host has enough resource for placement of the VM, if host has enough resource increase next-host-counter by one and return the allocated host. If host does not have enough resource for the VM, we increase the next-host-counter and get the next host in the path. Algorithm 1 shows the pseudo-code of PRVP algorithm.

4. PERFORMANCE EVALUATION

In this section we compare proposed algorithm with PABFD. We test algorithms with extensive simulations with real world traces of CoMon project and show the comparative results between them.

4.1 WORKLOAD DATA

The data gathered from the CoMon project workload traces. CoMon project is a monitoring system for PlanetLab [11]. Workload data consist of CPU utilization of

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

VMs from PMs located at different places in 10 randomly days. The data collected every 5 minutes. Table 1 shows the workload details.

4.2 POWER MODEL

It has proven that the power consumption of a host has a linear relationship with its CPU utilization [12]. For multi-core CPUs, CPU utilization of host is sum of

```
getNewVmPlacement:
input : vmsToMigrate, excludedHosts, hosts
output: A map of VMs to hosts

migrationMap emptyMap;
vCu ← sortByCpuUtilizationDec(vmsToMigrate);
vRu ← sortByRamUtilizationDec(vmsToMigrate);
vBu ← sortByBwUtilizationDec(vmsToMigrate);
hCa ← sortByAvailabeMipsDec(hosts);
hRa ← sortByAvailabeRamDec(hosts);
hBa sortByAvailabeBwDec(hosts);
typeHostListMap getMapByResAvail(hCa; hRa; hBa);
routingMap createRoutingMap(typeHostListMap);
for vm ϵ vCu do
  vmtype getVmTypeByUtilizations(vCu; vRu; vBu; vm);
  allocatedHost findHostForVm(vm; excludedHosts;
vmType;routingMap);
  if allocatedHost 6= NULL then
    migrationMap[vm] allocatedHost;
  end
end
return migrationMap;

findHostForVm:
input : vm, excludedHosts, vmType, routingMap
output: allocatedHost

allocatedHost NULL;
routingHosts routingMap[vmtype] ;
hostsNum sizeof(routingHosts) ;
counter   0 ;
while counter 6= hostsNum do
  host getNextHost(routingHosts);
  if host 2 excludedHosts then
    continue ;
  end
  if host has enough resource for vm then
    allocatedHost host;
    break;
  end
```

```
    counter   counter + 1 ;
end
return allocatedHost;
```

Algorithm 1: Priority routing VM placement

TABLE 1.
Workload data details from CoMon project.

Workload date	Number of VMs
03/03/2011	1052
06/03/2011	898
09/03/2011	1061
22/03/2011	1516
25/03/2011	1078
03/04/2011	1463
09/04/2011	1358
11/04/2011	1233
12/04/2011	1054
20/04/2011	1033

utilization of cores. Some servers power consumption are modeled in Cloudsim [13] based on the real power consumption of the servers in different CPU load levels (e.g. HP ProLiant G4, HP ProLiant G5). Each host with CPU utilization greater than zero is calculated for power consumption. Table 2 shows HP ProLiant G4 and HP ProLiant G5 power consumption by amount of CPU utilization.

TABLE 2.
Power consumption of servers by amount of CPU utilization in Watts.

CPU Load	HP ProLiant G4	HP ProLiant G5
0%	86	93.7
10%	89.4	97
20%	92.6	101
30%	96	105
40%	99.5	110
50%	102	116
60%	106	121
70%	108	125
80%	112	129
90%	114	133
100%	117	135

4.3 SLA VIOLATION METRIC

Service-level agreements (SLAs) are represent the QoS requirements in cloud computing environments. Cloudsim combined two metrics for SLA violation metric,

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Performance degradation due to host overload, Overload Time Fraction (OTF) and performance degradation due to migration of the VMs (PDM). SLAV calculated by:

$$SLAV = OTF \cdot PDM. \qquad (1)$$

4.4 PERFORMANCE METRIC

In cloud computing environments in addition to reducing energy consumption, reducing the level of SLA violations must be considered. A. Beloglazov et al. [3] proposed a performance metric by combination of energy consumption and SLA violations. They define ESV as:

$$ESV = E \cdot SLAV. \qquad (2)$$

4.5 PABFD ALGORITHM

PABFD sort all VMs by CPU utilization in decreasing order then search for the PM that will increase least power consumption after allocation. This is the default placement algorithm in Cloudsim.

4.6 SIMULATION SETUP

For simulation purpose we use the latest version of the Cloudsim (3.0.3). In simulation we use 1000 servers from two PM types, HP ProLiant ML110 G4 and HP ProLiant ML110 G5. We use stochastic utilization model for RAM and bandwidth so we choose 80 GB for RAM and 10 Gbit/s for bandwidth for each PM. Table 3 shows PMs characteristics. For VMs we use four Amazon EC2s VM instances [14] (HighCPU medium, extra-large, small, micro) Table 4 shows VM instance types. We used IQR (Inter Quartile Range) with safety parameter of 1.5, for host overload detection policy and MMT (Minimum Migration Time) for VM selection policy.

TABLE 3.
Hosts characteristics.

Host	**HP ProLiant G4**	**HP ProLiant G5**
CPU	1x Xeon 3040 (1860 MHz)	1x Xeon 3075 (2660 MHz)
Cores	2	2
RAM (GB)	80	80
Bandwidth(Gbits/s)	10	10

TABLE 4.
VM instance type.

VM Type	**High-CPU Medium**	**Extra Large**	**Small**	**micro**
MIPS	2500	2000	1000	500

RAM (GB)	0.85	3.75	1.7	0.633
Bandwidth (Mbit/s)	100	100	100	100

4.7 SIMULATION RESULT

We compare PABFD with our algorithm by workload data mentioned in Section 2.1. Figure 2 shows number of VMs in workload traces dates. Figure 3 shows total power consumption of data center by using PABFD and PRVM algorithms. Results shows our proposed algorithm reduce energy consumption 26% more than PABFD. Figure 4 shows number of active hosts after consolidation. Result shows by using PRVM we have on average 33% fewer active host after consolidation compare to PABFD. Figure 5 shows number of migrations. Results shows PRVM reduced number of migrations 44% compare to PABFD.

Figures 6, 7 and 8 shows resource utilization variance of hosts after consolidation by using PRVM and PABFD. The results shows PRVM has lower resource utilization variance compare to PABFD so our proposed algorithm has better load balancing. Results indicates that load balancing of resources are relative to their priority on our

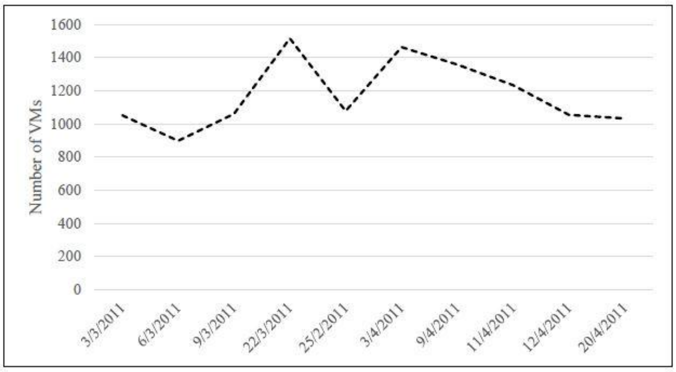


FIGURE 2. Number of VMs in workload traces dates.

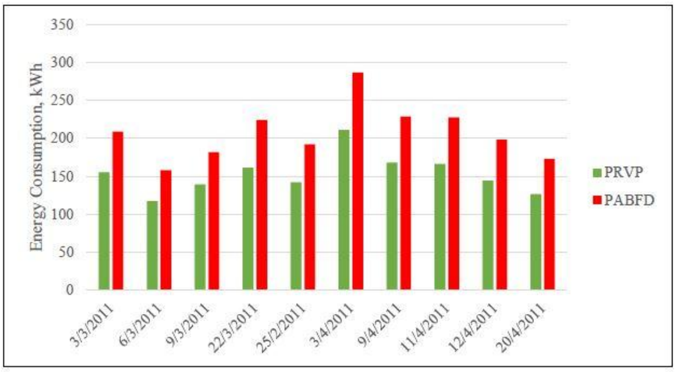


FIGURE 3. Energy consumption (kWh)

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

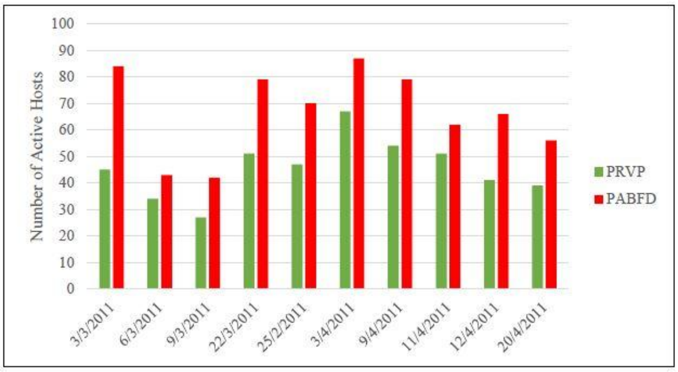


FIGURE 4. Number of active hosts after consolidation

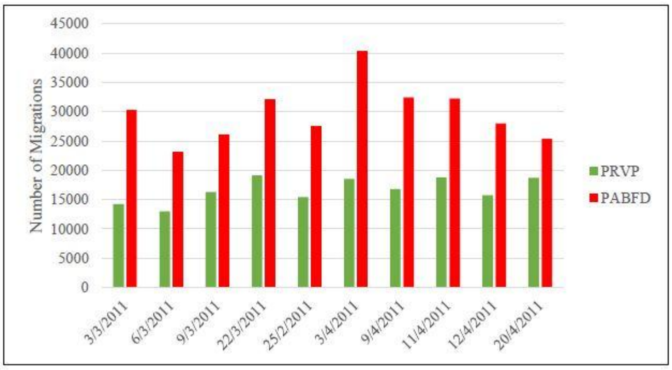


FIGURE 5. Number of migration.

proposed algorithm. CPU utilization is better balanced then RAM and bandwidth is less balanced because it has the lowest priority. We believed in cloud environments with more variety of VMs, bandwidth will better load balanced due to more host resource availability categories.

Figure 9 shows comparison of algorithms with regard to SLA violations. Results shows PRVM significantly reduced SLA violations compare to PABFD. SLA violations of our algorithm is 32% of PABFD SLA violations because of better load balancing lead to less host overload situations so OTF metric has reduced and placing VMs in PMs based on availability of resources has reduced DPM metric. Figure 10 shows comparison of algorithms with regard to ESV metric.

FIGURE 6. CPU utilization variance

FIGURE 7. RAM utilization variance

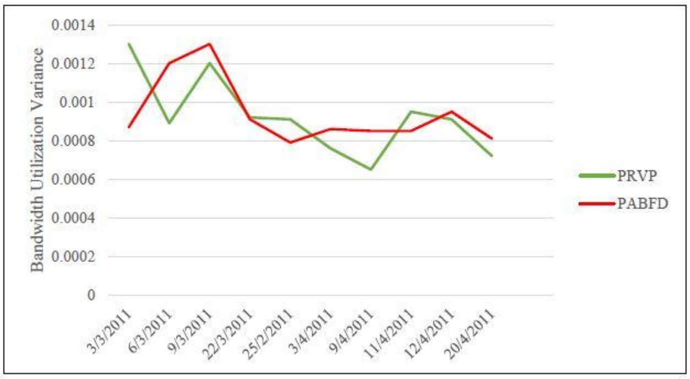


FIGURE 8. Bandwidth utilization variance

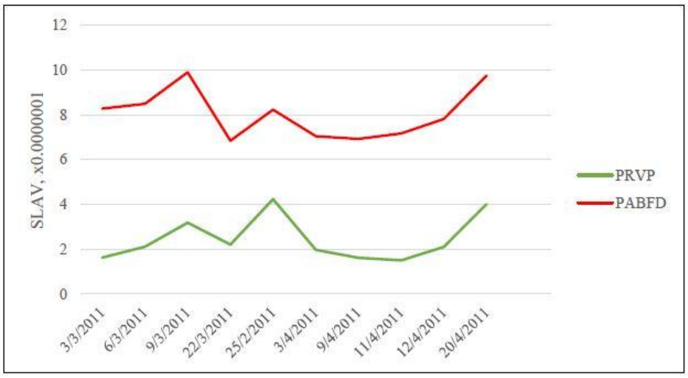


FIGURE 9. SLA violation metric

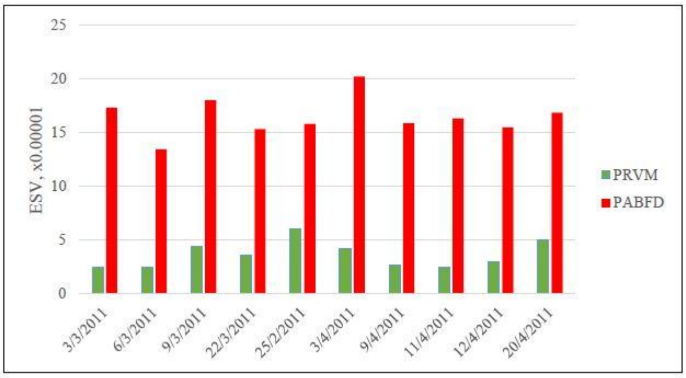


FIGURE 10. ESV metric

5. CONCLUSIONS

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

In this paper we introduced an online VM placement algorithm with goal of reducing power consumption of data center while keeping performance degradation low. Our main idea is to put VMs in PMs that VM resource usages are similar to PM resource availability. We proposed priority routing vm placement algorithm, which reduced energy consumption 26% more than PABFD. Proposed algorithm has 44% less migrations compare to PABFD. Our algorithm lead to better load balancing compare to PABFD so in term of SLA violations our algorithm results shows SLA violations of PRVM is 32% of SLA violations of PABFD. As aspect of performance metric (ESV) our algorithm has huge advantage over PABFD algorithm because PRVM reduced both energy consumption and SLA violations.

REFERENCES

[1]. C. Scroggie, "Predictions 2009: Symantecs craig scroggie," Website, 2008, http://searchstorage.techtarget.com.au/articles/28102-Predictions-29-Symantec-s-Craig-Scroggie.

[2]. A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[3]. A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[4]. I. Takouna, W. Dawoud, and C. Meinel, "Energy efficient scheduling of hpc-jobs on virtualize clusters using host and vm dynamic configuration," *ACM SIGOPS Operating Systems Review*, vol. 46, no. 2, pp. 19–27, 2012.

[5]. L. A. Barroso and U. Holzle, "The case for energy-proportional computing,"¨ *IEEE computer*, vol. 40, no. 12, pp. 33–37, 2007.

[6]. M. D. De Assuncao, J.-P. Gelas, L. Lefevre, and A.-C. Orgerie, "The green` grid5000: Instrumenting and using a grid with energy sensors," in *Remote Instrumentation for eScience and Related Aspects*. Springer, 2012, pp. 25–42.

[7]. C. Hyser, B. Mckee, R. Gardner, and B. J. Watson, "Autonomic virtual machine placement in the data center," *Hewlett Packard Laboratories, Tech. Rep. HPL2007-189*, pp. 2007–189, 2007.

[8]. J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*. IEEE, 2010, pp. 87–92.

[9]. 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*. IEEE, 2010, pp. 1–9.

[10]. J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, "Joint vm placement and routing for data center traffic engineering," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2876–2880.

[11]. K. Park and V. S. Pai, "Comon: a mostly-scalable monitoring system for planetlab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.

[12]. X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehousesized computer," in *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2. ACM, 2007, pp. 13–23.

[13]. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[14]. Amazon, "Amazon ec2 instance types," http://aws.amazon.com/ec2/.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)