

# A Traffic and Power-aware Algorithm for Virtual Machine Placement in Cloud Data Center

Hieu Trong Vu, Soonwook Hwang

*National Institute of Supercomputing and Networking  
Korea Institute of Science and Technology Information  
Daejeon 305-806, Korea  
{hieu,hwang}@kisti.re.kr*

## Abstract

*In cloud computing model, virtualized resources is provided to customers as a service over the Internet with reasonable price. This model effectively helps customers focus on their creative business because computing resources are provided on demand by cloud provider. Because of the convenient for cloud customers, the demand for cloud resource grows, thus make cloud data center enlarge and energy consumption becomes a critical issue for cloud provider as well as the carbon dioxide emission. In order to save energy, virtual machines are consolidated to reduce the number of servers required to host them. However, for applications that require communication among services, this migration can increase the latency which causes bad effect on their performance. This paper presents an algorithm that improve communication performance by reducing overall traffic cost of virtual machines and while energy by increasing CPU utilization.*

## 1 Introduction

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 [13, 22, 24].

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 [25] shows that service fragment can affect the data center network performance. In paper [15], 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 [11], 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 [5].

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 [6] . In order to save energy, VMs are consolidated to reduce the number of physical machines (PMs). Unused PMs are turned off 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 and background in Section II and Section III, respectively. In Section IV, 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 V. The last Section VI ends with conclusion and future works.

## 2 Related works

Virtual machine placement in cloud data center has been studied in different aspects. Paper [20] classifies the algorithms into two categories, power based approach and application QOS based approach. Each approach is further divide as dynamic and static placement. In dynamic placement, VMs can be migrated online among PMs, while in static method the mapping is not recomputed for a long period.

In paper [24], Wood et al. propose black-box and gray-box technique to monitor memory, CPU and network usage. Gray-box approach monitors VMs with OS and application level statistics, assume that data center owner has right to access VMs at application level, while black-box monitors VMs purely based on their behavior that can be observed from outside. Based on the collected information, VMs then can be migrated to suitable PMs to mitigate hotspots. This approach does not consider energy saving as well as data center topology.

Beloglazov et al [2] presents online heuristic for dynamic consolidation of VMs. Statistical method is used to calculate the upper CPU utilization threshold. If a PM is determined to be over-utilized, a VM is selected to be migrated from the PM to another suitable one. The statistical methods to determine over-utilized PMs and policies to select a VM to be migrated can be combined to form various strategies. On the other hand, if a PM is determined as under-utilized, all hosted VMs are selected to be migrated. The destination

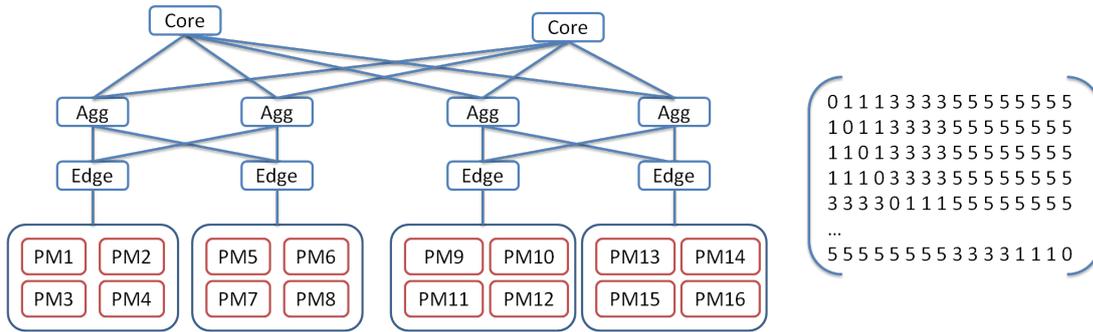


Figure 1: Three layer data center topology and cost matrix

PM is chosen in order to minimize power increment. This paper only considers energy consumption and does not refer to VM communication. In our paper, we use these strategies to determine over-utilized PMs and select VM to be migrated, and present new method to find destination PM in order to mitigate communication cost among VMs.

In paper [9], Dias et al. reduce traffic cost by grouping VMs with high communication into clusters and map these clusters to partitions of PMs using bin-packing algorithm. The algorithm simply maps the number of cores needed by each VM to the available cores in each server. However, in order to improve system utilization, one core usually be shared among several VMs with a QoS management mechanism.

In papers [8, 21], the destination PM is chosen so that the traffic cost of the migrated VM is minimized. With this algorithm, although the traffic cost for each VM can be minimized, it is not sure that the traffic cost of the whole system has the same effect.

### 3 Background and Motivation

#### 3.1 Data center topology

According to Cisco, most data centers use a three tier architecture [1]. At the bottom layer, physical hosts are grouped into partitions that connected to edge switches. Each edge switch is connected to one or more aggregation switch (for redundancy) at the middle layer. Finally, on the top layer there are core switches, used to interconnect aggregation switches and to outside data center.

In the layered topology, the number of switches participate in a connection are different depend on the position of the two servers. If they are connected to a ToR, then only one switch participate. In another situation, for 3 layer topology, a package may have to pass 5 switches before reaching the destination server. The number of switches participate in a connection is directly proportion to the possibility of network connection. Figure 1 illustrates a 3 tier data center with totally 16 PMs connected to 4 partitions. The cost matrix is based on [12].

#### 3.2 Traffic matrix simulation

In [12], traffic among VMs inside a data center follows uneven distribution. However, for each connection, the traffic is stable at large timescale. Benson et al. [4] trace SNMP

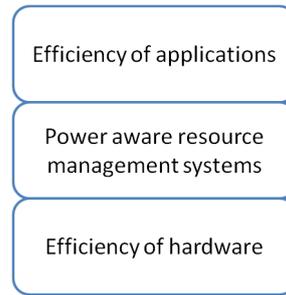


Figure 2: Energy consumption at different levels in computing systems.

logs from 19 data center with different topologies applications. They observe that traffic arrival rate follows ON/OFF pattern with a lognormal distribution. Paper [18] introduces Fast Network Simulation Setup (FNSS), a tool chain to parse or generate topologies and traffic matrices for data centers. The traffic matrices are classified as static or dynamic. A static traffic matrix reports the traffic volumes at a single point of time, whereas a dynamic matrix is a sequence of traffic volumes collected at different times. The static traffic volume follows lognormal distribution, and the dynamic traffic volume is the static traffic volume plus a zero-mean random variable.

### 3.3 Energy-aware VM placement

According to [13], energy-efficiency technologies, which consist of software policies executed on supported hardware, reduce the cost of computer system by improving resources usage while QoS is maintained.

Paper [3] classifies energy-saving technologies into three layers (Figure 2). The top layer includes energy saving algorithms implemented within applications. These technologies are used widely in mobile applications where battery capacity is limited. The middle layer consists of power aware management systems which control hardware. Finally, the bottom layer is hardware components that support energy saving technologies. Virtualization in data centers is one of the most important technologies for saving energy consumption which locates at the middle layer. Energy-aware mechanisms in cloud data centers could be classified as "dynamic processor scaling" (DPS) and "dynamic server pool resizing" (DSPR). DPS save energy by adjusting the server clock speed in accordance to workload variation. This approach depends on special hardware support and many uncontrolled environment factors. On the other hand, DSPR promises the most power saving by turning of idle or low- utilized servers using technologies such as Wake on LAN. However, due to server booting delay, future demand prediction is needed to assist the turning on/off decision-making process [10].

## 4 Virtual machine placement algorithm

Our algorithm targets to place VMs in a heterogeneous environment, which takes both energy consumption and traffic cost into account. In many papers [2, 8, 21], the general approach consist of three main steps:

- To check whether a PM is over or under-utilized.

CPU Util (%)	0	10	20	30	40	50	60	70	80	90	100
HP Proliant ML110 G4	86	86.9	92.6	96	99.5	102	106	108	112	114	117
HP Proliant ML110 G5	93.7	97	101	105	110	116	121	125	129	133	135

Table 1: Power consumption by the selected servers at different load levels in Watts

- If a PM is over-utilized, select one or more hosted VMs to be migrated. If it is under-utilized, all hosted VMs are selected.
- Find destination PMs for the selected VMs.

We use methods in [2] for the first two steps. and develop a new algorithm to find the destination PM to host selected VMs. Paper [2] simulates a heterogeneous environment which include two server configurations, HP Proliant ML110 G4 (Intel Xeon 3040, 2 cores x 1860 Mhz, 4 GB) and HP Proliant ML110 G5 (Intel Xeon 3075, 2 cores x 2660 MHz, 4GB). We also use these two profiles in our simulation. Table 1 illustrates each servers power consumption at different CPU utilizations levels. According to table 1, servers type HP Proliant ML 110 G5 consume around 1.1 times energy, while provide about 2660/1860 = 1.43 times CPU usage compared to ones type HP Proliant ML110 G4. If VMs are consolidated on PMs with higher CPU usage per energy consumption, defined as high capacity servers in our paper, the total energy consumption is reduced. Our algorithm takes both energy consumption and traffic cost into account. In general, VMs are consolidated on high capacity PMs, and heavy communicated VMs are hosted on PMs that located near each other. The detail algorithm is presented in the next sections.

#### 4.1 Traffic aware finding host for VM

Let  $v$  be the number of VMs, the set of traffic weight between two virtual machines is  $W = \{W_{ij} | 0 \leq i, j \leq v\}$ ,  $W_{ij} = 0$  if  $i = j$ , and the set of communication cost is  $C = \{C_{ij} | 0 \leq i, j \leq v\}$ ,  $C_{ij} = 0$  if VMs are hosted on a PM

Assume in time  $T$ , the traffic map inside a data center is stable, the traffic cost of the whole data center during time  $T$  is:

$$\sum_{i,j=0}^v W_{ij}C_{ij}T$$

For connections between each two virtual machines  $VM_i$  and  $VM_j$ , while  $W_{ij}$  is unchanged,  $C_{ij}$  is adjustable due to live migrations, thus the traffic cost can be reduced if the two VMs are located near to each other, for example, both of them are hosted on a PM, or on two PMs that connected to an edge switch. However, due to the limitation of computing resource, it is impossible to locate all connected VMs into a partition. In order to reduce the whole data center traffic cost, VMs with heavy weight communication should have higher priority to be located close to each other, while ones with low weight communication have lower priority.

However, to host two heavy weight communication VMs on a PM may increase the total power consumption and traffic cost instead of reduce them. If two VMs running network-aware applications, which have low CPU usage, are hosted on a PM, the traffic cost for that connection is decreased. However, due to the low CPU usage, the PM utilization is low. Consequently, more PMs are turned on, thus increase the whole system traffic cost and energy consumption.

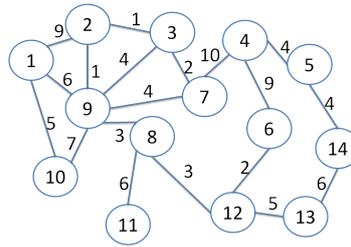


Figure 3: Original graph

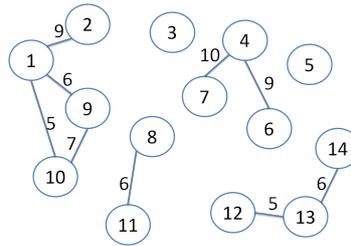


Figure 4: Isolated sub-graphs

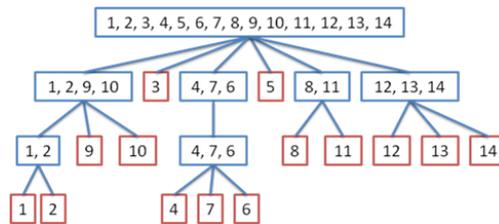


Figure 5: Original graph converted to tree structure

The best option is to host heavy communicated network-aware VMs on PMs that located close to each other. On each PM, network-aware VMs are mixed with CPU-aware and CPU-network-balance ones to maximize its utilization.

We model traffic among VMs as a graph where VMs are vertices and network communications are edges. All vertices are connected to the others. Edge weights are the traffic weight between the VMs. We recursively separate the original traffic map into communities until each community has only one node. According to Porter [16], a community is a group of vertices that connected more densely to each other than the other ones in the graph. All community-detection methods use various techniques to identify these set of nodes. Finally the original traffic map is converted into a tree structure which sibling nodes at low level are heavy communicated. These sibling nodes are then hosted on PMs that located close together as previous analysis.

In order to identify communities, we use the same technique as Dias [9]. Because every vertex is connected directly to the others, the link weight represents how much the two VMs are correlated. Links are sorted in low weight order and lowest weight links are removed from the graph. The original graph is separated into a set of isolated sub-graphs. The process is continued until each graph contains only one vertex. Figures 3, 4 and 5 illustrate the separation process.

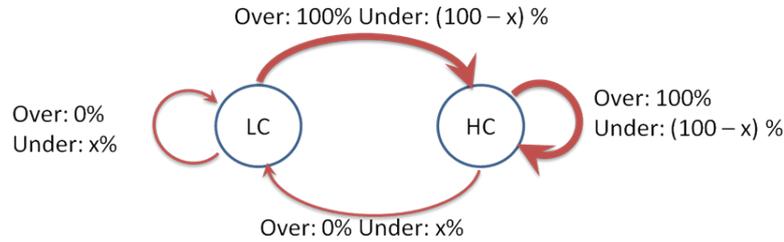


Figure 6: VM flows in heterogeneous environment

#### 4.2 Energy aware finding host for VM

In [2], for each VM to be migrated, whether from an over or under-utilized host, the destination PM is chosen so that the increased energy is minimum. After long term operation, VMs are consolidated on high CPU usage per energy consumption PMs. Our algorithm improves that approach to reduce network congestion while keeping energy consumption at an acceptable value.

In our approach, different policies are applied for VMs migrated from over and under-utilized hosts. Specifically, for VMs migrated from over-utilized hosts, we still use the same method as [2]. On the other hand, for ones migrated from under-utilized hosts, traffic aware placement policy is applied. With this approach, VMs still consolidate on high CPU usage per energy consumption PMs, while traffic cost is reduced.

Figure 6 illustrates the consolidation process. Let HC and LC stand for high and low capacity server sets in a heterogeneous environment;  $x$  is the ratio between the number of running PMs in each set

$$x = \frac{\# \text{running host in LC}}{\# \text{running host in HC}} * 100\%$$

and the arrows stand for VM flows between these two sets. In the figure, "over/under:  $x\%$ " means  $x\%$  of VMs from over/under utilized hosts follow the flow. When a PM in HC set is over-utilized, VMs is flowed to another PM in the set, while if a PM in LC set is over-utilized, VMs is migrated to another set. Consequently, PMs in LC set are not high-utilized and more easily to be turned off compared to ones in HC set. The VM flows become unequal in which the number of VMs migrated from LC set to HC set is bigger than the reverse way. After a long period,  $x \rightarrow 0$  and VMs are consolidated on PMs in HC set.

#### 4.3 Finding host for VM algorithm

Our algorithm named Traffic and Power aware Virtual Machine Placement (TPVMP) which combines the energy and traffic aware techniques is illustrated in Algorithm 1.

<b>Algorithm 1: TPVMP Algorithm</b>
<b>INPUT:</b> virtual machine $v$ , Host list $H$
<b>OUTPUT:</b> allocated host allocatedHost

```
1.  allocatedHost = null
2.  minPower = Double.MAX_VALUE, minDistance = Integer.MAX_VALUE
3.  totalDistance = 0
4.  excludedHost = new HashSet(Host)
5.  currentHost = v.getHost()
6.  if (currentHost != null)
7.    excludedHost.add(v.getHost)
8.  for each host h in H
9.    if (excludedHost.contains(h)) then continue;
10.   if (h.isSuitableForVm(v))
11.     if (h is over-utilized after allocation) then continue;
12.     powerAfterAllocation = getPowerAfterAllocation(h,v)
13.     if (powerAfterAllocation != -1)
14.       if (v is migrated from an over-utilized host OR v has no sibling node)
15.         powerDiff = powerAfterAllocation - h.getPower()
16.         if (minPower > powerDiff)
17.           minPower = powerDiff
18.           allocatedHost = h
19.       if (v is migrated from an under-utilized host)
20.         siblingVms = findSiblingLeaves(v)
21.         for each VM siblingVm in siblingVms
22.           siblingHost = siblingVm.getHost()
23.           totalDistance += distance(h, siblingHost)
24.         if (totalDistance < minDistance)
25.           minDistance = totalDistance
26.           allocatedHost = host
27.  return allocatedHost
```

## 5 Evaluation

We evaluate the efficacy of the traffic and power aware virtual machine placement in a simulation environment using Cloudsim toolkit [7]. 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 [23]. The VM and PM configurations are as same as [2], plus that all PMs in a partition have the same configuration.

We use FNSS [18] 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 [9]. These static volumes then added a zero-mean normal fluctuation value. According to [14], the relation between the standard deviation of this fluctuation ( $\sigma'$ ) and the mean traffic volumes is

$$\overline{x_{ij}(t)} = \psi \sigma'_{ij}{}^\gamma$$

We chose  $\gamma = 0.8$  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, 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 [2], 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 Energy-only. In the simulation, the experiment results when TPVMP, Traffic-only and Energy-only algorithms are applied are compared (Figures 7, 8, 9 and 10). 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 7, TPVMP saves about 10% of traffic cost compared to Energy-only algorithm. Moreover, TPVMP also save about 30% 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 TPVMP and Energy-only are nearly the same. The Traffic-only algorithm saves about 33% of traffic cost, but double the value of SLA violation, 20% number of migrations and 10% energy consumption (Figures 8, 9 and 10)

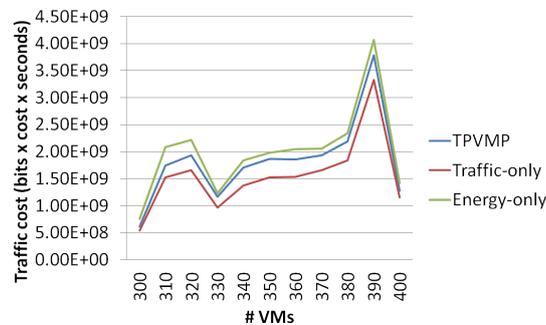


Figure 7: Traffic cost

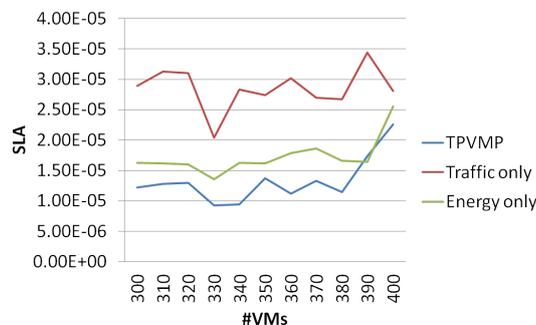


Figure 8: SLA

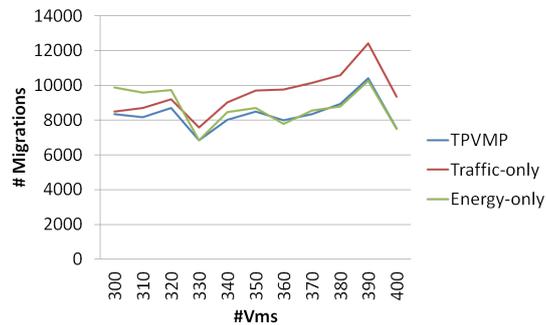


Figure 9: # Migrations

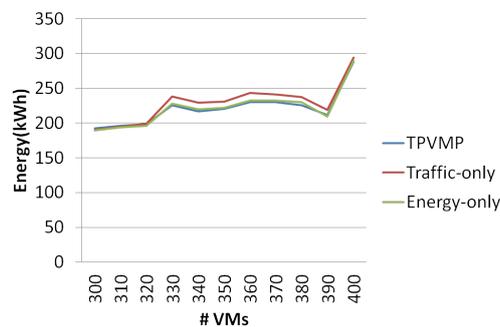


Figure 10: Energy consumption

## 6 Conclusion

This paper presents a VM placement algorithm in heterogeneous cloud data centers that minimize network congestion while energy consumption is unchanged. Different policies are applied for VMs migrated from over and under-utilized PMs. At the end of the simulation, VMs are consolidated on PMs with high CPU usage per energy consumption and heavy communicated VMs are hosted by PMs that located close together. The result shows that our algorithm produce better balance result considering VM communication cost, SLA violation and energy consumption.

## References

- [1] *Cisco Data Center Infrastructure 2.5 Design Guide*. May 2008.
- [2] Anton Beloglazov and Rajkumar Buyya. 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):1397–1420, September 2012.
- [3] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, and Albert Y. Zomaya. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *CoRR*, abs/1007.0066, 2010.
- [4] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang. Understanding data center traffic characteristics. *SIGCOMM Comput. Commun. Rev.*, 40(1):92–99, January 2010.
- [5] R. Bianchini and R. Rajamony. Power and energy management for server systems. *Computer*, 37(11):68–76, 2004.
- [6] Rajkumar Buyya, Anton Beloglazov, and Jemal H. Abawajy. Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. *CoRR*, abs/1006.0308, 2010.
- [7] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exper.*, 41(1):23–50, January 2011.

- [8] Jun Chen, Weidong Liu, and Jiaying Song. Network performance-aware virtual machine migration in data centers. *CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDS, and Virtualization*, pages 65–78, 2012.
- [9] D.S. Dias and L.H.M.K. Costa. Online traffic-aware virtual machine placement in data center networks. In *Global Information Infrastructure and Networking Symposium (GIIS), 2012*, pages 1–8, 2012.
- [10] Truong Vinh Truong Duy, Y. Sato, and Y. Inoguchi. Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. In *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1–8, 2010.
- [11] Irfan Gul and M. Hussain. Distributed cloud intrusion detection model. *International Journal of Advanced Science and Technology*, 34:71–82, September 2012.
- [12] James M. Kaplan, William Forrest, and Noah Kindle. Revolutionizing Data Center Energy Efficiency. Technical report, McKinsey & Company, July 2008.
- [13] Xiaoqiao Meng, Vasileios Pappas, and Li Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, 2010.
- [14] Ismael Solis Moreno and Jie Xu. Energy-efficiency in cloud computing environments: Towards energy savings without performance degradation. *IJCAC*, 1(1):17–33, 2011.
- [15] Antonio Nucci, Ashwin Sridharan, and Nina Taft. The problem of synthetically generating ip traffic matrices: initial recommendations. *SIGCOMM Comput. Commun. Rev.*, 35(3):19–32, July 2005.
- [16] Balaaji Palanisamy, Aameek Singh, Ling Liu, and Bhushan Jain. Purlieus: locality-aware resource allocation for mapreduce in a cloud. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11*, pages 58:1–58:11, New York, NY, USA, 2011. ACM.
- [17] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082–1097, February 2009.
- [18] Atiq Rehman and M. Hussain. Efficient cloud data confidentiality for daas. *International Journal of Advanced Science and Technology*, 35:1–10, October 2011.
- [19] Lorenzo Saino, Cosmin Cocora, and George Pavlou. A toolchain for simplifying network simulation setup. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, SIMUTOOLS '13, ICST, Brussels, Belgium, Belgium, 2013. ICST*.
- [20] Ahmad Shakeel, Ahmad Bashir, Saqib Sheikh Muhammad, and Khattak Rashid Muhammad. Trust model: Cloud's provider and cloud's user. *International Journal of Advanced Science and Technology*, 44:69–80, July 2012.
- [21] Anjana Shankar and Umesh Bellur. Virtual machine placement in computing cloud. Technical report, Indian Institute of Technology Bombay, 2010.
- [22] V. Shrivastava, P. Zerfos, Kang-Won Lee, H. Jamjoom, Yew-Huey Liu, and S. Banerjee. Application-aware virtual machine migration in data centers. In *INFOCOM, 2011 Proceedings IEEE*, pages 66–70, 2011.
- [23] A. Singh, M. Korupolu, and D. Mohapatra. Server-storage virtualization: Integration and load balancing in data centers. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*, pages 1–12, 2008.
- [24] Augustin Soule, Antonio Nucci, Rene Cruz, Emilio Leonardi, and Nina Taft. 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*, pages 73–84, New York, NY, USA, 2004. ACM.
- [25] Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif. Black-box and gray-box strategies for virtual machine migration. In *Proceedings of the 4th USENIX conference on Networked systems design & implementation, NSDI'07*, pages 17–17, Berkeley, CA, USA, 2007. USENIX Association.
- [26] Yueping Zhang, Ao-Jan Su, and Guofei Jiang. Evaluating the impact of data center network architectures on application performance in virtualized environments. In *18th International Workshop on Quality of Service, IWQoS 2010, Beijing, China, 16-18 June 2010*, pages 1–5. IEEE, 2010.

