Evaluating the Trade-off Between DVFS Energy-savings and Virtual Networks Performance

Fábio Diniz Rossi, Marcelo da Silva Conterato,
Tiago Ferreto, César A. F. De Rose
Faculty of Informatics
Pontiphical Catholic University of Rio Grande do Sul (PUCRS)
Porto Alegre/RS – Brazil
{fabio.diniz, marcelo.conterato}@acad.pucrs.br, {tiago.ferreto, cesar.derose}@pucrs.br

***Resumo*—Data centers usually employ virtualization techniques coupled with other techniques, such as Dynamic Voltage and Frequency Scaling (DVFS), in order to reduce overall energy consumption. However, changes in processor frequency may impact the network performance, specially in metrics such as throughput and jitter. This paper evaluates the trade-off between changes in processor frequency and network performance. Our results show that there is an opportunity to save energy by up to 15%, through the processor frequency reduction. However, this reduction in frequency may increase the response time of applications by up to 70%, directly influencing the quality of experience (QoE).**

Keywords*—*Benchmarking; DVFS; throughput; virtualization.

I. Introduction

Cloud computing aims at providing scalable and on-demand IT resources (e.g., processing, storage, database) through the Internet. These resources can be accessed from anywhere, anytime, using any sort of computing device, such as desktops, tablets or smartphones. The market movement towards cloud computing and IT services outsourcing favors the business of data centers, but the segment still faces major challenges, particularly regarding capital expenses and power consumption costs.

According to a report from Stanford University [1], power consumption in data centers has increased significantly in the last years. Between 2005 and 2010, energy consumption increased by 56% around the world (36% only in the United States). Beyond economics, energy consumption affects other issues, such as cooling and emission of harmful gases.

To enable energy-savings, new proposals have been presented from green data center designs, using natural air cooling, to the use of special technologies that optimize resources utilization. Virtualization [2], [3], [4] is one of these technologies, which serves as the core infrastructure of current cloud computing environments, and due to its features such as virtual machines migration and server consolidation, enables reduction in energy consumption. In addition, there are also technologies that allow energy-savings in data center servers, putting servers in standby or altering processing performance to adequate workloads demand, and consequently decreasing energy consumption.

In particular, Dynamic Frequency and Voltage Scaling (DVFS) [5], [6] is a technique frequently used to save energy on servers. DVFS is specially interesting in data centers that employ virtualization, where each server hosts a different group of virtual machines with diverse aggregate resources demands. However, several studies, such as [7], [8], show that changes in processor frequency can directly impact on the performance of network-dependent applications. This can be a decisive factor for the utilization of DVFS in data centers that support cloud services, since when the processor frequency is reduced, the processing capacity of the node is compromised, affectting all other components, including the network.

Clearly, there is a trade-off between using DVFS to save energy and network performance, which can directly impact on applications' Quality of Service (QoS) and Service Level Agreement (SLA). In addition, an important factor that may be impacted by this trade-off is the Quality of Experience (QoE). This parameter allows to measure the overall application performance from the users' point of view, showing a personal satisfaction perspective of the service offered by the service provider.

This paper aims to verify the impact of DVFS policies on network intensive applications performance running on a virtualized infrastructure (Citrix XenServer). The experiments were performed using three different DVFS policies, covering all possible configurations of processor frequencies allowed. The experiments were performed using a synthetic benchmark simulating a web application, in which an external client performs multiple requests through the network.

This paper is organized as follows: Section II introduces networking in virtualized environments and the DVFS technique; Section III presents related work; Section IV describes the testbed, the evaluation method and results obtained; finally, Section V concludes the paper and addresses future works.

II. Background

This section introduces the concepts of virtual networks and IO management in virtualized environments, in particular the Citrix XenServer, which is the platform used in our experiments. Finally, we show how the changes of processor frequency are performed by DVFS.

 ISBN: 978-1-61208-318-6

A. Virtual Networks and IO Management

Virtualization is a technique that allows the sharing of computer resources between virtual machines, each one hosting a complete operating system. Virtual machines management is performed by the Virtual Machine Monitor (a.k.a hypervisor). Each virtual machine has one, or more, virtual network interfaces, used to communicate with adjacent virtual machines (located in the same server) or machines located elsewhere.

In this paper, we use the Citrix XenServer environment in our experiments. Citrix Xen Server is a free virtualization platform, suited to build cloud infrastructures. It uses the Xen hypervisor as the core component of its architecture to provide a stable and elastic abstraction of the underlying infrastructure. In this section, we focus on two important points that may influence the network overhead: virtual network architecture and IO management.

Xen's architecture [9] is composed of a special virtual machine, called domain 0 (dom0), which is responsible for managing all other virtual machines, called domain U (domU). Dom0 has also privileged access to IO devices. The other virtual machines (domU) host regular operating systems and each one has a virtual driver which communicates with Dom0 in order to access physical IO devices.

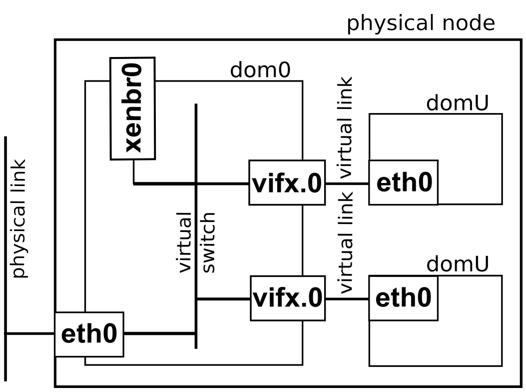

Figura 1. Virtual Network Architecture

Figure 1 shows how the virtual network is configured in Xen. Each virtual machine (domU) provides a complete hardware infrastructure, even if some devices do not exist physically or are shared by multiple virtual machines. An example of these devices is the virtual network adapter. The hypervisor may create one or more vifs (virtual interfaces) for each virtual machine, connected to a virtual link. Vifs are treated as regular NICs by the virtual machine, but in fact they only represent the interface for the physical NIC. These virtual and real networking components are connected with the use of a virtual switch. The hypervisor allows the construction of dynamic virtual network switches to enable communication between the virtual machines. Finally, the hypervisor also enables communication with the physical network infrastructure connecting the physical NICs of the server to the logical infrastructure of the hypervisor, enabling efficient communication between virtual machines, as well as with the external network.

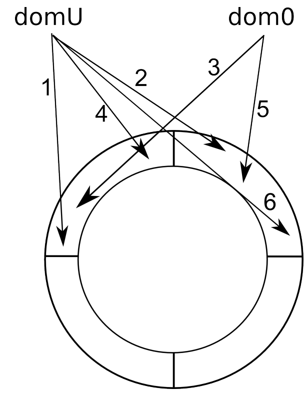

Figura 2. Ring Buffer Operations

IO is controlled by Xen through ring buffers. The data exchanged between dom0 and domUs in memory are controlled by a ring structure based on the producer-consumer model. This allows a model of locking in which there are two types of operations: request and response. Figure 2 shows how the communication occurs between dom0 and domU: (1) domU writes in the buffer a first request; (2) domU writes in the buffer a second request; (3) dom0 writes on buffer the response to the first request; (4) domU reads the dom0 answer about the first request and frees the buffer; (5) dom0 writes on buffer the response to the second request; (6) domU reads the dom0 answer about the second request and frees the buffer.

B. DVFS

Idle nodes still consume energy. Dynamic Voltage and Frequency Scaling (DVFS) is a technique that provides automatic adjustment of processor frequency with the intention to save energy. To make it happen, the processor must be able to operate in a range of frequencies, and these are adjusted according to processor utilization. Reducing the operating frequency reduces the processor performance and the energy consumption. Furthermore, reducing the voltage decreases the leakage current from the CPU's transistors, making the processor more energy-efficient resulting in further gains. Adjusting these parameters may result in a significant reduction in energy consumption per second.

Changing processor frequency decreases the number of instructions that can be executed per second, reducing overall server performance. Therefore, DVFS are usually not suitable for processes that are CPU-intensive. DVFS can be set by various operating policies such as:

- <u>Performance:</u> the frequency of the processor is always fixed at the highest, even if the processor is underutilized.
- <u>Ondemand:</u> the frequency of the processor as adjusted according to the workload behavior, within the range of frequencies allowed.
- <u>Powersave:</u> the frequency of the processor is always fixed at the smallest allowable frequency.

 ISBN: 978-1-61208-318-6

- Conservative: it has the same characteristics of the *On-demand* policy, but frequency changes are controlled, scaling gracefully between minimum and maximum according to processor utilization.
- Userspace: this allows setting a policy for each process in user space.

In order to evaluate the trade-off between DVFS operating policies and network throughput, the evaluations were performed with the three main policies: *performance*, *ondemand*, *powersave*.

III. Related Work

DVFS is a technology widely discussed in recent studies. It enables the reduction of processor frequency in order to save power when the processor utilization rate is not high, or even at times when the utilization rate changes over time.

In Takouna et al. [10] is shown that there are energy savings in virtualized clusters when used together DVFS and virtual machines consolidation. As an attempt to reduce the trade-off between energy consumption and average acceptance of jobs, a power consumption model was developed based on the number of cores, average processor frequencies and memory usage. The results show better energy savings, both in comparison with only DVFS, and DVFS with virtual machines consolidation.

Lago et al. [11] presents a strategy for resource allocation of virtual machines in a virtual cluster environment. The main focus of the work is the placement of the virtual machines in order to provide better cooling of the cluster, while DVFS is used as an alternative to decrease cooling requirements of each node individually.

The work of Belograzov et al. [12] proposes a linear interpolation model for predicting energy-savings of DVFS in cloud computing environments. The authors developed a model on the CloudSim simulator [13] in order to show the energy savings of correctly placing virtual machines and the impact of the ondemand DVFS policy. In Beloglazov and Buyya [14], the authors complement the previous work proposing a heuristic for placement of virtual machines with the intention to save energy, while meeting QoS requirements.

Kanga [15] proposed a change in the resource scheduler of Xen in order to optimize energy savings of DVFS. In another work of Kanga [16], the author presents the implementation of the solution proposed before, focusing on the ondemand DVFS policy, making this policy suitable to virtualized environments. The paper analyzes the changes of frequency-dependent rate of processor utilization, and offers pre-defined limits that make greater energy savings based on the accuracy of these adjustments.

Differently from the related works, this work evaluates the energy consumption in virtualized environments, focusing on the trade-off between different DVFS policies and their impact on network applications.

IV. Experiments

This section presents the experiments performed, describing the testbed, benchmarks, DVFS settings, and network metrics used. Afterwards, the results obtained are presented and analyzed.

A. Testbed

Evaluations were performed on a client-server architecture, simulating a client node accessing to virtualized applications in a server node, connected by a Gigabit Ethernet network. The server used in our experiments consists of 2 Intel Xeon E5520 (16 cores in total), 2.27GHz, 16 Gb RAM. This server runs the Citrix XenServer, a well-known virtualization solution in industry. In each set of tests, DVFS was configured with three operating policies: *performance*, *ondemand*, and *powersave*. The energy consumption was obtained using a multimeter which is connected between the power source and the server. This device (EZ-735 digital multimeter) has a USB connection that allows periodic external reading and gives the values of power consumption in watts-per-hour.

The network performance metrics evaluated during the experiments were: throughput and jitter. Throughput is the value that indicates the effective data rate transfer per second, while jitter is the variation in delivery time of packets in a given space of time. This variation is directly related to the network demand. The evaluation of throughput focused on the impact in energy savings and response time to the user. The evaluation of jitter aimed at analyzing the impact of the virtualization layer in the variation of data packets delivery, which consequently impacts on energy waste.

The experiment architecture is described in Figure 3. The client part of the benchmark performs requests, using the network, to applications hosted on two distinct virtual machines. Each virtual machine is associated with one of the two processors available, forcing that changes in frequency of both processors can directly influence each virtual machine, and consequently, the application within each one of them.

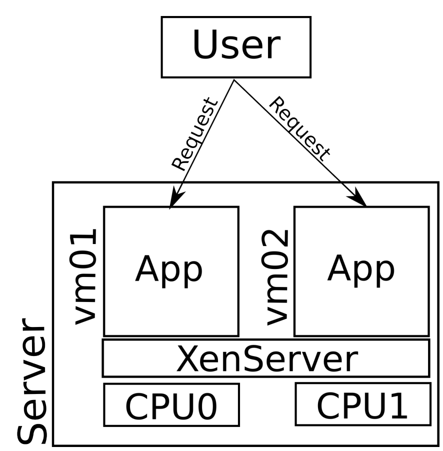

Figura 3. Experiment Architecture

Evaluations of the trade-off between the impact of changes in processor frequency and network throughput was evaluated and monitored through the benchmarks: Hping [17], T50 [18], Apache-Bench [19] and Iperf [20].

The first benchmark used was Hping. This benchmark is a packet generator that is used to analyze TCP/IP protocols. Currently, in its 3rd version, hping is one of the standard tools for security auditing and testing of firewalls and networks. Hping is programmable using the Tcl language, which allows programmers to develop their own scripts for manipulation and analysis packages.

The second benchmark used was T50 Sukhoi PAK FA Mixed Packet Injector. This tool was developed for the purpose of packet injection, designed primarily to test DoS/DDoS attacks. From the basic use of stress testing, T50 is capable of sending requests as follows: a value higher than one million packets per second of SYN Flood (+50% of the uplink network) to a network 1000BASE-T (Gigabit Ethernet) and more than 120.000 packets per second of SYN Flood (+60% of the network uplink) in a 100BASE-TX (Fast Ethernet). Additionally, it can send Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) protocols sequentially (with only microseconds difference). It is licensed under the GPL version 2.0.

The third benchmark used was Apache-Bench. This benchmark can measure the Hypertext Transfer Protocol (HTTP) server performance, running concurrent requests, and is especially efficient for test environments where Apache runs on multicore. The metric to be evaluated consists of requests per second at a given time interval, allowing to visualize the impact of various hardware components on web server performance.

The last benchmark used was Iperf. This benchmark is used to test various network metrics such as bandwidth and jitter, which can perform packet injection (TCP and UDP) to measure the performance of these networks. This tool was developed by Distributed Applications Support Team (DAST) and the National Laboratory for Applied Network Research (NLANR), and it can run on many platforms, including Linux, Unix, and Windows.

B. Results

The first evaluation shown in Figure 4 presents the virtualized server performance to answer requests in a given time interval. The results show that performance and ondemand policies kept the 10000 requests, ending his run in a shorter time than powersave which managed to answer on average smaller requests. The ondemand policy takes a little more time to complete its execution when compared to the performance policy, as there is an overhead in setting the frequencies to the behavior of the application. The powersave policy behavior is an expected result because the processor frequency is limited to one lower than the other two policies.

Figure 5 shows that there is little difference in energy consumption between performance and ondemand policies. This happens according to the benchmark behavior, which always tries to keep the processing to the highest during the test period. Therefore, the frequency variation that enables

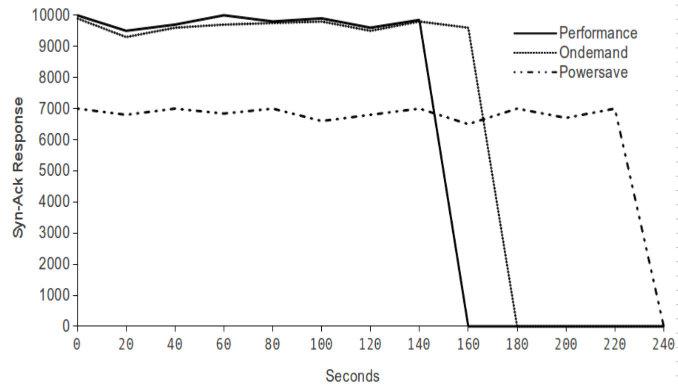

Figura 4. Hping Performance

ondemand policy is quite limited. A big difference could be seen in a case where there is a low rate of requests and, consequently, a low rate of processor utilization. However, there is a significant difference between these two DVFS policies and the powersave policy. Despite this policy save around 10% of energy, there is an increase in response time by 70%.

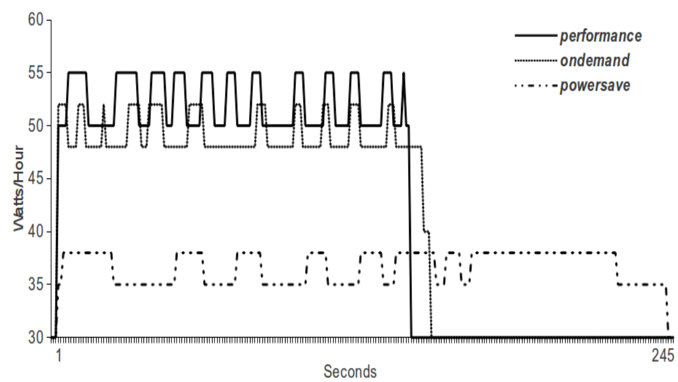

Figura 5. Hping Power Consumption

The second benchmark (T50) tested again the performance of the web server, through a flood of requests, trying to keep for a certain period of time, the most supported requests. The performance results can be seen in Figure 6. Performance and ondemand policies managed to keep the service in an average time of 150 seconds. Instead, the powersave policy was able to answer only an average between 6000 and 7000 requests over a period of about 68% higher.

The T50 benchmark shows similar results in power consumption behavior. These results can be seen in Figure 7. Again, there is no significant difference between the performance and ondemand policies. Regarding powersave policy, this enables energy savings of 15% when compared to the performance policy.

Tests using Apache-Bench perform requests to a real HTTP server. In this experiment, were performed a range between 100 and 1000 requests per second, evaluating how many milliseconds would lead the server to respond to all of them. Figure 8 shows the higher the number of requests, the greater

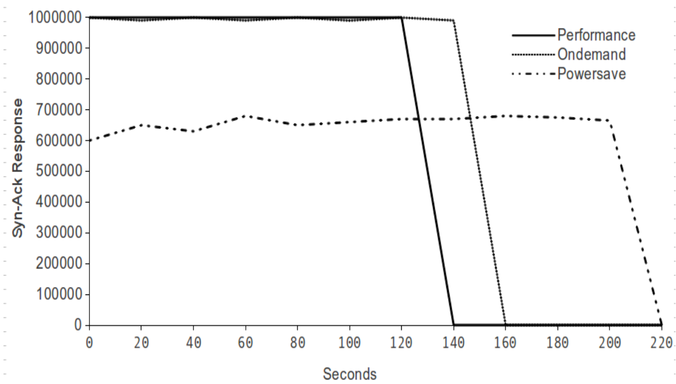

Figura 6. T50 Performance

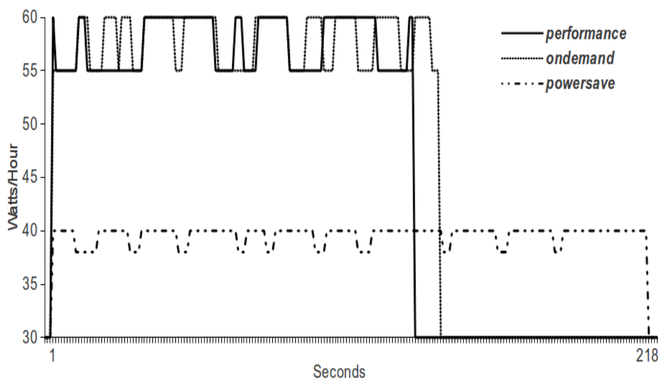

Figura 7. T50 Power Consumption

the response time in milliseconds. The ondemand policy is very near to the response times achieved by the performance policy. Both have a response time for all cases on average 35% faster than the powersave policy, which really shows that the frequency of the processor directly affects the network performance applications.

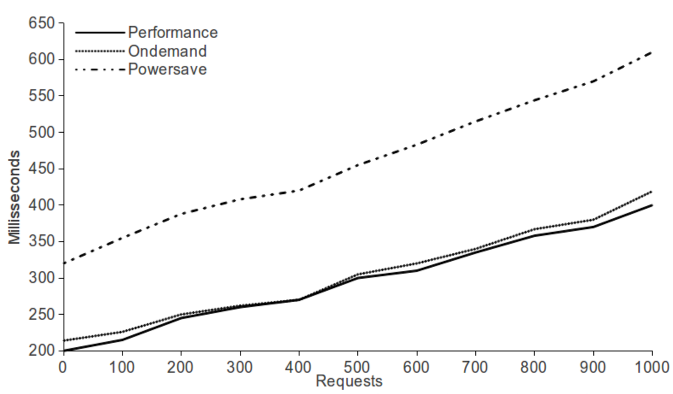

Figura 8. Apache-Bench Performance

Concerning power consumption, Figure 9 shows that performance and ondemand policies try to keep the highest processor utilization during the execution time of the application, to respond to the requests in the shortest time possible. With the limited frequency of the processor in powersave policy, there is much energy-saving, although its impact is significant on performance.

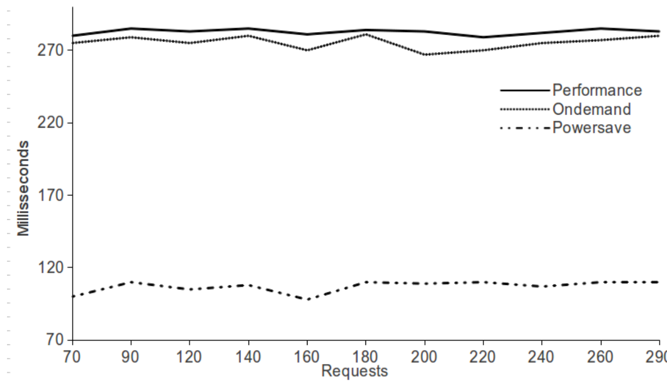

Figura 9. Apache-Bench Power Consumption

Figure 10 shows the jitter test. In these tests, DVFS policies from a native linux environment were compared to virtualized DVFS policies. The results showed that there are differences when comparing jitter on the environment in any of the native DVFS policies against a virtualized environment. Based on this, it can be verified that virtualized environments causes jitter overhead, which can cause an inefficient service for certain types of applications, such as video streaming. Furthermore, there is also a greater impact when using the powersave policy. This is probably due to the delay imposed by the structure of the ring buffer from Xen.

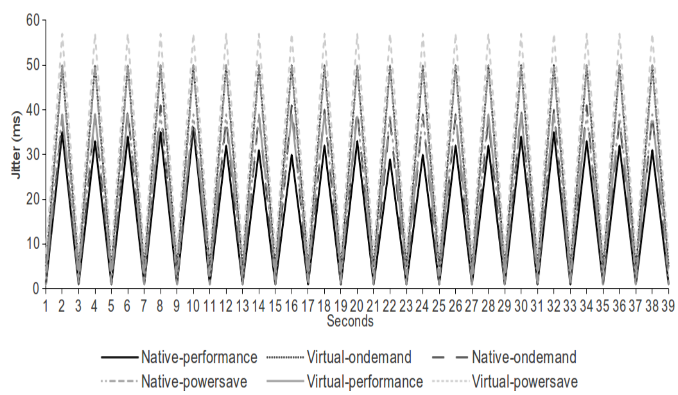

Figura 10. Iperf Jitter Evaluation

The evaluations performed allowed an examination on issues of QoS for virtualized networks. The QoS is defined in terms of the Service Level Agreements (SLA) with features such as the least throughput, maximum response time or latency time. A network architecture that can manage traffic dynamically according to SLAs is not only important for the future competitiveness, but can also set the basis for a systematic approach to energy efficiency. However, the implementation of QoS can actually increase the total network traffic and energy consumption of their virtualized environments.

The tests showed that by decreasing the bandwidth, latency increases. From the point of view of energy consumption, it is necessary to improve the latency by increasing the bandwidth, which directly impacts on energy consumption.

On this point, it must be dealt aspects such as component choice and consolidation of I/O. Likewise, it is necessary to investigate networks without loss in performance compared to the bandwidth and energy efficiency. For example, package lossless network protocols usually means more complex and more latency, as well as more processing power and low-bandwidth efficiency.

V. Conclusion and Future Work

In February 2007, the main leaders of the IT industry have announced The Green Grid, a nonprofit consortium whose mission is to improve the energy efficiency of data centers and business ecosystems based on computing. The strategy is to encourage the development of chips, servers, networks and other solutions that consume energy more efficiently.

Some of these efforts have focused on technologies such as virtualization. However, virtualization technology incurs in a processing overhead, through the addition of an abstraction layer that translates all requests between the virtual machine and physical host. This layer is affected by other technologies that attempt to promote energy-savings, such as DVFS.

This paper evaluated the impact of DVFS on network-dependent applications in virtualized environments, focusing on network performance. The choice of this metric is justified by the impact on response time for user applications. Furthermore, we also evaluated the overhead of the virtualization layer on jitter, a metric that can impact on energy waste, as well as quality of service.

As future work, we intend to evaluate changing some network parameters, such as the application buffer size, system buffer size, and Maximum Transmission Unit (MTU), that might influence in power consumption, as well as the influence of network throughput.

Referências

[1] G. Mone, "Redesigning the data center," vol. 55, no. 10. New York, NY, USA: ACM, Oct. 2012, pp. 14–16.

[2] R. Nathuji and K. Schwan, "Virtualpower: coordinated power management in virtualized enterprise systems," vol. 41, no. 6. New York, NY, USA: ACM, Oct. 2007, pp. 265–278.

[3] Q. Zhu, J. Zhu, and G. Agrawal, "Power-aware consolidation of scientific workflows in virtualized environments," in *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1–12.

[4] C. Humphries and P. Ruth, "Towards power efficient consolidation and distribution of virtual machines," in *Proceedings of the 48th Annual Southeast Regional Conference*, ser. ACM SE '10. New York, NY, USA: ACM, 2010, pp. 75:1–75:6.

[5] V. Spiliopoulos, G. Keramidas, S. Kaxiras, and K. Efstathiou, "Poster: Dvfs management in real-processors," in *Proceedings of the international conference on Supercomputing*, ser. ICS '11. New York, NY, USA: ACM, 2011, pp. 373–373.

[6] H. Hanson, S. W. Keckler, S. Ghiasi, K. Rajamani, F. Rawson, and J. Rubio, "Thermal response to dvfs: analysis with an intel pentium m," in *Proceedings of the 2007 international symposium on Low power electronics and design*, ser. ISLPED '07. New York, NY, USA: ACM, 2007, pp. 219–224.

[7] G. Mateescu, "Overcoming the processor communication overhead in mpi applications," in *Proceedings of the 2007 spring simulation multiconference - Volume 2*, ser. SpringSim '07. San Diego, CA, USA: Society for Computer Simulation International, 2007, pp. 375–378.

[8] T. Brecht, G. J. Janakiraman, B. Lynn, V. Saletore, and Y. Turner, "Evaluating network processing efficiency with processor partitioning and asynchronous i/o," in *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, ser. EuroSys '06. New York, NY, USA: ACM, 2006, pp. 265–278.

[9] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," vol. 37, no. 5. New York, NY, USA: ACM, Oct. 2003, pp. 164–177.

[10] I. Takouna, W. Dawoud, and C. Meinel, "Energy efficient scheduling of hpc-jobs on virtualize clusters using host and vm dynamic configuration," vol. 46, no. 2. New York, NY, USA: ACM, Jul. 2012, pp. 19–27.

[11] D. G. d. Lago, E. R. M. Madeira, and L. F. Bittencourt, "Power-aware virtual machine scheduling on clouds using active cooling control and dvfs," in *Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science*, ser. MGC '11. New York, NY, USA: ACM, 2011, pp. 2:1–2:6.

[12] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 826–831.

[13] 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," vol. 41, no. 1. New York, NY, USA: John Wiley & Sons, Inc., Jan. 2011, pp. 23–50.

[14] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 826–831.

[15] C. M. Kamga, G. S. Tran, and L. Broto, "Power-aware scheduler for virtualized systems," in *Green Computing Middleware on Proceedings of the 2nd International Workshop*, ser. GCM '11. New York, NY, USA: ACM, 2011, pp. 5:1–5:6.

[16] C. M. Kamga, "Cpu frequency emulation based on dvfs," in *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, ser. UCC '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 367–374.

[17] S. Sanfilippo. (2013, Oct.) Hping. [Online]. Available: http://www.hping.org

[18] N. Brito. (2013, Oct.) T50. [Online]. Available: http://t50.sourceforge.net

[19] A. Foundation. (2013, Oct.) Apache-bench. [Online]. Available: http://httpd.apache.org/docs/2.4/programs/ab.html

[20] NLANR/DAST. (2013, Oct.) Iperf. [Online]. Available: http://iperf.sourceforge.net