A Fuzzy analysis of CPU and DRAM energy consumption to determine load balance and scheduling of VMs in a Cloud*

Julio Machado, Guilherme Schneider, Rodrigo Duarte, Vitor Ataides, and Renata Reiser and Mauricio Pilla

Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPel)
{jmdsneto,gbschneider,rmduarte,reiser,pilla}@inf.ufpel.edu.br
http://www2.ufpel.edu.br/cdtec/

Abstract. Cloud Computing emerged a few years a go, enabling the creation of many services. Good management of datacenters resources is important because bad resource management directly impacts energy consumption, potentially leading to resource wasting and performance loss. In this paper we have made stress tests CPU and memory to see how much each one impacts in the energy consumption, with the generated results a classification machine used was made with Fuzzy logic.

1 Introduction

The energy consumption effciency in datacenters became important task since the paradigm of cloud computing became real. Cloud computing environments demand a large amount of computational power and, consequently, demands a large amount of energy consumption which diversify constantly [1]. Scheduling of VMs(Virtual Machines), which allow resources optimization. The definition of how much a machine is being used can help in this decision making. In this paper we use a Fuzzy approach as support for the decision making, allowing the classification of machines by energy consumption from CPU(Central Processing Unit) and DRAM(Dynamic Random Acess Memory), as: "underuse", "normal" or "overloaded".

NVM(non-volatile-memory) are considered an innovative promising memory alternative technique that features attractive advantages, such as high density, non-volatility, positive response to increasing temperature, zero standby leakage, and excellent scalability [2], [3], [4].

This paper is organized as follows: in section 2 we talk about cloud computing and some of its characteristics; in section 3 we briefly talk about what is Fuzzy Logic type 1 and 2 and what we can make with this; in section 4 is presented a study about static and dynamic power consumption and sources of power consumption; in section 5 presents our case study about CPU and memory energy consumption with results and a discussion about our study and in section 6 we conclude and briefly talk about future works.

* Work progress stage: completed.

2 Cloud Computing

Cloud Computing is a growing computational paradigm whose primary characteristic is a change in the way computational resources are delivered. Changing from a server based approach where the user buys hardware and rents a space in a datacenter to a web-based service where any user can buy computational time by requesting a virtualized instance from a cloud services provider [1].

With this paradigm shift the process of developing and testing innovative ideas does not have to be as expensive as it was before cloud computing. Companies can now get results quicker by using computational scalability (e.g. by using ten times as many servers for a tenth of the time).

But, even this new paradigm has your issues, like energy expensives by cooling datacenters that have machines in idle that could be poweroff until the scheduling system request a machine, or datacenters with machines overloaded by a lot of VMs causing loss performance. Trying to decrease the energy expensives of the physical servers, VM migration policies has been proposed based on some feedback received from the data center, a policy should decide when, how, and which VMs have to be migrated.

In this paper we present a Fuzzy Logic System to solve the problem of VM migration as this is a decision-making problem and Fuzzy Logic is a well known technique to lead with uncertainty obtained from the servers information and to make decisions about the VM allocation.

3 Fuzzy Logic

A Fuzzy Logic System (FLS) includes fuzzifier, rules, inference engine, and defuzzifier[5]. Quite often, the knowledge that is used to construct the rules in a FLS is uncertain. Three ways in which such rule uncertainty can occur are: 1) the words that are used in antecedents and consequents of rules can mean different things to different people; 2) consequents obtained by polling a group of experts will often be different for the same rule because the experts will not necessarily be in agreement; and 3) noisy training data. [6]

This characteristics present in FLS make this approach be an excelent model to evaluate uncertainties extracted from the infrastructures employed because of the variation on CPU and memory consumption. Another problem that FLS help to solve is the classification of the usage of the resources whereas FLS gives an fuzzified value of the energy consumption and not only a simple value.

4 Static and Dynamic Power Consumption

The major part of power consumption in complementary metal-oxide-semiconductor (CMOS) circuits comprises static and dynamic power. Static power consumption, or leakage power, is caused by leakage currents that are present in any active circuit, independently of clock rates and usage scenarios. This static power is

mainly determined by the type of transistors and process technology. The reduction of static power requires improvements of the low-level system design. More details regarding possible ways to improve energy efficiency at this level can be found in the survey by Venkatachalam and Franz [7].

Dynamic power consumption is created by circuit activity (i.e., transistor switches, changes of values in registers, etc.) and depends mainly on a specific usage scenario, clock rates, and I/O activity. The sources of dynamic power consumption are the shortcircuit current and switched capacitance. Short-circuit current causes only 10-15% of the total power consumption and so far no way has been found to reduce this value without compromising the performance. Switched capacitance is the primary source of dynamic power consumption; therefore, dynamic power consumption can be defined as 1 [8],

$$P_d =_a CV^2 f, \tag{1}$$

where a is the switching activity, C is the physical capacitance, V is the supply voltage, and f is the clock frequency. The values of switching activity and capacitance are determine by the low-level system design. The combined reduction of the supply voltage and clock frequency lies in the roots of the widely adopted DPM technique called Dynamic Voltage and Frequency Scaling (DVFS). The main idea of this technique is to intentionally scale down the CPU performance, when it is not fully utilized, by decreasing the voltage and frequency of the CPU. In the ideal case, this should result in a cubic reduction of dynamic power consumption [9].

4.1 Sources of Power Consumption

The main part of power consumed by a server is accounted for the CPU, followed by the memory and losses due to the power supply inefficiency, but a few years ago, this discrepancy started to decrease [10]. CPU in low-power mode can consumes a fraction of the total power, while preserving the ability to execute programs. As a result, current desktop and server CPUs can consume less than 30% of their peak power in low-activity modes, leading to dynamic power ranges of more than 70% of the peak power [11].

In contrast, dynamic power ranges of all the other server components are much narrower: less than 50% for Dynamic Random Access Memory (DRAM), 25% for disk drives, 15% for network switches, and negligible for other components [3]. The reason is that only the CPU supports active low-power modes, whereas other components can only be completely or partially switched off. However, the performance overhead of a transition between the active and inactive modes is substantial. For example, a disk drive in the deep-sleep mode consumes almost no power, but a transition to the active mode incurs a latency 1000x higher than the regular access latency. Power inefficiency of the server components in the idle state leads to a narrow overall dynamic power range of 30%: even if a server is completely idle, it still consumes more than 70% of its peak power.

The adoption of multi-core CPUs, that are substantially more efficient than conventional single-core processors, along with the increasing use of virtualization and data-intensive applications resulted in the growing amount of memory in servers. In contrast to the CPU, DRAM has a narrower dynamic power range, and power consumption by memory chips is increasing [9].

But, even with this energy reductions and the flexibility by the virtualization, datacenters consumes 1.4% of the whole world energy [12].

5 Case Study

Our case of study consists in to stress CPU and memory with the benchmark Stress-NG [13] and then compare to see how much which one affects in a machine energy consumption and when a VM can be migrated to a node or a machine shutdown to save energy.

All of our tests were made in a NUMA (non-uniform memory access) architecture machine, AMD Opteron with 64 cores, 128GB of memory and Ubuntu 14.02 LTS 64 bits. Our tests consists in doing stress on CPU and memory to see what is the impact in the current power consumption. For this, in both tests we use the tool stress-ng. To get the current power consumption, we use the tool free ipmi [14].

5.1 CPU test

In our CPU test, we get the current consumption while we charge all 64 cores in: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%.

Two method are chosen to make this test in CPU, the first is *cpu N:* start N workers exercising the CPU by sequentially working through all the different CPU stress methods. Instead of exercising all the CPU stress methods, one can specify a specific CPU stress method with the -cpu-method option.

And the second is *CPU-load P:* load CPU with P percent loading for the CPU stress workers. 0 is effectively a sleep (no load) and 100 is full loading. The loading loop is broken into compute time (load%) and sleep time (100% -load%). Accuracy depends on the overall load of the processor and the responsiveness of the scheduler, so the actual load may be different from the desired load. Note that the number of bogo CPU operations may not be linearly scaled with the load as some systems employ CPU frequency scaling and so heavier loads produce an increased CPU frequency and greater CPU bogo operations.

The figure 1 show the energy consumption according with the use of CPU.

5.2 Memory test

In our Memory test, we get the current consumption while we do write and read about a certain percentage of all 128GB memory: 20%, 40%, 60%, 80% and 100%.

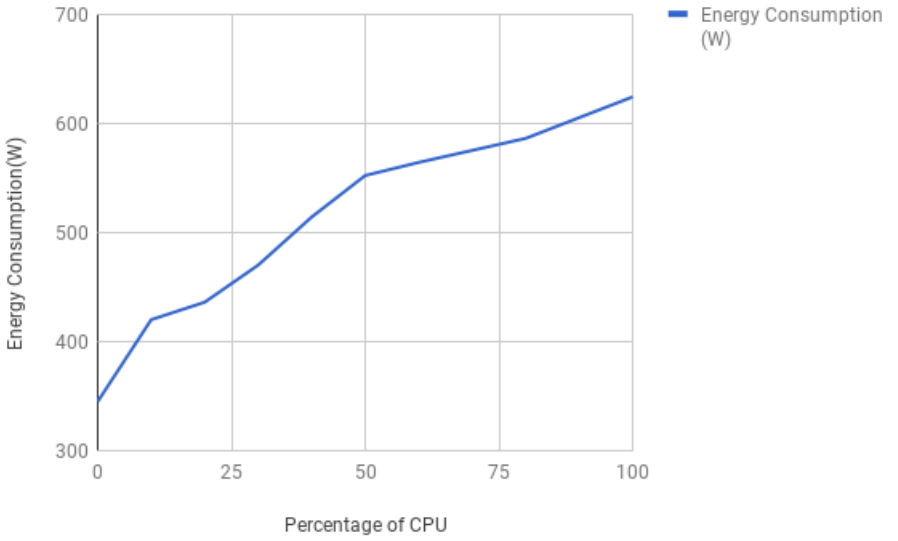

Fig. 1. Energy Consumption by CPU

Two method are chosen to make this test in memory, the first is *memrate N:* start N workers that exercise a buffer with 64, 32, 16 and 8 bit reads and writes. This memory stressor allows one to also specify the maximum read and write rates. The stressors will run at maximum speed if no read or write rates are specified. And the second is *memrate-bytes:* specify the size of the memory buffer being exercised. The default size is 256MB. One can specify the size in units of Bytes, KBytes, MBytes and GBytes using the suffix b, k, m or g. The size limit of memrate-bytes is 4GB, so, for our tests we divide the memory to reach the percentage that we need to test.

The table 1 show the energy consumption by the use of memory, for that we had to use a minimun of CPU that allocate and makes writes/reads, each core can lead with 4GB of memory. (i.e, 50% of CPU means 32 cores, that each one will manage 4GB of memory doing reads and writes, totalizing 128GB of memory). The case where CPU is on 0% and memory does not have nothing means that the machine is idle.

The figure 2 shows the energy consumption on table 1 minus the energy consumption in figure 1, so that removing the others server components it remains only the memory energy consumption.

Table 1. Table of Memory Energy Consumption

CPU(%)	Memory(%)	EnergyConsumption(W)
0		345
10	20	444
20	40	544
30	60	642
40	80	700
50	100	780

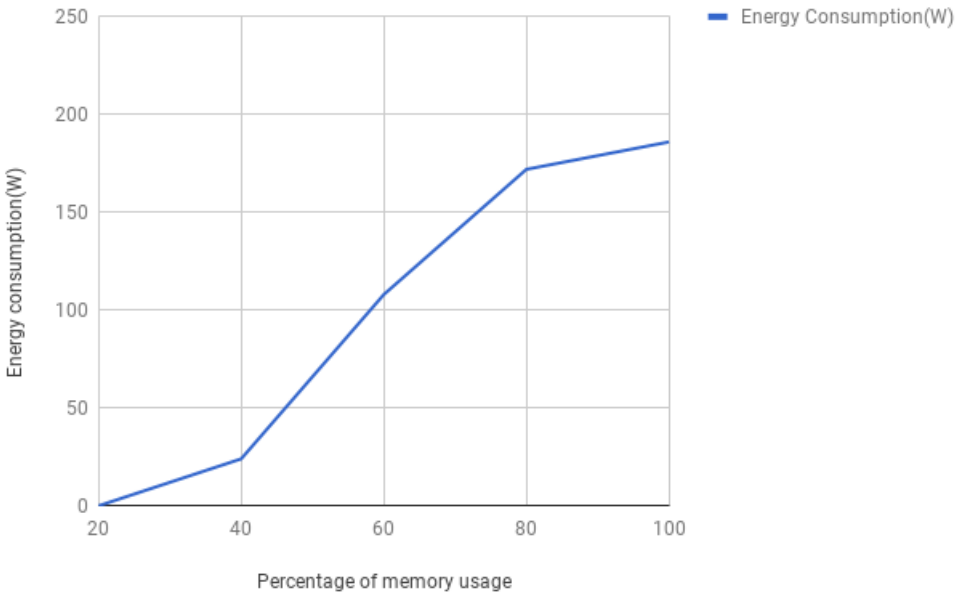

Fig. 2. Energy Consumption by Memory

5.3 Results

We generate a Fuzzy analysis about the results with two inputs and one output. The first input is the CPU usage, the functions were defined by the results of our energy consumption test, always trying to don't have performance loss [15]. The other input is the Memory usage, we define these functions analysing the energy consumption percent of memory per Watt, trying to allocate VMs in machines that have a bad memory utilization considering the energy consumption and always looking for don't have performance loss. The output is the priority that the machine have to receive VMs considering the CPU and Memory usage.

Figure 3 show the use of CPU is adapted to a standard scale, and the linguistic terms for fuzzy systems used are: “underuse” ,“normal” and “overloaded”. Being $P = c$ and $c \in [0; 10]$.

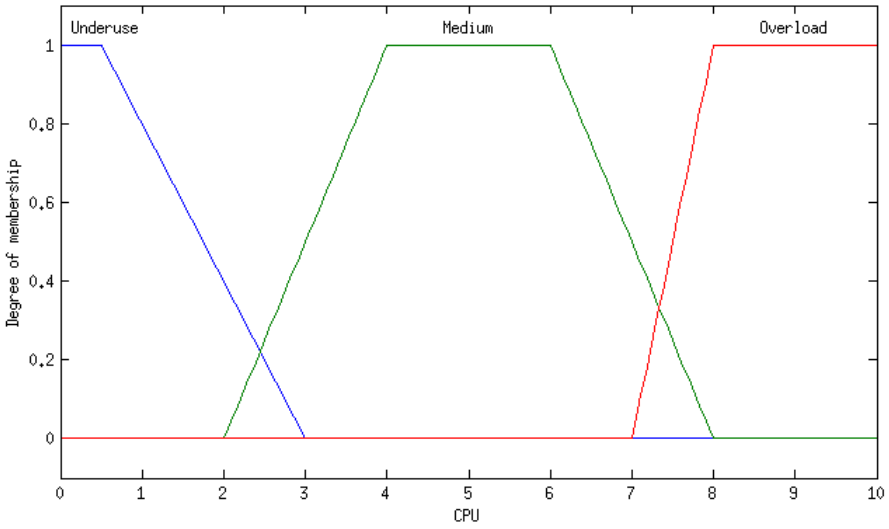

Fig. 3. CPU in the default scale

The figure 4 show the use of Memory is also adapted to a standard scale, and the linguistic terms for fuzzy systems used are: “underuse” ,“normal” and “overloaded”. Being $P = c$ and $c \in [0; 10]$.

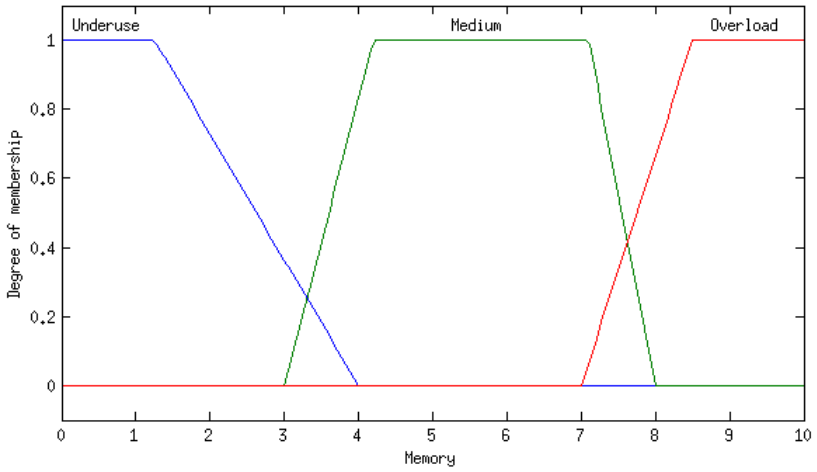

Fig. 4. Memory in the default scale

For the output, the results lead us to the figure 5, which show the degree of membership about a machine and if it can receive a VM in a node. This Fuzzy system is also adapted to a standard scale, and the linguistic terms for fuzzy systems used are: "Low" ,"normal" and "High". Being $P = c$ and $c \in [0; 10]$. With this, we can classify the use of a machine based on its energy consumption, if they need to be shutdown because it is in underuse or if they need to be scheduled to another node because they don't have resources anymore.

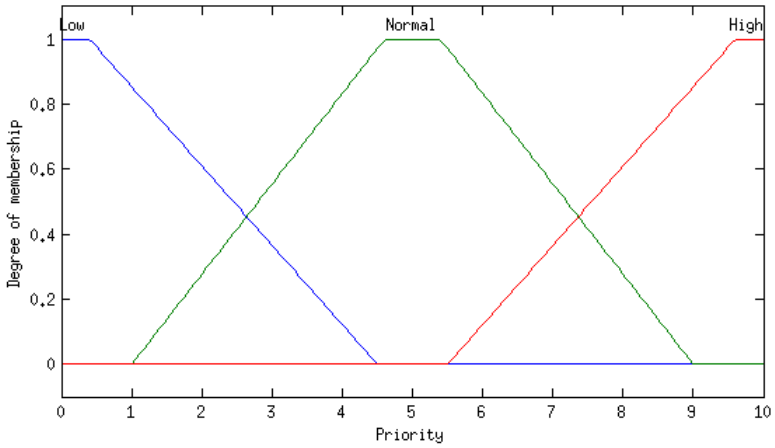

Fig. 5. Migration Priority Membership Function

In order to define the machine use, we did nine rules to help in this decision, figure 6 show they.

1. If (CPU is Underuse) and (Memory is Underuse) then (Priority is High) (1)
2. If (CPU is Underuse) and (Memory is Medium) then (Priority is Normal) (1)
3. If (CPU is Underuse) and (Memory is Overload) then (Priority is Low) (1)
4. If (CPU is Medium) and (Memory is Underuse) then (Priority is High) (1)
5. If (CPU is Medium) and (Memory is Medium) then (Priority is Normal) (1)
6. If (CPU is Medium) and (Memory is Overload) then (Priority is Low) (1)
7. If (CPU is Overload) and (Memory is Underuse) then (Priority is Normal) (1)
8. If (CPU is Overload) and (Memory is Medium) then (Priority is Low) (1)
9. If (CPU is Overload) and (Memory is Overload) then (Priority is Low) (1)

Fig. 6. Fuzzy rules

6 Conclusion

The good management of resources in a cloud can save energy that is wasted with idle machines that could be poweroff and avoiding loss performance by scheduling machines overloaded by VMs.

To help this management, the classification of a machine by your energy consumption can save resources put machines which are underused in poweroff and do load balance in machines which are overload. Besides that, this kind of analysis also covers non-volatile memory that are considered the next step on memory technology [16].

For the future works, we will propose a extension to type-2 Fuzzy Logic that treats the uncertainties associated with fuzzy sets which is not contemplated in fuzzy logic type 1.

References

1. Mell, P.M., Grance, T.: Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, United States (2011)
2. Qiu, M., Chen, Z., Liu, M.: Low-power low-latency data allocation for hybrid scratch-pad memory. IEEE Embedded Systems Letters **6**(4) (Dec 2014) 69–72
3. Fan, X., Weber, W.D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. In: ACM SIGARCH Computer Architecture News. Volume 35., ACM (2007) 13–23
4. Dong, X., Xie, Y.: Adams: Adaptive mlc/slc phase-change memory design for file storage. In: Proceedings of the 16th Asia and South Pacific Design Automation Conference, IEEE Press (2011) 31–36
5. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems: theory and design. IEEE Transactions on Fuzzy systems **8**(5) (2000) 535–550
6. Alvarez, C., Corbal, J., Valero, M.: Fuzzy memoization for floating-point multimedia applications. IEEE Transactions on Computers **54**(7) (July 2005) 922–927

7. Venkatachalam, V., Franz, M.: Power reduction techniques for microprocessor systems. ACM Computing Surveys (CSUR) **37**(3) (2005) 195–237
8. Vogelsang, T.: Understanding the energy consumption of dynamic random access memories. In: Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on, IEEE (2010) 363–374
9. Beloglazov, A.: Energy-efficient management of virtual machines in data centers for cloud computing. PhD thesis (2013)
10. Nathuji, R., Isci, C., Gorbatov, E.: Exploiting platform heterogeneity for power efficient data centers. In: Fourth International Conference on Autonomic Computing (ICAC'07). (June 2007) 5–5
11. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. Computer **40**(12) (Dec 2007) 33–37
12. Awada, U., Li, K., Shen, Y.: Energy consumption in cloud computing data centers. International Journal of Cloud Computing and services science **3**(3) (2014) 145
13. : Canonica. stress-ng. `:http://manpages.ubuntu.com/manpages/wily/man1/stressng.1.html.` (2011) Acessed: 03-05-2018.
14. : GNU freeipmi. `:https://www.gnu.org/software/freeipmi` (2011) Acessed: 03-05-2018.
15. Ismail, H.A., Riasetiawan, M.: Cpu and memory performance analysis on dynamic and dedicated resource allocation using xenserver in data center environment. In: 2016 2nd International Conference on Science and Technology-Computer (ICST). (Oct 2016) 17–22
16. Mittal, S., Vetter, J.S., Li, D.: A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches. IEEE Transactions on Parallel and Distributed Systems **26**(6) (2015) 1524–1537