An Efficient Energy-Aware Mechanism for Virtual Machine Migration

Leonardo P. Cardoso*, Diogo M. F. Mattos*†, Lyno Henrique G. Ferraz*†,
Otto Carlos M. B. Duarte*, Guy Pujolle†

*Grupo de Teleinformática e Automação - Universidade Federal do Rio de Janeiro (COPPE/UFRJ)
Rio de Janeiro, Brazil - Email:{cardoso,menezes,lyno,otto}@gta.ufrj.br

†Laboratoire d'Informatique de Paris 6 - Sorbonne Universities, UPMC Univ Paris 06
Paris, France - Email: Guy.Pujolle@lip6.fr

***Abstract*—A major concern for cloud infrastructure providers is to decrease both their carbon footprint and their energy consumption. In this paper, we propose an elastic and energy-aware virtual machine migration mechanism that decreases power consumption by turning off physical machines and turns them on according to the demand. We develop heuristics that reallocate virtual machines in a smaller number of physical machines, turns off the idle physical machines and, consequently, saves energy. When the demand increases, the proposal turns on physical machines to supply the required resources. The contributions of the paper are four-fold: i) an efficient and elastic mechanism for energy saving on data centers; ii) a simple formal model for minimization of idle physical machines; iii) an implementation in a real environment, and iv) a simulation of the optimization algorithm for larger scenarios. The mechanism was developed and tested in Future Internet Testbed with Security (FITS). The results show the effectiveness of the mechanism on reallocating virtual machines to decrease power consumption.**

I. Introduction

Cloud Computing allows the infrastructure provider to offer computational resources, such as processing, memory, and bandwidth, in a dynamic and flexible way according to the clients' demand. Besides avoiding operational costs of maintenance, configuration, and repair; the clients can also obtain the required resources at the moment they need, preventing over or under invests on infrastructure. Virtualization techniques expand the flexibility in resources allocation by abstracting hardware, allowing resource sharing among users of the same machine [1]. Virtualization also provides isolation between virtual machines; thus, there is no interference among virtual machines processes. In order to avoid overload, it is common to distribute resources among the physical machines linked in a network or in clusters. The migration of virtual machines is an effective way to dynamically reallocate resources among the existing physical machines. Live migration [2], [3] allows the virtual machines to remain in operation during the migration process, ensuring high service availability.

Resource allocation is a major challenge in cloud computing, since it defines if the providers can meet the Service Level Agreements (SLAs) without compromising their revenues. On the one hand, an efficient resource allocation must meet all clients' quality of service (QoS) requirements. In addition, the meeting of the clients' requirements must ensure the lowest use of physical resources to decrease the number of active computers, thus reducing power consumption. Current virtualization platforms, as Xen [4], do not natively support a management mechanism that allows an efficient resource allocation. Besides, allocating different resources in restricted capacity machines is a Generalized Assignment Problem (GAP), thus NP-Hard [5]. The time to calculate the solution exponentially grows as the number of considered resources and of machines linearly increase. Finding an exact solution does not scale.

In this paper, we propose a simple formal model for reducing energy consumption on data centers. Besides, we propose and implement a simple and automatic management mechanism based on virtual machines migration intra data center to minimize power consumption and to decrease idle resources. The energy consumption and the idle resources minimization is performed based on the monitoring and on the analysis of the resource usage profiles of virtual machines and of physical machines. From the gathered information, a heuristic reallocates resources through virtual machines migration, minimizing the number of active physical machines. Our heuristic is based on the Simulated Annealing meta-heuristic because it is proved to converge to an optimal solution when time tends to infinity. Nevertheless, when we limit the number of iterations of Simulated Annealing, we also achieve an optimized solution. The algorithm provides an optimized mapping of virtual machines on physical machines that searches for a reduced number of active physical machines. The mechanism also considers the total amount of bytes transferred over the network as the cost of a virtual machine migration. Then, the virtual machines migrate one by one to attenuate Service Level Agreements violations, which could be caused by the increase on the network traffic that the migration data transfer requires. After migrations, the physical machines in idle state shut down to reduce idle resources and the energy consumption. The machines are kept shut down until the clients demand becomes higher than the offered resources. In this case, physical machines are activated to meet the demand.

The management mechanism was implemented and tested in the Future Internet Testbed with Security (FITS) [6]. We evaluated our proposal through two different approaches: an experiment and a simulation. First, we perform an experi-

This work was sponsored by CNPq, CAPES, and FAPERJ.

ment to demonstrate the correct and efficient operation of the mechanism. The results show that the mechanism finds lower cost solutions, migrates virtual machines to achieve this solution, turns off the idle machines and, when the requirement for resources increases, turns on machines according to the demand. After that, we performed a simulation of the proposed algorithm on a large set of virtual and physical machines. Besides, we show that the implemented Simulated Annealing algorithm minimizes the number of physical machines and outperforms other greedy heuristics results.

The reminder of the paper follows. In Section II, we present the related works. In Section III, we propose our efficient resource allocation mechanism, describing data collection, the implemented heuristics and the mechanism. The results are discussed in Section IV. Section V concludes the paper.

II. Related Works

Scheduling algorithms for cloud computing environments are still a research challenge. Most of them are proposed for increasing the number of new virtual machines a provider can accept and, thus, increasing provider's revenue [7]. Traditional approaches focus on web servers and server farms [8]. Nevertheless, in this paper, we propose a different approach. Our approach is to reduce the idle resources shutting down the physical machines that are idle on the datacenters.

Tian *et al.* propose a simulation and modeling tool-kit for scheduling virtual machines on a cloud datacenter [8]. Their proposal bases on a predictor of virtual machine workloads and simulates it on a cloud environment. The main idea is to provide a tool-kit to help datacenter administrators on the decision-making process and to improve resources allocation to answer virtual machine demands. Besides, Huang *et al.* propose a job scheduling algorithm which packs job resources into virtual machines [9]. This proposal, however, focuses on reducing job completion time and does not consider the energy-efficiency of the entire datacenter infrastructure.

The development of mechanisms that aim to minimize power consumption is a current issue and a challenge because of the number of variables involved. Several proposals address the allocation of virtual elements without regard to the energy consumption. Nejad *et al.* also propose a resource allocation mechanism for cloud environments [7]. Their proposal is based on an incentive mechanism which is used to benefit the virtual machines that reveal their real resource demand. The main goal of this proposal is to predict the accurate resource demand of all virtual machines, to correctly allocate them, and, thus, accept new virtual machines on the datacenter, increasing providers' revenue. This proposal does not focus on the energy consumption. Dabbagh *et al.*, however, propose an energy-efficient resource allocation framework that predicts the number of virtual machine requests, and, thus it is able to provides estimations of the number of physical machines. The energy-saving mechanism focuses on putting into sleep mode the unneeded physical machines [10]. Although the prediction algorithm is accurate, it demands a calibration for each environment and for each workload. If there is a sudden change on the cloud workload, the prediction algorithm is not able to follow it. On the contrary, our proposed mechanism runs real-time monitoring and analysis on the physical and virtual machines. The real-time monitoring enables instantly reactions against flash-crowds.

Wu *et al.* compare the First Fit Decreasing (FFD) greedy heuristic and Simulated Annealing meta-heuristic to reallocate virtual machines and to provide energy saving [11]. Wu *et al.* conduct simulations varying the number of physical and virtual machines as well as the capacity and the resource usage of each of them. The minimization is performed over an energy consumption function. The energy function relies on parameters such as the energy spent by the physical machine processing, which depends on the hardware. They concluded that the use of Simulated Annealing in conjunction with FFD finds solutions that minimize more energy consumption than just using the FFD or Simulated Annealing. The authors, however, focus on the proposal and algorithm simulation. Thus, the solution lacks a practical development that handles with the related effective management problems of a virtualized environment. The proposal has only assessed the possibility of allocating virtual machines and the time the algorithm takes to achieve a solution, without regard to the time necessary for the system to converge, which may be critical in a real application.

Rodriguez *et al.* implement a branch and cut [12] heuristics based on a linear programming 0-1. The algorithm tries to find a mapping of virtual routers and virtual links in physical routers and physical links. The paper evaluates the trade-off between the minimization of energy consumption by request and the bandwidth to allocate virtual resources. The authors optimize the virtual routers instantiation on the network creation phase. In contrast, in this paper, we discuss the migration of instantiated virtual machines that are in use by the clients. The clients have pre-defined service level agreements with the infrastructure provider, and the proposed solution attenuates violations caused by migration and overloads.

The proposed manager mechanism implements the Simulated Annealing meta-heuristics, an Energy Manager module, integrates and modifies the Resources Monitor from Volume Optimization Layer To AssIgn Cloud resources (VOLTAIC) [13]. VOLTAIC is a resource management system to cloud computing which provides quality of service and avoids the waste of resources. The proposed manager mechanism, as well as VOLTAIC, is based on resource usage profiles to analyze the Physical Machines capacity.

III. The Proposed Mechanism

We propose a management mechanism that is able to decrease the number of active physical machines on a cloud datacenter, using heuristics that consider the costs that affect the migration time, such as virtual machine memory transfer bandwidth and network traffic. The proposed mechanism uses the live migration technique to decrease the downtime of virtual machines while they are migrated. Thus, when a migration occurs, just the virtual machine memory is transferred through the network. This technique allows the virtual machine to remain running during the migration, entering in an idle state for a short period of time. The mechanism also avoids network overload, caused by the migration, as well as memory or processing overload of the physical machines. We address network overload by choosing solutions that migrate virtual machines with less memory to transfer without increasing

the solution cost. The processing and memory overloads are avoided by using the Wake on Lan (WoL) protocol to turn on physical machines on the network whenever the demand is higher than the offer.

The evaluation of the proposed mechanism was held in Future Internet Testbed with Security (FITS), which provides a virtualized environment in which Internet of the Future proposals are tested [6]. FITS is an inter-university test network based on Xen and OpenFlow technologies, counting with partners in Brazil and in Europe. FITS adopts a pluralist approach and allows the execution of distinct operational systems and applications on top of virtual networks.

The proposed mechanism aims to minimize the power consumption and the idle resources. Fan *et al.* state that an idle server does not consume less than 50% of the power it would consume in a peak phase [14]. Therefore, a physical machine must be completely shutdown to achieve a substantially decrease of power consumption. Thus, the key idea is to migrate the virtual machines to a reduced number of physical machines to be able to shutdown the physical machines that do not host any active virtual machines. The proposed and developed mechanism has four main modules: the Resource Monitor, the Optimizer, the Migration Orchestrator and the Energy Manager, which are detailed in Section III-B. These modules generate usage profiles, calculate an energy-saving state, migrate virtual machines, and turn on or shutdown physical machines.

The memory amount of each migrated virtual machine influences on the system convergence time because it is necessary to transfer the memory bytes of the virtual machine from the source to the destination physical machine. Thus, whenever a solution with the same cost as the stored is achieved, the implemented Simulated Annealing chooses the solution with the lowest amount of memory to transfer, decreasing the migration time. Therefore, whenever this condition occurs, the algorithm checks which virtual machines changed position regarding the initial solution and calculates their memory amount. After this calculation, the algorithm chooses and stores the migration plane with the lowest amount of memory to transfer in order that the migration process causes the network overload as low as possible.

A. The Formal Model

We model the problem of the minimum number of active physical machines as follows. The parameters are:

- V - set of instantiated virtual machines.
- F - set of active physical machines.
- R - processing, memory or network resource.
- R_v, $v \in V$ - resource used by the virtual machine v.
- R_f, $f \in F$ - resource used by the Domain-0 of the physical machine f.
- $T_{R|f}$, $f \in F$ - resource threshold of the physical machine f.
- $X(f,v) = \begin{cases} 1, & \text{if } v \in V \text{ is instantiated in } f \in F \\ 0, & \text{otherwise} \end{cases}$
- $X(f) = \begin{cases} 1, & \text{if } \sum_{v \in V} X(f,v) \geq 1 \\ 0, & \text{otherwise} \end{cases}$

The minimization model follows.

Minimize:

$$\sum_{f \in F} X(f) \tag{1}$$

subject to:

$$\forall r \in R, \forall f \in F, (\sum_{v \in V} r_v * X(f,v)) + r_f \leq T_{R|f} \tag{R1}$$

$$\forall v \in V, \sum_{f \in F} X(f,v) = 1 \tag{R2}$$

$$\{V, F\} \subset \mathbb{Z}_+^* \tag{R3}$$

The membership function $X(f,v)$ verifies if the physical machine $f \in F$ contains the virtual machine $v \in V$. In addition, the membership function $X(f)$ verifies if the physical machine $f \in F$ contains any virtual machines. The Expression 1 is the problem objective function, which is the minimization of the summation of the active physical machines $X(f)$. Thus, it is the number of active physical machines.

The Restriction R1 ensures that the resources usage does not surpass a threshold. Thus, any resource usage for any active physical machine $f \in F$, is necessarily lower or equal to the summation of the resource used by the virtual machines $v \in V$ belonging to the machine f $(X(f,v))$, and by the resource usage of the machine f Domain-0[1]. The Restriction R2 enforces that a virtual machine v belongs to just one physical machine f. Therefore, the summation of the physical machines f, which contain the virtual machine v, must be 1. This restriction prevents that the same virtual machine belonging to the set of instantiated virtual machines V be instantiated in two or more physical machines at the same time, as well as it ensures that a virtual machine v has just one physical machine f as destination. The Restriction R3 limits the set V and the set F to the domain of the integers higher than zero.

B. The Modules of Proposed Mechanism

The mechanism has four main modules, as shown in Figure 1. The Resource Monitor collects the physical and virtual machines resource usage. The Optimizer executes the heuristics to minimize the number of active physical machines. The Migration Orchestrator redistributes the virtual machines in the physical machines. The Energy Manager shutdowns and turns on physical machines according to the demand.

The Resource Monitor collects CPU, memory, and bandwidth consumption from physical and virtual machines. The `libvirt` library performs the information gathering through the communication with each physical machine hypervisor. The CPU consumption of each virtual machine is directly obtained by `libvirt`. The physical machine CPU consumption is calculated from the summation of its virtual machines CPU consumption. The memory usage profile is generated from the amount of memory allocated to the physical and to the virtual machines. The network usage of the virtual machines is collected from the data traffic in each virtual network interface.

[1]Administrative domain is referenced as Domain-0.

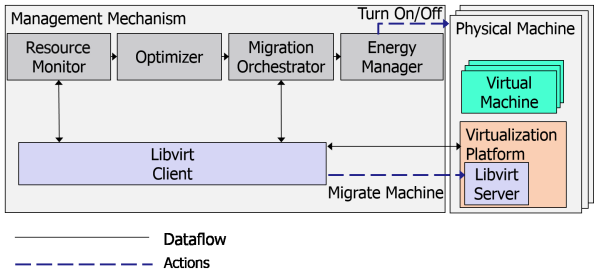


Figure 1. Architecture of the proposed elastic Management Mechanism. The mechanism analyses the resource usage using Libvirt and automatically migrates virtual machines. The idle physical machines are turned off. Otherwise, when the resouce demand increases, new physical machines are turned on through Wake On Lan.

As the mechanism does not consider the network topology, the network usage of the physical machines contributes just for the Domain-0 processing.

The Optimizer obtains the usage profiles from the Resource Monitor and executes a Simulated Annealing meta-heuristics[2] to minimize the amount of active physical machines. After the execution, the Optimizer generates a new distribution of virtual machines over the physical machines. The information of which virtual machines will be migrated between which physical machines is sent to the Migration Orchestrator.

The Migration Orchestrator manages the migration of the virtual machines. The Orchestrator uses Xen live migration. The migration time depends on the memory size to transfer, the memory-data update rate and the physical machine resource usage. In order to mitigate network overload due to migration, virtual machines are migrated one by one.

The Energy Manager shuts down the physical machines that, after the migration, have no more virtual machines instantiated. This module also turns on the physical machines when it is no longer possible to meet all customers' requirements due to an increasing demand of resources. This increase may be observed when the resources consumption of a physical machine reaches a certain threshold in a given number of followed monitoring intervals, which characterizes an overload. If the optimization algorithm does not find a lower cost solution, another physical machine is turned on through the `Wake On LAN` and the virtual machines are redistributed after a new round of algorithm execution.

As the multiplatform `libvirt` library gathers usage profiles and migrates virtual machines, the management mechanism can be used with virtualization platforms such as Xen or KVM. In addition, it does not require any modification on the physical machines, or the installation of any additional software. Thereby, it preserves the virtual machines isolation. The mechanism execution schedule depends on the administrator's policies and can be kept in continuous execution or it may be executed just as answer to an event. In such a case, the time the algorithm takes to find a solution must be taken into account, which varies with the quantity of physical and virtual machines. Whenever a migration takes places, the algorithm enters in a lock state to avoid calculating inconsistent solutions.

[2]The Simulated Annealing (SA) meta-heuristics uses the proposed model of the resource allocation problem as an objective function. SA minimizes the Expression 1.

IV. Evaluation and Results

We evaluated the proposed mechanism through two different experiments: an implementation in a real environment; and a simulation for evaluating scalability. For the first experiment three FITS physical machines were monitored, Leblon, Pao de Acucar and Itanhanga. Leblon and Pao de Acucar are equipped with a CPU Intel i7 3.2 GHz and 16 GB of RAM. Itanhanga is equipped with CPU Intel i7 3.1 GHz and 8 GB of RAM. The machines run Debian Wheezy as operational system and execute Xen version 4.1.3. The images of the virtual machines are in a FITS central node, and then it is not necessary to copy the disk through the network. The proposed management mechanism runs in a machine with CPU Intel Core 2 Quad and 3 GB of RAM. This machine runs externally of FITS to not interfere on the measures. The Domain-0 consumes 2 GB of RAM. The virtual machines present heterogeneous configurations of memory and processing to evaluate the effectiveness of the proposed mechanism. The virtual machine `lpcvm1` was configured with 2 virtual CPUs and 2 GB of RAM. The virtual machines `lpcvm2` and `lpcvm3` were configured with 1 virtual CPU, 4 GB and 3 GB of RAM, respectively. The processing of the virtual machines was generated with the Stress[3] program that generates controlled processing workloads.

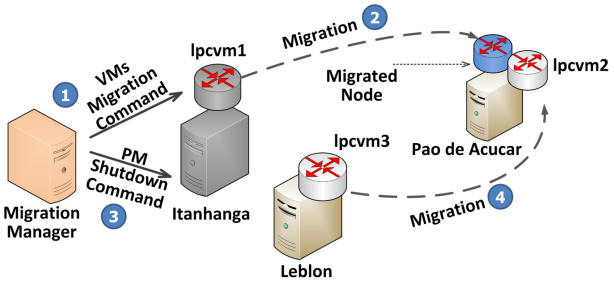
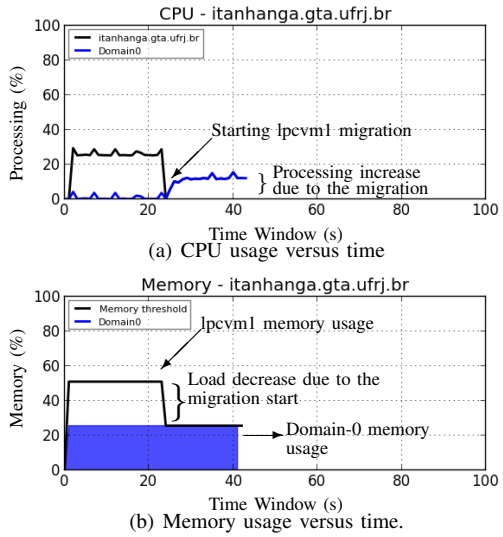


Figure 2. Virtual machine migration experiment: 1) The Migration Manager calculates the solution and sends migration command, 2) migrates the lpcvm1 virtual machines from Itanhanga physical machines to Pao de Acucar physical machine, and 3) shutdowns Itanhanga and Leblon. 4) After, the Migration Manager acts on Leblon physical machine and migrates lpcvm3.

The topology of the experiment is shown in Figure 2. The Migration Manager Machine calculates the solution, migrates the `lpcvm1` and `lpcvm3` virtual machines from Itanhanga and Leblon physical machines to Pao de Acucar physical machine, and shutdowns Itanhanga and Leblon. Figures 3 and 4 show the migration of `lpcvm1` from Itanhanga to Pao de Acucar. Figure 5 shows `lpcvm3` instantiated in Leblon while `lpcvm1` is being migrated from Itanhanga to Pao de Acucar. Following, `lpcvm3` is migrated from Leblon to Pao de Acucar, as shown by Figure 6. Finally, Itanhanga and Leblon are shutdown to save energy. This result shows that the mechanism executes the optimization algorithm, performs the necessary migrations and shutdowns the idle physical machines, minimizing the number of active physical machines. Consequently, the mechanism achieves the objective of decreasing energy consumption. Furthermore, the migrated virtual machines are the ones that have the lowest amount of memory, 2 GB and 3 GB, reducing the the network load due to the migrations.

After this experiment, a resource overload scenario was performed. In this experiment a fourth virtual machine is

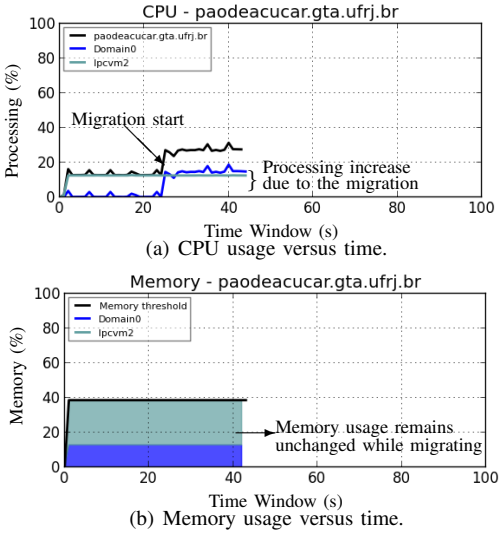
[3]http://people.seas.harvard.edu/~apw/stress/



(a) CPU usage versus time

(b) Memory usage versus time.

Figure 3. Migrating the virtual machines to the lowest cost solution obtained by the Simulated Annealing optimization. Migration of `lpcvm1` from Itanhanga to Pao de Acucar. This virtual machine has the smallest amount of memory.



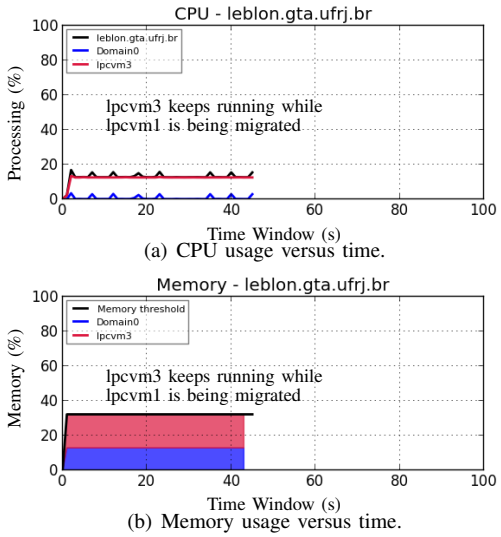
(a) CPU usage versus time.

(b) Memory usage versus time.

Figure 4. Pao de Acucar physical machine state while `lpcvm1` is being migrated from Itanhanga to Pao de Acucar.

instantiated in the physical machine Pao de Acucar. Pao de Acucar reaches the memory threshold and the mechanism detects the overload. As the resource offer is lower than the demand, the Energy Manager turns on a physical machine, in this case, Itanhanga, and the Optimizer calculates a new solution and lpcvm1 is transferred to Itanhanga.

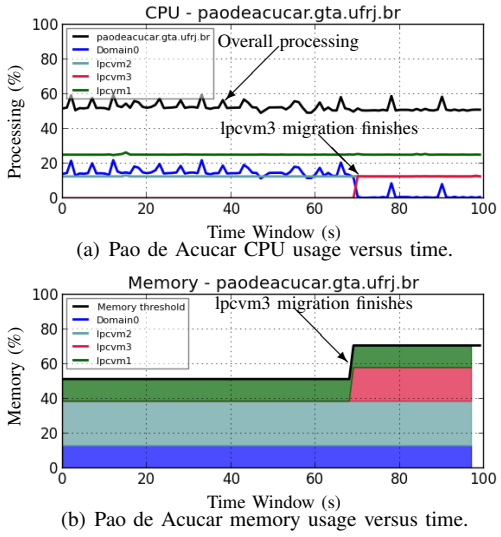
The optimization test compares three virtual machines allocation techniques through simulation. The techniques are the Simulated Annealing meta-heuristics, and the greedy heuristics First Fit (FF) and First Fit Decreasing (FFD). The FF and FFD heuristics are commonly used for solving the bin-packing problem. In this paper, the First Fit heuristics attempts to allocate the virtual machines in the lowest index physical machine with available resources, otherwise it creates a physical machine to allocate the virtual machine. The First Fit Decreasing has the same procedure of the First Fit but sorts the virtual machines in decreasing order of processing, memory



(a) CPU usage versus time.

(b) Memory usage versus time.

Figure 5. The `lpcvm3` virtual machine instantiated in Leblon physical machine while `lpcvm1` is being migrated from Itanhanga to Pao de Acucar.



(a) Pao de Acucar CPU usage versus time.

(b) Pao de Acucar memory usage versus time.

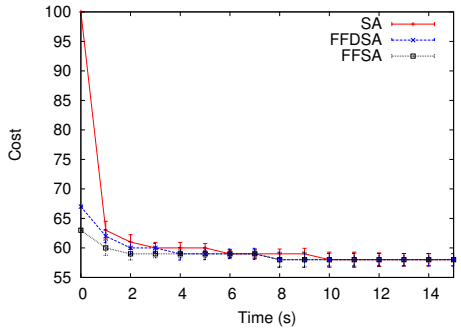
Figure 6. After the `lpcvm1` migration, the lpcvm3 is migrated and the idle physical machines Itanhanga and Leblon are turned off.

and bandwidth consumption before starting to allocate the virtual machines. Initially, 50, 100, 500 and 1000 virtual machines were instantiated in 50, 100, 500 and 1000 physical machines, respectively. In this test, all physical machines have 100% of processing capacity and 16 GB of memory.

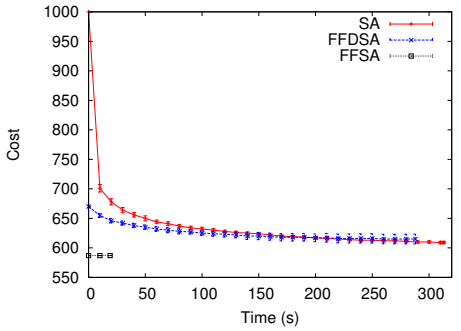
In order to create a heterogeneous virtual machines scenario, each virtual machine resource, such as processing, memory, and bandwidth, was generated following a normal distribution. The memory resources were generated between 0 and 16 GB with an average of 8 GB and standard deviation of 4 GB. The CPU and bandwidth resources were generated between 0 and 100% of the physical machines capacity, with an average and standard deviation of 50%. The tests comprehend 10 rounds limited to 320 seconds and the Simulated Annealing initial temperature[4] is set to 10^6. Those parameter values were

[4]Temperature stands for the parameter of the Simulated Annealing meta-heuristic that limits the search space, allowing the acceptance of higher cost solutions. This parameters decreases when the solution converges to the optimal solution.

obtained through successive algorithm runs and were set to simulate an environment in which the algorithm response time is limited. The algorithm stops when the temperature reaches 0 or the time reaches 320 seconds.



(a) Performance for 100 virtual machines.



(b) Performance for 1000 virtual machines.

Figure 7. Optimization tests for 100 and 1000 physical and virtual machines.

Figure 7 illustrates the average time that the Simulated Annealing takes to find a lower cost solution, in terms of physical machines, or to improve the solutions obtained by the First Fit Decreasing (FFD) and First Fit (FF) greedy heuristics. In the figures, we depict the Simulated Annealing running apart from extra heuristics (SA), the Simulated Annealing improving the First Fit Decreasing solution (FFDSA) and the Simulated Annealing improving the First Fit solution (FFSA). The time that the First Fit and First Fit Decreasing take to find a solution is lower than 1 second and is not shown on the figures. Each point represents the average and standard deviation for 10 rounds of simulation. The data is taken whenever the algorithm finds a lower cost or same cost solution regarding the number of physical machines. Apart from the initial state, the Simulated Annealing algorithm decreased the cost function in all rounds. Thus, the number of active physical machines dropped up to 60% in all configurations, which proves that the our proposed mechanism is able to save energy by shutting down more physical machines than other proposals based on greedy heuristics.

V. Conclusion

Our proposal minimizes the energy consumption by migrating virtual machines and turning off the idle physical machines, decreasing to a minimum the number of active physical machines. The proposal is also elastic and, on the other hand, turns on physical machines when the resources demand is greater than the available resource. The optimization mechanism bases on Simulated Annealing to find lower energy-cost solutions. We implemented the mechanism in the Future Internet Testbed with Security (FITS). The simulations of a larger environment compares our proposal with greedy heuristics. The results show that the mechanism finds lower energy-cost solutions and migrates accordingly the virtual machines, shutting down the idle physical machines. The optimization results show that the Simulated Annealing outperforms the greedy heuristics and is able to decrease the number of active physical machines up to 60%. As a future work we intend to test the mechanism in a global scale and enhance the automatic migration algorithm to support migration inter-datacenters.

References

[1] N. C. Fernandes, M. D. D. Moreira, I. M. Moraes, L. H. G. Ferraz, R. S. Couto, H. E. T. Carvalho, M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "Virtual networks: Isolation, performance, and trends," *Annals of Telecommunications*, pp. 1–17, 2010.

[2] D. M. F. Mattos and O. C. M. B. Duarte, "XenFlow: Seamless migration primitive and quality of service for virtual networks," in *IEEE GLOBECOM 2014*, Dec. 2014, pp. 2326–2331.

[3] D. M. F. Mattos, L. H. G. Ferraz, and O. C. M. B. Duarte, "Virtual machine migration," in *Cloud Services, Networking and Management*, N. L. S. da Fonseca and R. Boutaba, Eds. Hoboken, EUA: Wiley-IEEE Press, Apr. 2015.

[4] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, "Towards high performance virtual routers on commodity hardware," in *Proceedings of the 2008 CoNEXT Conference*. ACM, 2008, pp. 1–12.

[5] N. Jain, I. Menache, J. Naor, and F. Shepherd, "Topology-aware VM migration in bandwidth oversubscribed datacenter networks," in *Automata, Languages, and Programming*. Berlin, Germany: Springer, 2012, vol. 7392, pp. 586–597.

[6] I. M. Moraes, D. M. F. Mattos, L. H. G. Ferraz, M. E. M. Campista, M. G. Rubinstein, L. H. M. Costa, M. D. de Amorim, P. B. Velloso, O. C. M. Duarte, and G. Pujolle, "FITS: A flexible virtual network testbed architecture," *Computer Networks*, vol. 63, no. 0, pp. 221 – 237, 2014.

[7] M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 594–603, Feb. 2015.

[8] W. Tian, Y. Zhao, M. Xu, Y. Zhong, and X. Sun, "A toolkit for modeling and simulation of real-time virtual machine allocation in a cloud data center," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 153–161, Jan. 2015.

[9] D. Huang, P. Du, C. Zhu, H. Zhang, and X. Liu, "Multi-resource packing for job scheduling in virtual machine based cloud environment," in *IEEE Service-Oriented System Engineering (SOSE)*, Mar. 2015, pp. 216–221.

[10] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Energy-efficient resource allocation and provisioning framework for cloud data centers," *Network and Service Management, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[11] Y. Wu, M. Tang, and W. Fraser, "A simulated annealing algorithm for energy efficient virtual machine placement," in *IEEE SMC*, 2012, pp. 1245–1250.

[12] E. Rodriguez, G. Alkmim, D. Batista, and N. Da Fonseca, "Trade-off between bandwidth and energy consumption minimization in virtual network mapping," in *IEEE LATINCOM*, 2012, pp. 1–6.

[13] H. E. T. Carvalho and O. C. M. B. Duarte, "VOLTAIC: volume optimization layer to assign cloud resources," in *Proceedings of the 3rd ICICS*. ACM, 2012, pp. 3:1–3:7.

[14] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the 34th ISCA*. ACM, 2007, pp. 13–23.