Energy-aware migration of groups of virtual machines in distributed data centers

Rodrigo A. C. da Silva, Nelson L. S. da Fonseca
Institute of Computing
State University of Campinas, Brazil
Email: rodrigo@lrc.ic.unicamp.br, nfonseca@ic.unicamp.br

***Abstract*—This paper proposes the Topology-aware Virtual Machine Selection (TAVMS) algorithm to choose sets of communicating groups of virtual machines (VMs) to be migrated to other data centers, aiming at global energy savings. It considers the migration of groups of VMs as well as the data center network topology, selecting VM groups with network proximity in order to increase the potential number of equipments to be switched off. Results obtained show that relevant energy savings can be achieved by using the proposed algorithm in the allocation of servers to the migration of virtual machines in a distributed data center scenario.**

I. Introduction

Cloud computing relies on numerous data centers containing thousands of servers to provide services to cloud customers. The main elements of data centers are servers for processing, storage devices for storing customers' data and network switches connecting servers as well as storage devices. The way switches and servers are connected defines the data center network topology. The most popular topologies are the Fat Tree [1] and the one defined by Cisco [2]. Moreover, some providers have more than one data center connected via long-distance links in different regions.

In the pay-per-use Infrastructure-as-a-Service, users request virtual machines (VMs) which are allocated on physical servers designated by a VM placement algorithm. Placement algorithms usually try to consolidate the workload on few servers. In line with that, placing communicating virtual machines in nearby servers helps to reduce switch usage, and, consequently, energy consumption.

Another approach to consolidate the workload is to migrate virtual machines between different data centers to take advantage of underutilized servers and switches in the destination data center, switching off physical resources in the original data center, and promoting overall energy savings.

Data centers consume large amounts of energy and such consumption is expected only to increase in the near future [3]. Studies to improve energy-aware operation in data centers [4]–[6] have aimed at moving VMs between data centers to reduce overall energy consumption. The migration of VMs allows switching off servers and network switches in the source data center and accommodating this (VM) workload in underused servers in the destination data center.

In this paper, we consider several data centers located in different cities and connected to a backbone network. All data centers belong to the same cloud provider and receive requests to establish groups of VMs. A characterization of the resources allocated to the workload is periodically performed at each data center for choosing sets of VM groups for potential migration to another data center. A set of VMs is then migrated only if the migration results in energy savings.

We introduce the Topology-aware Virtual Machine Selection (TAVMS) algorithm to improve overall energy efficiency in distributed data centers. The goal of the algorithm is to choose a set of virtual machines to be migrated to other data centers. It considers the data center network topology in migration decisions and aim at reducing the overall energy consumption. The TAVMS algorithm differs from previous proposals by the consideration of hierarchical network topologies, the most common topologies, as well as the consideration of sets of virtual machines rather than individual VMs. The output of the proposed algorithm is used for the negotiation of potential migration of workload between data centers.

Although several papers [7]–[10] focused on geographical load balancing, only a few considered workload migration [4]–[6]. When migration is considered, authors take into account the cost of electricity and the availability of renewable energy, but model the workload migrated as the entire load of a data center. In addition, the relevance of network topology on energy consumption has been neglected when deciding on the set of VMs to be migrated. In the present work, we make two distinguished contributions. We consider workload migration by choosing groups of VMs rather than the entire workload of a data center. Moreover, we analyze the effects of the data center network topology on energy consumption, when choosing the virtual machines to be migrated.

The remainder of this paper is organized as follows. Section II introduces the proposed algorithm. Section III presents the performance evaluation of the proposed solution. Finally, Section IV draws conclusions and indicates future work.

II. Proposed solution

A. System model

We consider a scenario with data centers located in different cities and managed by a single cloud provider. Data centers are connected via a backbone optical network, not owned by the cloud provider. Requests are modeled as groups of virtual machines, having network flows between the VMs and also to the Internet [11]. VMs must be mapped on the available servers and flows on network links. Requests for

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

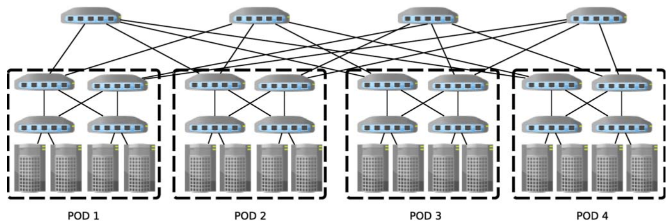


Fig. 1: Data center employing Fat-tree topology with $K = 4$ and PODs highlighted.

allocation of groups of virtual machines arrive at each data center and these VMs are placed on physical servers following the recommendation of a VM placement algorithm.

Each data center periodically evaluates how the total workload can be splitted, identifying sets of VM groups for potential migration. The energy consumption of these sets of VM groups in other data centers is estimated and this workload is migrated only if it reduces the energy consumption.

In this paper, the TAVMS algorithm considers only the Fat-tree topology, though other hierarchical topologies can be used. The Fat-tree topology [1] has three levels and use only commodity switches; it can be seen as structured as series of PODs, which are groups of connected servers by switches. In this topology, there is a parameter K defining its size. A POD has K servers connected to $\frac{K}{2}$ edge switches, which are then connected to $\frac{K}{2}$ aggregation switches. There are K PODs linked to K core switches in the data center network. Figure 1 illustrates a Fat-tree data center with $K = 4$.

B. Algorithm

The TAVMS algorithm attempts to migrate VMs in nearby areas of the source data centers, allowing servers close to each other to be switched off as well as the switches connecting these servers. These VMs can be migrated to underloaded areas in the destination data center. Migration decisions involve two steps: the selection of potential sets of VMs in a data center to be migrated, achieved by performing the Selection (SEL) algorithm, and the negotiation of migration of these sets with other data centers, achieved by performing the Negociation (NEG) algorithm. The SEL algorithm considers a hierarchical data center topology and VMs located in contiguous area of the data center. In this step, different sets of VM groups are identified. The NEG algorithm evaluates the energy impact of each of the selected sets of VMs in each potential data center in order to find potential migration leading to the largest energy savings. The SEL algorithm is executed at the data center from which sets of VM groups will be potentially migrated while the NEG algorithm should run in each data center that can potentially host a set of migrated groups of VMs. If there is no set of VMs which migration would reduce the energy consumption, no workload is selected for migration. In addition, the NEG algorithm analyses migration downtime and the residual lifetime of VMs to prevent unnecessary migrations. Sets of VMs with migration time greater than their remaining lifetime are not migrated.

TABLE I: Notation used for the TAVMS algorithm.

Notation	Description
$\mathcal{D}$	data center
$getPODs(\mathcal{D})$	function that obtains all the PODs from the Fat-tree topology of data center $\mathcal{D}$
$getVMs(\mathcal{A})$	function that returns all the VM groups deployed in area $\mathcal{A}$ of the data center
$energyImpact$ $(groups,$ $destDC,$ $fromDC)$	function that returns the difference between VMs energy consumption in current data center $fromDC$ and destination data center $destDC$, calculated using the destination data center placement algorithm to estimate the power consumption
$migrationTime$ $(groups,$ $fromDC,$ $toDC)$	function that calculates the time to transfer the VM groups in set $groups$ from data center $fromDC$ to $toDC$, based on the virtual machine memory and disk sizes and the backbone network
$firstFinishTime$ $(groups)$	function that returns the remaining time for the first VM group to be released in $groups$.

The SEL algorithm attempts to cluster groups of virtual machines by taking advantage of the network topology. The Fat-tree network contains K PODs, as shown in Figure 1 for $K = 4$, and these PODs can be seen as contiguous areas in the network, such as POD 1 to 4 in Figure 1. Since resource provisioning algorithms, such as [11], tend to place VMs in the same POD or nearby PODs, the SEL algorithm considers sets of VMs in sequential PODs, under the restriction that all VMs in a group should be migrated if the group is migrated.

The output of the SEL algorithm is used as input to the NEG algorithm. The NEG algorithm computes the energy that the set of groups of VMs would consume if migrated to a data center. The data center in which the set of groups of VMs leads to the greater reduction in energy is then chosen to host the set in case the consumption is lower than in the original data center. Furthermore, the NEG algorithm performs an analysis on the completion time of the groups of virtual machines. VMs finishing before the migration time are not considered to be sent to another data center.

The SEL algorithm is presented in Algorithm 1. Its input is the data center network topology and the parameter K of a Fat-tree. In Line 2, inner PODs are identified. At each step, the size of a sequential group of PODs is enlarged considering different levels of the Fat-tree (1 to K, Line 3). For each size, sets of VM groups are identified (Line 5).

Input: $\mathcal{D}$, K
Output: The sets of virtual machines
1 $sets \leftarrow \emptyset$;
2 $pods \leftarrow getPODs(\mathcal{D})$
3 **for** $size \leftarrow 0$ **to** $K - 1$ **do** ;
4   **for** $i \leftarrow 1$ **to** $size$ **do** ;
5         $sets \leftarrow sets \cup getVMs(pods[i, \ldots, i + size])$
6 **return** $sets$

Algorithm 1: SEL (Selection of virtual machines)

The NEG algorithm is described in Algorithm 2. Its input consists of the set of VM groups ($\mathcal{S}$, selected by algorithm SEL) and the source data center. The migration decision is made by computing the energy consumption considering all combinations of sets of VM groups (Line 1) and potential destination data centers (Line 2). VMs terminating before this migration time are not selected to be migrated, since it is not worth migrating them (Lines 9 to 11). Sets of groups of VMs are migrated to the data center that gives the largest energy savings (Line 12). If there is no data center leading to energy reduction, no workload is transferred (Line 13).

Input: $\mathcal{S}, localDatacenter$
Output: The chosen data center
1 $\forall set \in \mathcal{S}$
2 $\quad \forall datacenter$
3 $\quad\quad savings \leftarrow energyImpact(set, datacenter, localDatacenter)$
4 $\quad\quad$ **if** *savings is maximum* **then**
5 $\quad\quad\quad chosenDatacenter \leftarrow datacenter$
6 $\quad\quad\quad chosenSet \leftarrow set$
7 **if** *migration with energy gains was identified* **then**
8 $\quad downTime \leftarrow migrationTime(choosenSet, localDatacenter, choosenDatacenter)$
9 $\quad \forall group \in choosenSet$
10 $\quad\quad$ **if** $remainingTime \leq downTime$ **then**
11 $\quad\quad\quad chosenSet \leftarrow choosenSet \setminus \{group\}$
12 $\quad$ **return** $chosenDatacenter, chosenSet$
13 **return** *No migration*

Algorithm 2: NEG (Negotiation of VM workloads)

III. Performance evaluation

The performance and energy consumption of the TAVMS algorithm were compared to those in scenarios in which no migration is performed as well as those in scenarios in which migration decisions are made based on pre-defined ammount of workload. The policies Topology-aware threshold (TT) and Random Threshold (RT) choose a fixed proportion of the VMs in a data center to be migrated. The former considers the network topology when choosing VMs for migration, while the latter chooses the workload in a random manner.

In the literature, algorithms for migrating workloads between different data centers do not take into consideration the position of individual VMs [4]–[6]. However, the position of virtual machines in the network affects the energy consumption. Releasing VMs from the same server increases the chances of this server being switched off. Moreover, the switches connecting switched off servers are most likely to be switched off. Since our proposal considers the position of individual VMs, differently than previous work [4]–[6]. A direct comparison between our work and others in the literature would not have a common ground. Therefore, we implemented policies that recommend migration of fixed size portion of data center workload.

The TT and TR policies always choose a fixed fraction of the workload of the data center. In the simulations it was set to 10%. The description of the workload is then sent to all other data centers so that its impact on the energy consumption in each data center is calculated. As in the NEG algorithm, the hosting data center which leads to the largest energy reduction is choosen. The TT policy chooses the workload in nearby servers in the Fat-tree topology (Figure 1), from the left to the right. By choosing VMs in this manner, VMs from nearby areas are selected for migration. The RT policy chooses randomly virtual machines in the network.

TABLE II: Configuration of data center physical servers and VM instances

Instance	MIPS	RAM (Mb)	CPU cores	Disk (Mb)
VM Instance 1	2500	1024	1	500
VM Instance 2	2500	2048	1	500
VM Instance 3	2500	4096	1	500
VM Instance 4	2500	3840	1	4000
VM Instance 5	2500	7680	2	32000
VM Instance 6	2500	15360	4	40000

Configuration	MIPS	RAM (Mb)	CPU cores
Hp ProLiant Dl360 G7 Xeon X5675	3067	65536	6
Hp ProLiant Dl380 G7 Xeon X5675	3067	65536	6

The interval between two sequential executions of a policy must be sufficiently large in order to avoid recurrent large transfers across the backbone network. This interval was set to 8 hours in the simulations.

Our proposal considers the VMs already instantiated in a data centers. A virtual machine placement algorithm must be used to an arriving VM group. We employed the algorithm described in [11], which considers inter-VM traffic. The placement algorithm is also employed to estimate the energy costs in the data center which will potentially host a migrated workload, described in the NEG algorithm previously.

A. Simulation Settings

The Cloudsim Simulator [12] was employed to evaluate the proposed solution. It was extended by including data center network topology, energy consumption model for switches, backbone topology for distributed data centers, and migration between data centers. Results were obtained performing up to 36 different executions for each point in the graphs and using a 95% confidence interval derived by the independent replication method.

Data centers have Fat-tree topology [1], with the only difference being that a switch, named Internet switch, is connected to the core switches. In this topology, the parameter K defines the data center size and varied in the range 10 to 16. The numbers of machines and switches (the Internet switch is ignored) are: for $K = 10$, there are 250 servers and 125 switches; for $K = 12$, 432 servers and 180 switches; for $K = 14$, 686 servers and 245 switches; for $K = 16$, 1024 servers and 320 switches. Smaller values of K were not used since they do not represent real data centers, and higher values of K demand long simulation time.

The data center is composed of two types of servers and it hosts six different instances of VMs (Table II). The types of server and VM instances are random variables uniformly distributed. CPU usage is taken from the data set in [13].

B. Backbone network

The topology of the backbone optical network used in the simulations was the National Science Foundation's Network (NSFNet) topology shown in Figure 2. The capacity of all link is set to 100 Gbps, as suggested in [7]. Data centers are

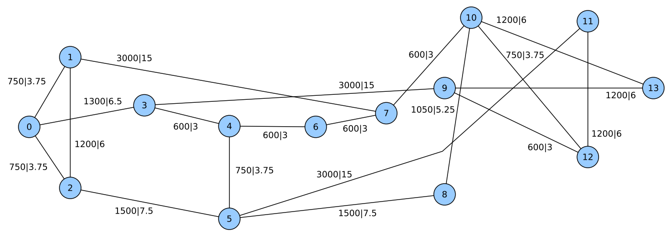


Fig. 2: NSFNet topology with 14 nodes. The link labels is in the format "X|Y": X stands for the link length (kilometers) and Y for the delay (miliseconds).

located in different locations and migration time is calculated by considering the transfer of VMs disk and memory over the links along the shortest path between the two involved data centers.

According to [14], typical values of PUE for modern data centers are between 1.5 and 1.7. In the simulations, four data centers were placed in the NSFNet topology, and their locations and values of PUE are shown in Table III.

TABLE III: Data centers location and power usage effectiveness. Nodes correspond to the vertices in Figure 2

Location	**Node**	**PUE**
Palo Alto, California	0	1.7
Boulder, Colorado	4	1.5
Atlanta, Georgia	8	1.7
Princeton, New Jersey	13	1.5

C. *Energy Consumption Model*

The energy consumption considered in this work has three components: servers, switches and cooling infrastructure. The models used for each component are described as follows.

The model for energy consumption of servers is the one used in [13]. Whenever a server is idle, its consumption is about 70% of the consumption under full load, which grows linearly with the CPU load. When no workload is being processed, a server can be switched to a low consumption mode, thus saving energy. We employed this model with energy consumption values taken from the SPECpower benchmark[1]. Linear interpolation is used to estimate the energy consumption according to the current processing load caused by the VMs load hosted.

The energy model for switches [15] is calculated by considering three components: the switch chassis, line cards and ports. The following formula expresses this model:

$$P_{switch} = P_{chassis} + n_{lc}P_{lc} + \sum_{i=0}^{r} n_{r_i} \cdot P_{r_i}$$

P_{switch} is the total power consumed by a switch. $P_{chassis}$ is the fixed power for maintaining it powered on; P_{lc} is the power consumed by each line card in use and n_{lc} is the number of line cards. Each r_i is a potential transmission rate; n_{r_i} and P_{r_i} are the number of ports transmitting at rate r_i and the power used by a port transmitting at rate r_i, respectively.

[1]http://www.spec.org/power_ssj2008/

TABLE IV: Power consumption of data center equipment

Switches Power Consumption (W)			
Type of switch	$P_{chassis}$	P_{lc}	P_r
Rack, aggregation or core	146	Included in Chassis	0.42
Internet	1558	1212	27

Servers Power Consumption (W)						
	CPU load (%)					
Type of server	**0**	**10**	**20**	**30**	**40**	**50**
Hp ProLiant Dl360 G7 Xeon X567	55.6	95.4	107	115	124	133
Hp ProLiant Dl380 G7 Xeon X567	52.3	93.6	106	116	126	136
Type of server	**60**	**70**	**80**	**90**	**100**	
Hp ProLiant Dl360 G7 Xeon X567	142	155	173	192	216	
Hp ProLiant Dl380 G7 Xeon X567	147	163	180	199	222	

The cooling infrastructure is accounted by using the PUE metric. The total energy spent in servers and switches is multiplied by the correspondent data center PUE value in order to obtain the total energy consumption. Table IV shows the power consumption values used in the simulations for servers and switches, while Table III presents the PUE values used. The network status is periodically checked, and idle servers and switches are powered off.

D. *Traffic Model*

Requests were modeled to arrive at the data center in groups of VMs, as suggested in [7], [11], which is a realistic model of users' requests. A request is composed of a number of VMs that arrive at the same time. These VMs produce network flows, either between pairs of virtual machines in the data center or to the Internet.

According to [16], the VM arrival and departure processes on different time scales exhibit self similarity. We model these processes on a 10-minute time scale and use the generator described in [17] to create self similar series with the parameters described in Table V.

TABLE V: Parameters used for the evaluated scenarios. **M** stands for mean, **SD** for standard deviation and **H** for the Hurst parameter

Parameter	**Model**	
Medium groups (MG)	Self-similar series M: 10; SD: 5; H: 0.7	
Large groups (LG)	Self-similar series M: 20; SD: 10; H: 0.7	
Low-intensive traffic (LT)	Internet flow: Gaussian M: 2 Mbps; SD: 0.2 Mbps	Pair flow: Gaussian M: 5 Mbps; SD: 0.5 Mbps Probability: 0.75
Medium-intensive traffic (MT)	Internet flow: Gaussian M: 4 Mbps SD: 0.4 Mbps	Pair flow: Gaussian M: 10 Mbps; SD: 1 Mbps Probability: 0.75
High-intensive traffic (HT)	Internet flow: Gaussian M: 10 Mbps; SD: 1 Mbps	Pair flow: Gaussian M: 25 Mbps; SD: 5 Mbps Probability: 0.75

Since a group of VMs departs at the same time, we match the generated number of VMs leaving the data center (departure process) with a group of the same size in execution to identify the group of VMs leaving the data center. For the inter VM traffic, we generate the values of transmission rates according to those suggested in [18]. In each group, one VM has a flow directed to the Internet and there is a pre-defined probability that two VMs communicate. If there is a flow between a pair of VMs, a Gaussian distribution is used to generate the transmission rate. Table V summarizes the values used for generating sizes of groups in the arriving/departing series, as well as traffic demands. Six scenarios were generated, identified by S and T: S is the size of the groups (MG or LG) and T is the traffic demand (LT, MT or HT).

E. Numerical Evaluation

The metrics used to evaluate TAVMS performance are the total energy consumption and blocking ratio. The total energy spent i.e., the sum of the energy in all data centers, accounting the PUE metric by multiplying each data center consumption by its PUE value, is shown in Figure 3.

Results show that migration can save energy in geographically distributed data centers. The TT policy reaches up to 6.6% of energy savings when compared to the no migration policy. The energy savings given by the TAVMS algorithm range from 7.1% to 14%, outperforming the other policies in all the scenarios evaluated. Results obtained by TT and by TAVMS policies outperform the other strategies, making evident the importance of addressing the data center network topology when performing migration between data centers.

The blocking ratio is the percentage of VMs not deployed in the data center in relation to the total number of virtual machine requested. Whenever the placement algorithm cannot find servers to place a group of VMs, the request is refused. The overall blocking ratio as a function of the data center size is presented in Figure 4.

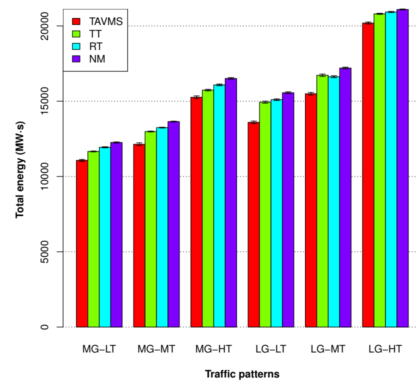
Results given by all algorithms outperform the no migration policy. This improvement shows that migration can save energy as well as enhance virtual machine accessibility, improving service-level agreements. Since the workload is balanced among servers in data centers, the algorithms help to avoid bottlenecks in the data center network. In scenarios with HT traffic, the network congestion occurs due to large network demands, producing higher blocking ratios.

The blocking ratio slightly decreases from $K = 10$ to $K = 12$ since small data centers have a significant limitation on the number of computational resources to cope with incoming requests. From $K = 12$ to $K = 16$, the blocking ratio increases since the number of VMs grows with the data center size, demanding more infrastructure resources.

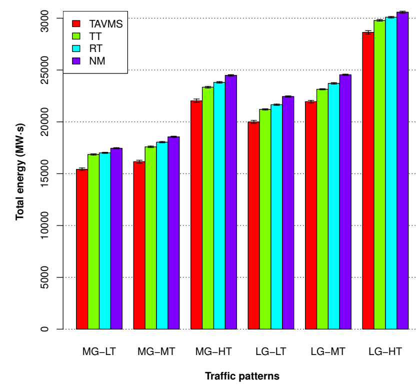
IV. Conclusion

This paper introduced the Topology-aware Virtual Machine Selection algorithm designed to balance the workload between distributed data centers by means of virtual machine migration. The selection of virtual machines considers the data center network topology, aiming at migrating groups of VMs that will allow the switching off of physical servers and switches. Results show that the employment of the TAVMS algorithm can lead to significant energy savings up to 14% in comparison to scenarios without VM migration.

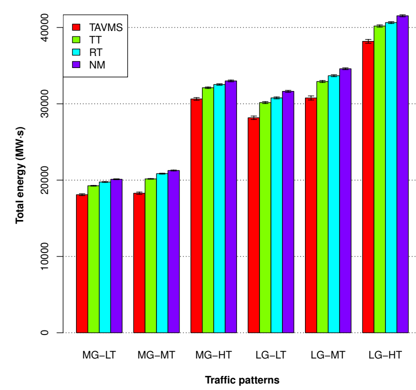
As future work, the TAVMS algorithm should consider electricity prices in different locations at the edge of the



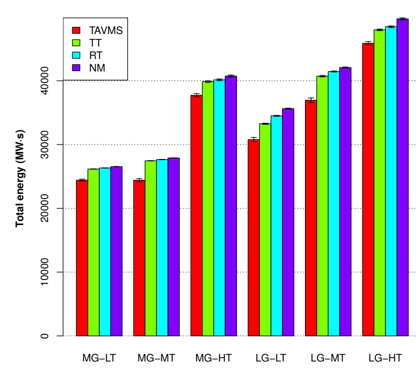
(a) $K = 10$



(b) $K = 12$

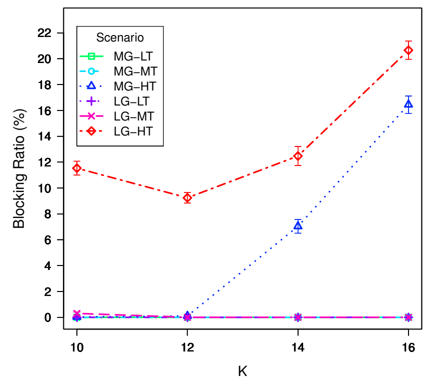


(c) $K = 14$

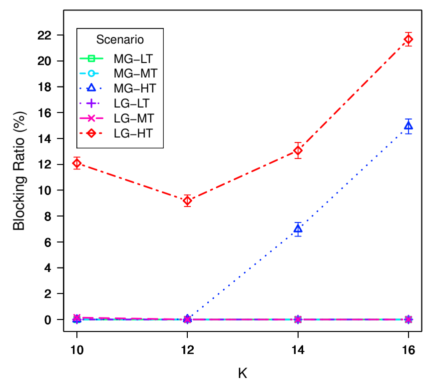


(d) $K = 16$

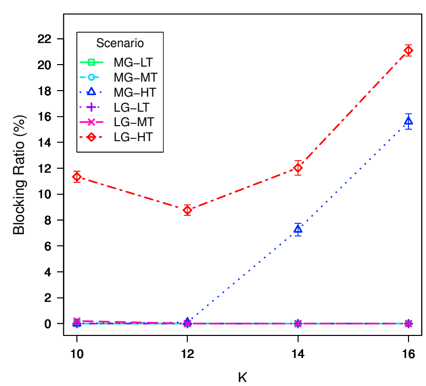
Fig. 3: Total energy for different data center sizes.



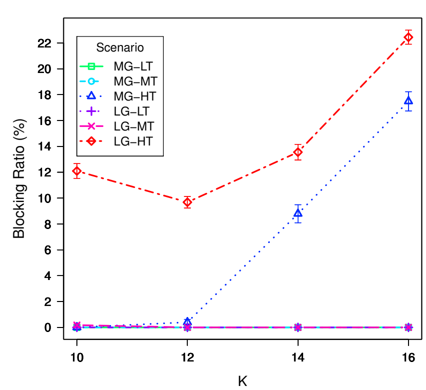
(a) TAVMS



(b) TT



(c) RT



(d) NM

Fig. 4: Blocking ratio produced by different policies as a function of the data center size.

backbone network, as well as different sizes and arrival rates for different data centers.

ACKNOWLEDGMENT

The authors would like to thank the grant 2014/01154-4, São Paulo Research Foundation (FAPESP), and 131821/2013-0, CNPq, for the financial support.

REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[2] *Cisco Data Center Infrastructure 2.5 Design Guide*, Dec. 2007.

[3] J. Koomey, “Growth in data center electricity use 2005 to 2010,” *Oakland, CA: Analytics Press. August*, 2011.

[4] U. Mandal, M. Habib, S. Zhang, B. Mukherjee, and M. Tornatore, “Greening the cloud using renewable-energy-aware service migration,” *Network, IEEE*, vol. 27, no. 6, pp. 36–43, November 2013.

[5] M. Ilyas, S. Raza, C.-C. Chen, Z. Uzmi, and C.-N. Chuah, “Red-bl: Energy solution for loading data centers,” in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 2866–2870.

[6] J. Ruddy and M. O’Malley, “Global shifting of data centers demand,” in *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES*, Oct 2014, pp. 1–5.

[7] A. Amokrane, F. Zhani, Q. Zhang, R. Langar, R. Boutaba, and G. Pujolle, “On satisfying green slas in distributed clouds,” in *IEEE/ACM/IFIP International Conference on Network and Service Management (CNSM)*, Rio de Janeiro, Brazil, November 2014.

[8] Y. Li, H. Wang, J. Dong, J. Li, and S. Cheng, “Operating cost reduction for distributed internet data centers,” in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, May 2013, pp. 589–596.

[9] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew, “Greening geographical load balancing,” *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[10] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, “Data centers power reduction: A two time scale approach for delay tolerant workloads,” in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 1431–1439.

[11] R. A. C. da Silva and N. L. S. da Fonseca, “Topology-aware virtual machine placement in data centers,” *Journal of Grid Computing*, pp. 1–16, 2015.

[12] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[13] 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.

[14] R. Brown *et al.*, “Report to congress on server and data center energy efficiency: Public law 109-431,” *Lawrence Berkeley National Laboratory*, 2008.

[15] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, “Energy aware network operations,” in *INFOCOM Workshops 2009, IEEE*, 2009, pp. 1–6.

[16] Y. Han, J. Chan, and C. Leckie, “Analysing virtual machine usage in cloud computing,” in *Services (SERVICES), 2013 IEEE Ninth World Congress on*, June 2013, pp. 370–377.

[17] V. Paxson, “Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic,” *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 5, pp. 5–18, Oct. 1997.

[18] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, “A stable network-aware vm placement for cloud systems,” in *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, May 2012, pp. 498–506.