Optimizing Power and Energy Efficiency in Cloud Computing

Naveed Khan
naveid.khan@outlook.com
Oslo Metropolitan University (OsloMet)
Oslo, Norway

Hårek Haugerud
haugerud@oslomet.no
Oslo Metropolitan University (OsloMet)
Oslo, Norway

Raju Shrestha
raju.shrestha@oslomet.no
Oslo Metropolitan University (OsloMet)
Oslo, Norway

Anis Yazidi
anisy@oslomet.no
Oslo Metropolitan University (OsloMet)
Oslo, Norway

ABSTRACT

With the exponential growth in cloud computing, the steadily increasing amount of power consumption due to the use of physical and virtual machines is becoming a serious challenge. In this context, we report a study on optimizing the power and energy efficiency of physical and virtual machines in a cloud computing environment. The energy profile of different workloads is thoroughly investigated under different configurations. This paper presents the findings from our study which provides a good understanding of how different workloads affect power and energy efficiency of both physical and virtual machines.

KEYWORDS

Cloud computing, Virtual Machines, Energy Efficiency, Optimization

1 INTRODUCTION

In today's cloud era, an extensive amount of cloud applications are being developed and used by all kinds of businesses from small to large scale [4, 15, 25]. These applications are hosted in the cloud and consume a large amount of resources from computation to storage and bandwidth. To fulfill the increasing demand of resources, cloud service providers are expanding their infrastructure [2, 29, 35, 36, 42, 43]. The dramatic increase in the number and size of cloud data centers is impacting the global electric power consumption on our planet [5, 25]. Power consumption depends on other factors also, such as complexity of cloud infrastructure design, physical hardware and operation.

The amount of electric power consumed by data centers and their corresponding carbon footprint are becoming an environmental challenge for our planet. Data centers are predicted to consume around 150 billion kWh of power by 2020 and around 1430 million metric tons of carbon emissions. Data centers will be responsible for about 18% of the carbon emission [1]. Virtualization helps in reducing energy consumption by allowing scalable virtual machines to be hosted on a single physical machine [13, 30].

In this perspective, the field of green computing has attracted a lot of research interest during the last years [33, 45]. The advent of virtualization technology has offered a set opportunities for the cloud computing providers to optimize the use of physical hardware in the cloud computing data centers. However, virtualization technology comes short when we consider the idle state of virtual machines, as virtual machines consume resources even in the idle state.

There is variety of techniques that help in reducing power consumption [9, 20, 32, 40]. Dynamic voltage frequency scaling (DVFS) is a technique that operates by both adjusting the frequency of the CPU and voltage. In this context, a per-core DVFS techniques was suggested in [8] to reduce power consumption. Vary-on/Vary-off (VOVO) technique powers on and off the server, based on server workload to reduce wasting of energy [40]. There exist workload-aware consolidation techniques that reduce energy consumption using consolidation fitness metric [38].

In this paper, we have studied and presented the findings on energy efficiency of physical and virtual machines in cloud computing data center. We have addressed a problem of energy efficiency optimization by tuning the CPU of both physical and virtual machines. The main objective is to find an answer to how energy efficiency can be optimized by tuning the resources of physical and virtual machines in a cloud computing environment. Jiang et al.[16] compared the energy efficiency of different hypervisors. In this work, we are comparing energy efficiency of physical and virtual machines. Then we attempt to find an optimal frequency that can be selected for achieving the highest energy efficiency.

2 RELATED WORK

Virtualization technology has been considered as one of the key enabling technologies behind cloud computing. Cloud computing data centers host tens of thousands of clusters of servers with the help of virtualization technology to reduce hardware footprint. On the other hand, these data centers consume a huge amount of energy to keep up their services.

A lot of research has been done in this direction which deals with performance analysis of virtualized platforms and measuring energy efficiency. The work reported in [7] presented a detailed study related to virtualized systems performance-energy trade-off. They suggested that virtualization overhead, performance-energy trade-off, and virtual machine interference are some of the key areas that need more attention from the research community. In [12], a model is presented to predict performance of different virtualization platforms by evaluating the behavior of different workloads to see their mutual influence. The authors showed that CPU and memory performance is the same for both systems while there are deviations in the case of input/output (I/O). Understanding energy efficiency and power usage of hardware systems is necessary in order to properly perform capacity planning of hardware. Jiang et al. [16] presented a study on energy efficiency by comparing hypervisors on different servers. They showed that using the same hardware for different hypervisors exhibits different power and energy characteristics.

Energy efficiency can be measured along with resource usage showing the relationship between power consumption and server load [26]. It was found that homogeneous virtual machines finished the job with the same load and time while heterogeneous virtual machines finished jobs faster with less powerful virtual machines, and the power consumed by virtual machines with the highest degree of heterogeneity is twice compared to homogeneous virtual machines.

Using power metering methods at the virtual machine level and identifying inefficiency in big applications provide data for managing power efficiency. In [27] the efficiency of data-intensive applications was shown and penalties introduced in terms of power consumed by virtual machines in a cluster. To investigate the performance of data-intensive applications and evaluate the power consumed, a Hadoop MapReduce cluster was implemented. Results showed that homogeneous virtual machines with the same load finished the transfers at the same time while heterogeneous virtual machines with high-performance characteristics finished the transfers faster and then waited for low-performance characteristics virtual machines to finish the transfer. In this case, high characteristics virtual machines remain idle with no useful jobs while consuming power.

In [40], a combined frequency scaling and application elasticity approach for energy-efficient cloud computing using three techniques to control a cloud data center in an energy-efficient manner was proposed. A feedback controller for configuration is used to reduce power consumption while maintaining system performance. A power meter is attached to the power distribution unit (PDU) to measure power consumption. Power scaling techniques that are horizontal, vertical and hard power scaling have been identified to understand and investigate their impact on application performance and power usage. The study achieved 34% of energy saving, meeting targeted performance when each policy is applied in isolation.

To reduce power at the processor level, Intel SpeedStep technology allows the processor to adjust frequency and voltage at the hardware level [14]. AMD PowerNow technology controls processor power consumption by controlling frequency and voltage on the fly [3]. AMD Cool'n Quite PowerNow permits to reduce both noise and heat [44]. Some other popular techniques include dynamic voltage scaling and workload consolidation which help in the reduction of power consumption [31, 34]. Liang et al. [20] have shown that running a processor at a specific frequency instead of running at the lowest frequency can achieve optimal operation speed and optimal energy efficiency. They found that a modified form of DVFS governor lowers power consumption. Energy-aware scheduling refers to another research direction to reduce power consumption. A representative example that falls under the latter research direction is reported in [21] where the authors proposed an energy-aware virtual machine scheduling and consolidation algorithm. The study reports evidences that a dynamic round robin algorithm reduces 43.7% of power compared to other scheduling algorithms. In [38], Sharifi et al. presented a scheduling algorithm by assigning a set of virtual machines to a set of physical machines to reduce the total power consumption in a data center. Energy-aware workload consolidation is yet another approach in which workloads are applied to least physical machines to minimize energy consumption. In [28], a work consolidation technique using operating system and dynamic control mechanisms is presented to adjust the allocation of cores and frequency based on delays for latency critical workloads.

Energy-efficient techniques play an important role to reduce power consumption. Energy management policies are used at the architectural level for efficiency. Energy efficient resource management techniques that are used to reduce operational costs by saving a substantial amount of energy without reducing the quality of service (QoS) have been shown to be effective [1, 6].

Virtualization technology reduces power consumption in a cloud data center if virtual machines are properly provisioned and proper management policies are deployed. Still, virtual machines placement is a big challenge as it impacts energy efficiency and QoS significantly [17]. According to [6], leveraging virtual machines live migration and dynamically reallocating them yields a substantial amount of energy saving in cloud data centers.

Proper resource management can also reduce power consumption of servers that host performance-oriented applications. A resource estimation model that draws a relationship between server performance and power based on utilization ratio provides a method to reduce the power consumption. Properly adjusting resources to improve performance of applications achieved a power saving within 12% to 20% compared to scaling performance governor [41].

3 EXPERIMENTAL DESIGN

We ran a series of experiments in a custom-built environment. The setup consisted of three physical servers equipped with Intel (R) Xeon (R) Processors (7 Sockets, 28 Cores, 56 GB Physical Memory) with hyperthreading-enabled, and running a Ubuntu 16.04.3 LTS operating system. Hyperthreading takes advantage of superscalar architecture to improve parallelization of x86 processors by running multiple instructions in a clock cycle [24]. A topology diagram of the experimental data center is shown in Figure 1. The set-up consists of two hosts, Host-16 with 16 cores and Host-4 with 4 cores, and a monitoring server. An external rack-mounted APC PDU [37] was used for power supply to the server machines. Power consumption data is collected from the PDU using the simple network management protocol (SNMP).

The cpufrequtils package [9] was used for scaling CPU frequency. The frequency of the CPU cores was scaled between 0.8 GHz and 3.00 GHz. Scaling Governors in Linux are power management policies that control the CPU frequency. To allow dynamic scaling of frequency, the cpufreq interface was used to adjust CPU frequency. The Scaling Governor was changed from the default on-demand to allow power management policies based on dynamic voltage frequency scaling [9, 20, 32].

To provision virtual machines, we used QEMU with the KVM module, which is a full virtualization solution commonly used in Linux-based systems.

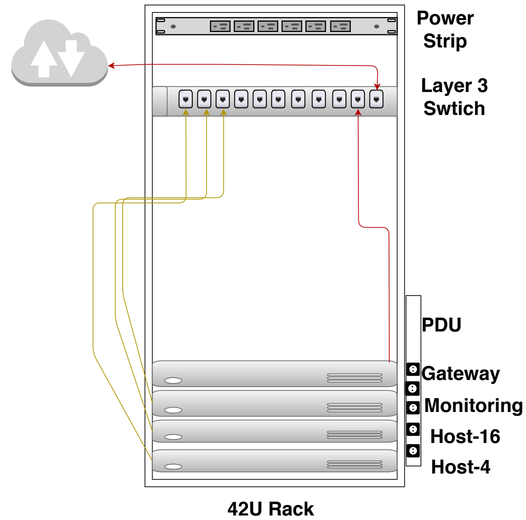

Figure 1: Topology of experimental setup

To study the power usage under realistic workloads, we used Minerd [11], which is a CPU miner used for Litecoin Bitcoin mining. First, we ran the stress tool [39] as our benchmark tool that imposes stress on the systems to see the power used when the CPUs are fully utilized. Later, we ran the Minerd application to see the difference in power usage. The output power was visualized using ELK stack [10, 18, 23], an open-source stack for log-aggregation and analytics.

Since changing the number of cores of the virtual machines at the time was not possible on the fly using the KVM hypervisor [40], we rather started and stopped the virtual machines. The frequency of individual CPU cores was changed while pinning virtual machine cores to physical cores using core pinning, more particularly, core affinity [12].

Power and energy are related as follows [1]:

$$P = E/T \tag{1}$$

where P is total power in Watts, E is energy in Joules, and T is the time in seconds. The results of the minerd bitcoin-miner is given as hashes/second and represents the efficiency of the process. The metric we use for energy efficiency in this paper is the hashing efficiency divided by power, giving hash/Joule as a measure of how much work the algorithm is able to do per unit of energy.

We ran several experiments using different physical and virtual machine combinations with the frequency of CPU cores set between 1.6 and 2.4 GHz. The number of threads was set to 1, 4, 8, and 16 for each physical and virtual machine running on different frequencies while hyperthreading was turned on.

4 RESULTS AND DISCUSSION

Figure 2 shows power consumption resulting from the stress test on CPU, memory, and disk in case of no-virtualization and virtualization, in Host-16. Similarly, Figure 3 shows the test results in the case of Host-4. In both figures, we see power consumption dropping down every 5 minutes, which was because the stress-tool was stopped and restarted every 5 minutes. From these figures, we see that the results in terms of power consumption of heterogeneous servers with different resources are different in magnitude, but at the same time the results are also quite similar. The average total power consumption for the larger Host-16 server was 252 watts when stressing the CPU while for the Host-4 server the average power consumption was 81 watts. The results when running on bare metal and when using virtualization are quite similar.

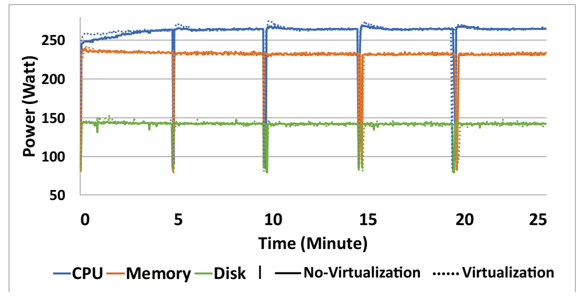

Figure 2: Power consumption under stress test on Host-16, with and without virtualization.

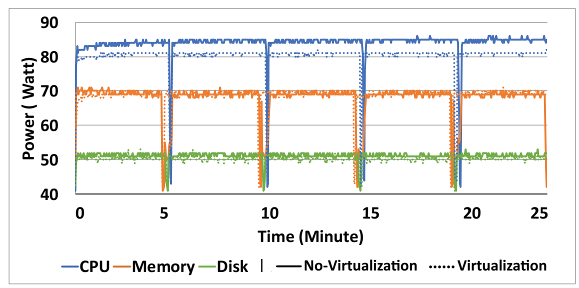

Figure 3: Power consumption under stress test on Host-4, with and without virtualization.

In order to investigate the energy efficiency, we ran a CPU intensive workload to study the impact on power consumption and how much of the total energy was useful. The workload was a CPU miner, which performed the jobs based on hashes per second. The frequency of CPU cores was scaled between 1.6 GHz to 2.4 GHz and variations in threads were set to 1, 4, 8, 12 and 16, both in case of virtualization and no-virtualization. Figure 4 shows power consumption at different frequencies and the different number of threads in both cases. Similarly, Figure 5 shows energy efficiency in terms of Hash/Joule for the same combinations.

From Figure 4, we can observe that as the number of CPUs is increased, power consumption also increases. It can also be observed that increasing the threads improves the performance of an application and that is because multiple threads generate higher hash rate in this case. On the other hand power consumption also increases when threads are increased.

Figure 5 shows the highest energy efficiency of 0.67 Hash/ Joule at frequency 2.26 GHz and 16 threads, in case of no-virtualization. When the same number of CPU cores and the same frequency were assigned to the virtual machine, we can see that a 10% performance

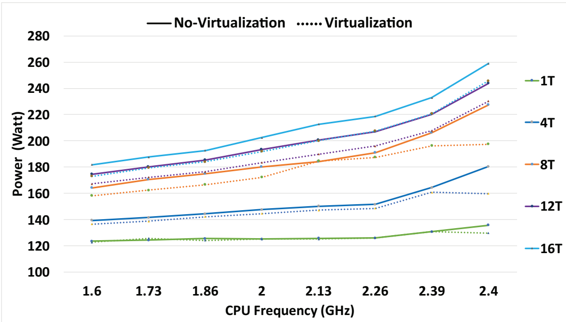

Figure 4: Power consumption with different CPU frequencies and different number of threads.

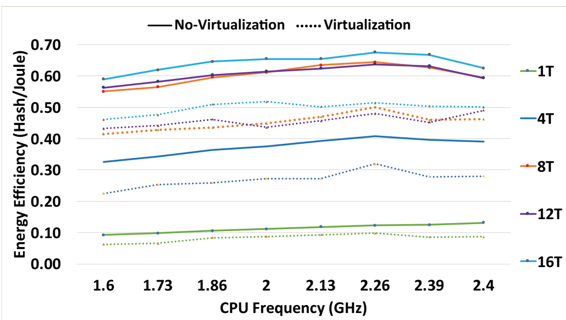

Figure 5: Energy efficiency with different CPU frequencies and different number of threads.

decrease has been recorded when running the workload inside the virtual machine. The workload generated 0.5 Hash/Joule with 16 threads while running the CPUs at 2.26 GHz frequency.

Next, we have investigated the influence of the architecture of the underlying system on energy efficiency using non-uniform memory access (NUMA), which is a method used to increase the performance and capability of a multi-processor system allowing a processor to access local memory faster compared to non-local memory [19]. The command line tool *numactl* [22] was used to bind processes to physical CPUs, allowing control of policies for processes and shared memory. Figure 6 and 7 show the power consumption and energy efficiency respectively for 1, 4 and 8 threads with a CPU frequency range from 1.6 to 2.4GHz. Comparing Figure 7 with Figure 5, we see that there is no real difference between these two cases, which is as expected as the memory usage of the minerd process is not extensive. Forcing the process to use local memory, labeled as LM in the figures, does also not lead to any difference in energy efficiency. For workloads which are more memory demanding, this kind of binding could make a difference.

In Figure 8 the resulting energy efficiency of using the optimal value of the CPU frequency is compared to that of the operating system scaling governors *Performance* and *Powersave* on physical and virtual machines. The figure shows that with the optimal frequency of 2.26 GHz in both cases gives the highest energy efficiency. Table 1 and Table 2 provide comparative results on power usage and energy efficiency. The results were better in terms of energy

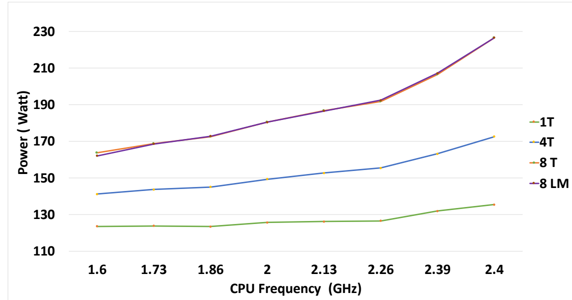

Figure 6: Power consumption with the physical machine using NUMA scheduled workload. LM means restricted to local memory.

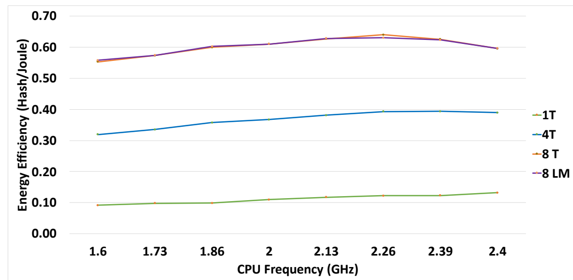

Figure 7: Energy efficiency with the physical machine using NUMA scheduled workload.

efficiency on physical machines at the optimal CPU frequency of 2.26 GHz. In case of virtualization when comparing the optimal frequency 2.26 GHz to the scaling governors as shown in Figure 8, energy efficiency is decreased by 0.14 Hash/Joule while still the optimal frequency 2.26 performs better compared to the *Performance* and *Powersave* governors.

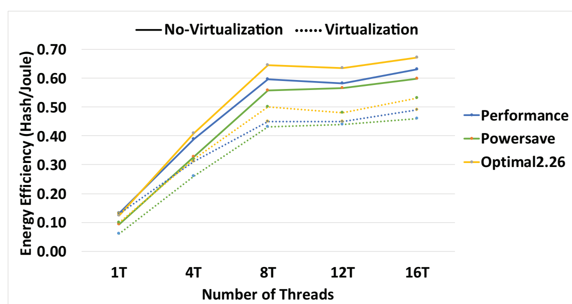

Figure 8: Energy efficiency for the two scaling governors and the optimal frequency.

We also ran the application workload to measure the energy efficiency in the cases of enabling and disabling hyperthreading and the results are shown in Figure 9. When hyperthreading was disabled on the physical machine (pNHT), energy efficiency was identical for 1, 4 and 8 threads compared to hyperthreading enabled (pHT). This is as expected since the server has 8 hyperthreading cores. In both cases the threads will run on separate cores when

Table 1: Average power usage (in Watts) for various scaling governors and number of threads.

	1T	4T	8T	12T	16T
Performance	135	182	215	231	247
Powersave	123	138	158	166	173
Optimal 2.26	126	149	187	196	207

Table 2: Power and energy efficiency for physical and virtual machines.

	Frequency (Ghz)	Physical machine		Virtual machine	
		Power (Watt)	Efficiency (Hash/Joule)	Power (Watt)	Efficiency (Hash/Joule)
Performance	1.6-2.4	229	0.60	214	0.45
Powersave	1.6-2.4	162	0.56	158	0.43
Optimal 2.26	2.26	191	0.64	187	0.50

there are 8 or less of them and then there will be no hyperthreading taking place. When running more than 8 threads with hyperthreading disabled, this will lead to multitasking between two processes on the same CPU which in general is less efficient than hyperthreading. And as seen in Figure 9 hyperthreading leads to higher efficincy for 16 threads. For 4 and 8 threads the virtualized server performs slightly better when hyperthreading is turned off. This is not as expected but the difference is small and further investigations are needed in order to establish whether this is not just a statistical variation.

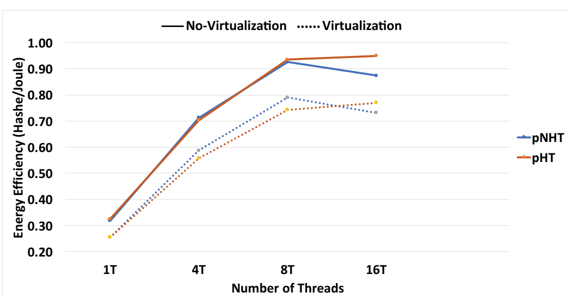

Figure 9: Energy efficiency with (red) and without (blue) hyperthreading.

When running a CPU bound workload, our results show that in this specific case it is possible to find an optimal frequency that gives higher energy efficiency than any other frequency. The results also show that running at this frequency gives a higher efficiency than when running using frequency scaling governors like *Powersave* or *Performance*. The latter is typically the scaling governor run on a normal server, unless the server is configured to always run on maximum frequency for obtaining the maximum performance. However, our results show that compared to running at the maximum frequency, reducing it slightly might lead to a higher energy efficiency. In general the results indicate that utilizing as many of the cores of the system as possible is the most energy efficient approach. When there are more workload threads than cores of the underlying physical server, our results show that hyperthreading improves not only the performance but also makes the system as a whole more energy efficient.

5 CONCLUSION AND FUTURE WORK

In this paper, we performed experiments and in-depth analysis of the results obtained to see the power and energy efficiency of physical and virtual machines. Our experimental results and observations demonstrated that properly scaling the resources of a physical and virtual machines can reduce power consumption and improve energy efficiency. The methods and frameworks provided in this work not only enable measuring accurate power of each CPU that runs workload and calculating power consumption and energy efficiency but also provide methods for determining energy-efficient configurations. This study can serve as a guideline to improve energy efficiency in any cloud environment of any size.

For future work, these methods and framework presented here can be extended to large data centers to maximize energy efficiency while minimizing power consumption. It would be interesting to design a scheduler that can handle scaling of frequencies automatically and to find an optimal frequency for the current workload in order to achieve improved energy efficiency.

REFERENCES

[1] Nasrin Akhter and Mohamed Othman. 2016. Energy aware resource allocation of cloud data center: review and open issues. *Cluster Computing* 19, 3 (2016), 1163–1182.

[2] Amazon. 2018. *Start Building on AWS Today.* Online https://aws.amazon.com, Last access: Feb. 2019.

[3] AMD. 2000. Technology Platform Design Guide for Embedded Processors. *AMD Document* 24267a (2000).

[4] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (2010), 50–58.

[5] Angela Bartels. 2011. *Data Center Evolution: 1960 to 2000.* Online https://blog.rackspace.com/datacenter-evolution-1960-to-2000, Last access: Feb. 2019.

[6] Anton Beloglazov and Rajkumar Buyya. 2010. 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*. IEEE Computer Society, 826–831.

[7] Belen Bermejo, Carlos Juiz, and Carlos Guerrero. 2018. Virtualization and consolidation: a systematic review of the past 10 years of research on energy and performance. *The Journal of Supercomputing* (2018), 1–29.

[8] Sridutt Bhalachandra, Allan Porterfield, Stephen L Olivier, Jan F Prins, and Robert J Fowler. 2017. Improving energy efficiency in memory-constrained applications using core-specific power control. In *Proceedings of the 5th International Workshop on Energy Efficient Supercomputing*. ACM, 6.

[9] Dominik Brodowski. 2013. CPU frequency and voltage scaling code in the Linux kernel. Online https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt, Last access: Feb. 2019.

[10] Elasticsearch. 2018. *The Heart of the Elastic Stack.* Online https://www.elastic.co/products/elasticsearch, Last access: Feb. 2019.

[11] FreeBSD. 2018. *FreeBSD Man Pages - MINERD* . Online https://www.gsp.com/cgi-bin/man.cgi?section=1&topic=minerd, Last access: Feb. 2019.

[12] Nikolaus Huber, Marcel von Quast, Michael Hauck, and Samuel Kounev. 2011. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments.. In *CLOSER*. 563–573.

[13] IBM. 2018. *Virtualization in Education.* Online http://www-07.ibm.com/solutions/in/education/download/Virtualization%20in%20Education.pdf, Last access: Feb. 2019.

[14] Intel. 2017. *Enhanced Intel SpeedStep® Technology for Intel® Processors.* Online https://www.intel.com/content/www/us/en/support/processors/000007073.html, Last access: Feb. 2019.

[15] Yashpalsinh Jadeja and Kirit Modi. 2012. Cloud computing-concepts, architecture and challenges. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*. IEEE, 877–880.

[16] Congfeng Jiang, Yumei Wang, Dongyang Ou, Youhuizi Li, Jilin Zhang, Jian Wan, Bing Luo, and Weisong Shi. 2017. Energy efficiency comparison of hypervisors. *Sustainable Computing: Informatics and Systems* (2017).

[17] Atefeh Khosravi, Lachlan LH Andrew, and Rajkumar Buyya. 2017. Dynamic VM placement method for minimizing energy and carbon cost in geographically distributed cloud data centers. *IEEE Transactions on Sustainable Computing* 2, 2 (2017), 183–196.

[18] Kibanana. 2018. *Introduction*. Online https://www.elastic.co/guide/en/kibana/current/introduction.html#introduction, Last access: Feb. 2019.
[19] Christoph Lameter. 2013. *NUMA (Non-Uniform Memory Access): An Overview*. Online https://queue.acm.org/detail.cfm?id=2513149, Last access: Feb. 2019.
[20] Wen-Yew Liang, Ming-Feng Chang, and Yen-Lin Chen. 2017. Optimal Energy Saving DVFS Approach of Embedded Processors. *Journal of Information Science & Engineering* 33, 5 (2017).
[21] Ching-Chi Lin, Pangfeng Liu, and Jan-Jan Wu. 2011. Energy-aware virtual machine dynamic provision and scheduling for cloud computing. In *2011 IEEE International Conference on Cloud Computing (CLOUD)*. IEEE, 736–737.
[22] Linux Manual. 2018. *Linux Administrator's Manual*. Online http://man7.org/linux/man-pages/man8/numactl.8.html, Last access: Feb. 2019.
[23] Logstash. 2018. *Centralize, Transform & Stash Your Data*. Online https://www.elastic.co/products/logstash, Last access: Feb. 2019.
[24] Debbie Marr, Frank Binns, D Hill, Glenn Hinton, D Koufaty, et al. 2002. Hyper-threading technology in the netburst® microarchitecture. *14th Hot Chips* (2002).
[25] Peter Mell, Tim Grance, et al. 2011. The NIST definition of cloud computing. (2011). National Institute of Standards and Technology.
[26] Catalin Negru, Mariana Mocanu, and Valentin Cristea. 2015. Impact of Virtual Machines Heterogeneity on Data Center Power Consumption in Data-Intensive Applications. In *International Workshop on Adaptive Resource Management and Scheduling for Cloud Computing*. Springer, 91–102.
[27] Catalin Negru, Mariana Mocanu, Valentin Cristea, Stelios Sotiriadis, and Nik Bessis. 2017. Analysis of power consumption in heterogeneous virtual machine environments. *Soft Computing* 21, 16 (2017), 4531–4542.
[28] Efthymios Oikonomou, Dimitra Panagiotou, and Angelos Rouskas. 2015. Energy-aware Management of Virtual Machines in Cloud Data Centers. In *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks*. ACM, 35.
[29] Openstack. 2019. *Open source software for creating private and public clouds*. Online https://www.openstack.org, Last access: Feb. 2019.
[30] Oracle. 2018. *Brief History of Virtualization*. Online https://docs.oracle.com/cd/E26996_01/E18549/html/VMUSG1010.html, Last access: Feb. 2019.
[31] Venkatesh Pallipadi. 2009. Enhanced Intel SpeedStep technology and demand-based switching on Linux. *Intel Developer Service* (2009).
[32] Venkatesh Pallipadi, Shaohua Li, and Adam Belay. 2007. cpuidle: Do nothing, efficiently. In *Proceedings of the Linux Symposium*, Vol. 2. 119–125.
[33] Yashwant Singh Patel, Neetesh Mehrotra, and Swapnil Soner. 2015. Green cloud computing: A review on Green IT areas for cloud computing environment. In *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*. IEEE, 327–332.
[34] Padmanabhan Pillai and Kang G Shin. 2001. Real-time dynamic voltage scaling for low-power embedded operating systems. In *ACM SIGOPS Operating Systems Review*, Vol. 35. ACM, 89–102.
[35] Tiago Rosado and Jorge Bernardino. 2014. An overview of OpenStack architecture. In *Proceedings of the 18th International Database Engineering & Applications Symposium*. ACM, 366–367.
[36] Matthew NO Sadiku, Sarhan M Musa, and Omonowo D Momoh. 2014. Cloud computing: opportunities and challenges. *IEEE potentials* 33, 1 (2014), 34–36.
[37] Schneider-Electric. 2018. *Rack Power Distribution Unit (PDU)*. Online https://www.schneider-electric.com/en/product/AP8459EU3_APC/rack-pdu-2g, Last access: Feb. 2019.
[38] Mohsen Sharifi, Hadi Salimi, and Mahsa Najafzadeh. 2012. Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques. *The Journal of Supercomputing* 61, 1 (2012), 46–66.
[39] Stress. 2014. *Stress project page*. Online https://people.seas.harvard.edu/~apw/stress/, Last access: Feb. 2019.
[40] Selome K Tesfatsion, Eddie Wadbro, and Johan Tordsson. 2014. A combined frequency scaling and application elasticity approach for energy-efficient cloud computing. *Sustainable Computing: Informatics and Systems* 4, 4 (2014), 205–214.
[41] Selome Kostentinos Tesfatsion, Eddie Wadbro, and Johan Tordsson. 2016. Autonomic Resource Management for Optimized Power and Performance in Multi-tenant Clouds. In *Autonomic Computing (ICAC), 2016 IEEE International Conference on*. IEEE, 85–94.
[42] VMware. 2018. *Virtualization*. Online https://www.vmware.com/solutions/virtualization.html, Last access: Feb. 2019.
[43] Matt Weinberger. 2017. *The cloud wars explained: Amazon is dominating, but Microsoft and Google are striking back*. Online http://www.businessinsider.com/why-amazon-is-so-hard-to-topple-in-the-cloud-and-\where-everybody-else-falls-2017-7?r=US&IR=T&IR=T, Last access: Feb. 2019.
[44] Mark W Welker and One AMD Place. 2003. AMD processor performance evaluation guide. Online http://datasheets.chipdb.org/AMD/All%20CPU/30579_3.71.pdf, Last access: Feb. 2019.
[45] Wikipedia. 2018. *Green Computing*. Online https://en.wikipedia.org/wiki/Green_computing, Last access: Feb. 2019.