An Experimental Methodology to Evaluate Energy Efficiency and Performance in an Enterprise Virtualized Environment

Jesus Omana Iglesias
School of Computer Science
and Informatics
University College Dublin
Dublin, Ireland
10289844@ucdconnet.ie

Philip Perry
School of Computer Science
and Informatics
University College Dublin
Dublin, Ireland
philip.perry@ucd.ie

Liam Murphy
School of Computer Science
and Informatics
University College Dublin
Dublin, Ireland
liam.murphy@ucd.ie

Teodora Sandra Buda
School of Computer Science
and Informatics
University College Dublin
Dublin, Ireland
teodora.buda@ucdconnect.ie

James Thorburn
IBM
Software Group
Toronto, Canada
jthorbur@ca.ibm.com

ABSTRACT

Computing servers generally have a narrow dynamic power range. For instance, even completely idle servers consume between 50% and 70% of their peak power. Since the usage rate of the server has the main influence on its power consumption, energy-efficiency is achieved whenever the utilization of the servers that are powered on reaches its peak. For this purpose, enterprises generally adopt the following technique: consolidate as many workloads as possible via virtualization in a minimum amount of servers (i.e. maximize utilization) and power down the ones that remain idle (i.e. reduce power consumption). However, such approach can severely impact servers' performance and reliability.

In this paper, we propose a methodology to determine the ideal values for power consumption and utilization for a server without performance degradation. We accomplish this through a series of experiments using two typical types of workloads commonly found in enterprises: TPC-H and SPECpower_ssj2008 benchmarks. We use the first to measure the amount of queries responded successfully per hour for different numbers of users (i.e. *Throughput@Size*) in the VM. Moreover, we use the latter to measure the power consumption and number of operations successfully handled by a VM at different target loads. We conducted experiments varying the utilization level and number of users for different VMs and the results show that it is possible to reach the maximum value of power consumption for a server, without experiencing performance degradations when running individual, or mixing workloads.

ICPE'14, March 22–26, 2014, Dublin, Ireland.

ACM 978-1-4503-2733-6/14/03 ...$15.00.
http://dx.doi.org/10.1145/2568088.2568099.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

1. INTRODUCTION

Nowadays large investments have been made to build data centers (i.e. purpose-built facilities composed of thousands of servers, providing storage and computing service within and across organizational boundaries). In our previous work, we show that one of the main contributors to the overall cost of running a server is the energy-related cost [23]. Nevertheless, it is often the case that enterprises have idle servers operating at very low levels of utilization, whilst consuming between 50 and 70% of their peak power [10]. Moreover, data collected from more than 5,000 production servers over a six-month period showed that servers operate only at 10-50% of their full capacity most of the time, leading to expenses on over-provisioning, and thus extra Total Cost of Acquisition (TCA) [9]. With the increase of energy costs, the research community has been trying to solve the energy efficiency problem from two different perspectives: (i) creating servers that consume energy proportional to their utilization level, and (ii) maximizing the utilization of the minimum number of servers. In this work, we solely address the impact of the second perspective.

In order to maximize utilization, enterprises generally consolidate workloads via virtualization. Virtualization is a technique that enables applications to share the same physical server by creating multiple virtual machines in such a manner that each application can assume the ownership of the virtual machine (VM) [14]. One of the benefits of virtualization is the potentially more efficient utilization of the server's hardware resources. This can be accomplished when VMs with complementary workload patterns achieve the maximum utilization level that a server can handle, or when the workloads in each VM have complementary resource usage patterns. Furthermore, our work is focused on virtualized environments that run the type of workloads that are typically used in enterprises. We define in detail the performance degradation for a VM depending on the type of workload in section 4.3.

However, high levels of utilization tend to be associated to a higher performance degradation. For instance, it is expected that an application with average utilization approaching 100% is likely to have difficulty meeting its performance requirements (i.e. throughput and latency service-level-agreements). Due to this fact, enterprises will not risk the performance of their VMs in order to maximize the utilization of their servers, thus leading to low levels of utilization. Furthermore, to the best of our knowledge this is the first attempt of analyzing empirically whether it is possible to reach a balance between performance and energy efficiency for typical enterprise workloads. Moreover, previous work [13, 21], assume that when VMs with complementary resource usage patterns reach the maximum utilization level, there will be no performance degradation. However, through our experiments, we show conclusive results that this hypothesis is not always true. In addition, previous work did not study whether the state of a VM influences the performance degradation. For instance, we show that idle VMs do not affect performance regardless of how intensively the active VMs are used. We present this in detail in section 4.4.

The main goal of this paper is to present a methodology for achieving energy-efficiency in a server with no performance degradation per VM. For this purpose, we experimentally investigate the relationship between performance degradation and energy-efficiency for two typical workloads used in enterprises: SPECpower_ssj2008 [4] and TPC-H [8] benchmarks. We use SPECpower_ssj2008 for exploring the number of operations successfully handled by a VM and the power consumption associated at several levels of utilization. We further refer to SPECpower_ssj2008 as SPECpower in the remainder of this paper. We use TPC-H for evaluating the response time for business-oriented ad-hoc queries and concurrent data modifications. In particular, we use the metric *Throughput@Size* from TPC-H which measures the amount of queries that can be executed per hour for different amounts of users.

We explore different scenarios in which a server increases its utilization and dependability and especially how that relates to performance degradation. In order to vary the scenarios we introduced the concept of states of a VM (e.g, idle, constant) to aids the detection of performance degradations. We experimentally studied the relationship between throughput (i.e, queries per hour or number of operations) and performance degradation. Furthermore, we also investigated the relationship between the type of workload (i.e., TPC-H or SPECpower) and performance degradation. Finally, we examined the consequences in performance and energy-efficiency of mixing these two types of workloads. It is important to mention that this experimental methodology can be adapted to a concrete implementation, and provide an automated technique to find a balance between power consumption and performance.

This paper is structured as follows: Section 2 presents the two benchmarks used in this work, namely SPECpower and TPC-H. Section 3 presents a set of relevant questions that need to be addressed in order to find a balance between energy-efficiency and performance. In section 4 we address these questions with the use of a set of scenarios. Section 5 provides an overview of the related work. Finally, section 6 concludes the paper and presents future work.

2. BACKGROUND

In this section, we present the main features of the two benchmarks used in this work, namely, an overview of their design and functionality in order to facilitate the understanding of the questions addressed in this work and their results.

2.1 Overview of SPECpower benchmark

The SPECpower benchmark was designed to produce consistent and repeatable performance and power measurements. The purpose of the benchmark is to imitate a server-side Java transaction processing application. SPECpower strains the CPU, caches, and memory hierarchy, as well as the implementations of the Java virtual machine (JVM), just-in-time (JIT) compiler, and garbage collection. The benchmark is based on the SPECjbb2005 benchmark [4]. It has strict rules for compliance in case the user wants to upload the results into their website. For example, it is necessary to use two systems, namely a system under test (SUT) and a system for control and collection (CCS) [5]. However, due to the amount of experiments that we needed to perform we set the SUT and CCS in the same VM. Furthermore, we first perform the experiments using a laptop as a CCS and did not find any significant difference in the results.

SPECpower executes different types of transactions such as: new order, payment, order status, delivery, stock level, and customer report. The input for each transaction is randomly generated and it modifies in-memory data structures such as warehouses and customers. Transactions are grouped together in batches for scheduling purposes. The delay between batches is calculated to achieve the desired throughput for each target load. In essence, SPECpower runs the workload at different load-levels and reports the power and performance at each load-level [6].

The benchmark starts with a calibration phase, which determines the maximum throughput. The calibrated throughput is set as the throughput target for 100% load-level. The throughput target for the rest of the load-levels is calculated as a percentage of the throughput target for 100% load-level. For example, if the calibrated throughput is 200,000 server-side Java operations per second (i.e. ssj_ops), the 50% target load would have a target throughput of 100,000 ssj_ops. With 10 warehouses, each warehouse needs to sustain a throughput of 10,000 ssj_ops, since there are 2,000 transactions per batch, each warehouse will execute an average of 5 batches per second. Thus, the mean delay between batches is 200 ms. The downside of this feature is that in some cases the benchmark may have an error between the target and the actual load, which may be easily mistaken as performance degradation. For instance, in figure 4(a) there is a ± 1% of error margin due to the way the ssj_ops are batched.

The benchmark supports a set of configurable parameters. For example, the maximum target throughput can be manually configured, as well as the sequence of load-levels, the time of each load, the number of calibrations, etc. The reader can refer to [7] for more information on which parameters are configurable. The flexibility, coupled with the consistency and repeatability of SPECpower, allow us to investigate a balance between energy-efficiency and performance for enterprise-class server workloads.

2.2 Overview of TPC-H benchmark

The TPC-H benchmark is a decision support benchmark comprising a suite of business oriented ad-hoc queries and data modifications [8]. The term decision support implies that managers and executives would need to retrieve data from the database in order to draw a pattern of the company financial results and facilitate their decision making process. TPC-H creates the data for a relational database comprised of eight tables that stores typical product supply information. The benchmark also involves ad-hoc workload that aims to produce unpredictable queries [17].

The workload of the benchmark consists of 22 queries and 2 update procedures, all representing frequent decision making questions. Furthermore, the 22 queries have a high-level of complexity and give answers to real-world business questions. The queries include a rich scope of operators and selectivity constraints, access a large percentage of the populated data and tables and generate intensive disk and CPU activity on the part of the database server. The update procedures are called refresh functions (RFs) [8]. The refreshing functions are not included in the benchmark and they were created by us following the TPC-H guidelines. For instance, RF_1 adds new sales information to the database, the insertion takes place in the two most populated tables and represents 0.1% of the initial population of these two tables. Moreover, RF_2 removes old sales information from the database.

We use DBGEN, which is a data generator provided in the TPC-H package to create the data to be inserted and deleted by the refreshing functions.

Setting the Environment for TPC-H.

In order to perform any test with TPC-H it is necessary to: (i) create the database with the exact schema proposed by the Transaction Performance Council (TPC); (ii) add the constraints; (iii) generate the raw data files using the DBGEN tool; (iv) load the data into the database tables; (v) generate the workload queries to be executed using the TPC-H query generation tool (QGEN); and (vi) develop and install the necessary stored procedures, i.e, RF_1 and RF_2. After all these steps are finished, we are able to run the workload and measure the execution times.

Throughput Test.

The purpose of the throughput test is to measure the ability of the system to process the most queries in the least amount of time [17, 8]. In other words, this test is used to demonstrate the performance of the system against a multi-user workload. Each user is represented by a stream (S), which runs 21 queries in a random order. The stream executes queries in a serial manner but the streams themselves are executed in parallel. We did not consider one of the queries because its execution time was too long (i.e. more than 1 hour). The throughput test must be executed in parallel with a single refresh stream session [8].

The *Throughput@Size* metric is defined as:

$$Throughput@Size = \frac{n \times 21}{T_s} \times 3600 \times SF$$

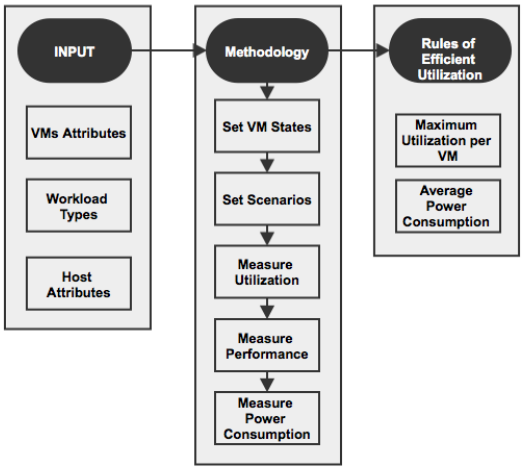


Figure 1: Overview of the proposed experimental methodology

Where n, is the total amount of users on the test. T_s is the time (in seconds) passed until all the streams executed their 21 queries. SF is the size of the database in GB. For this work $SF = 1$.

3. BALANCING ENERGY-EFFICIENCY AND PERFORMANCE

The main objective of this experimental methodology is to study the feasibility of running a server at high levels of utilization without performance degradation. In particular, we investigate the maximum value for energy consumption without any performance degradation in order to maximize the utilization of the server and reach energy-efficiency. Moreover, we study the effects of other scenarios (e.g. idle VMs) on both performance and energy consumption.

The diagram in Figure 1 presents an overview of the inputs, the methodology and finally the expected outputs. The inputs of our experimental methodology are a set of VM attributes (e.g. Number of VMs, Number of VCPU, RAM and Disk), workload types (e.g. SPECpower or TPC-H) and the characteristics of the host (e.g. CPU Cores, RAM, and Disk). After receiving the inputs, our methodology measures the performance, levels of utilization and power consumption in several scenarios using predefined VM states. The output is a set of rules of thumbs that will explicitly define the maximum level of utilization per VM in order to not experience performance degradation, together with the power consumption achieved at such levels of utilization.

3.1 Proposed Study

In this subsection, we categorize this analysis under three important relationships between performance and energy-efficiency:

1. Relationship between power consumption and performance, when there are idle VMs in a server and while varying the levels of utilization of the other VMs. We analyze this relationship on the grounds that generally users are not willing to wait the time required for a VM to boot up and demand having their VM always ready to receive workload (section 3.3).

2. Relationship between throughput, performance degradation, and energy-efficiency when reaching maximum levels of utilization. We study this relationship in order to discover whether performance degradation occurs just after the server attains certain levels of utilization, or on the contrary, VMs can have performance degradations even if the server is not running at particularly high levels of utilization (section 3.4).
3. Relationship between workload mixes, performance degradation and energy-efficiency when reaching maximum levels of utilization. This relationship explores one of the main benefits of virtualization, namely to have different VMs in the same server running workloads that consume different hardware resources (e.g, CPU, RAM, Disk) in order to maximize the utilization of the server. Thus, in this analysis we use benchmarks that simulate real applications that are typically used in enterprises, instead of considering benchmarks that use a single resource. It is important to mention that in the majority of the real-case scenarios, workloads consume more than one type of hardware resource at the same time (section 3.5).

Each of these relationships explores a set of questions that will be answered through the testing of a set of inquiries (INQ). Moreover, it is necessary to evaluate each benchmark separately since their performance metrics are different from each other. Before presenting the inquiries in detail, we first introduce the concept of VM states.

3.2 VM States

We define the state of a VM v as a pattern of utilization aimed to discover performance degradation symptoms. We denote by $U(v)$ the utilization levels of v during a single run and we define it as: $U(v) = \langle u_1, u_2, \dots, u_n \rangle$, where u_i represents the level of utilization of v at step i, $u_i \in \{0, 10, 20, \dots, 100\} \wedge i \in [1, n]$. Below we present the VM states considered in function of their utilization pattern:

- **Idle**: The VM runs at 0% of utilization at all steps: $\forall i \in [1, n], u_i = 0$.
- **Constant**: The VM runs SPECpower at a constant percentage of utilization at all steps: $\forall i \in [1, n], u_i = ct$, where $ct \in \{10, 20, \dots 100\}$.
- **Active**: The VM runs SPECpower at different percentages of utilization during a single run and respects the following condition: $\forall i \in [1, n], u_i \in \{10, 20, \dots 100\} \wedge |u_{i+1} - u_i| \leq 10$.
- N **Users**: VM running the *Throughput@Size* test with N users, where $N \in [2, 7]$.

3.3 Presence of Idle VMs

In this subsection we present two inquiries that intend to explain the effects, if any, of idle VMs on performance and power consumption. Moreover, the *baseline* for the TPC-H performance degradation was established after averaging the results from six identical experiments in which there were no *Idle* VMs.

INQ 1. *Is there an increment in the power consumption of the server due to the presence of Idle VMs? If so, is it constant regardless of the percentage of utilization in the Active VM?*

We plan to validate this relationships by running VMs at different target loads in the case of SPECpower and with different number of users in the case of TPC-H.

Regarding SPECpower we answer this inquiry by comparing the average power consumption of running one and two *Active* VMs from 10% to 100% of its maximum target load against running the same experiment but with one and two additional *Idle* VMs for the case that there is one *Active* VM, and with one *Idle* VM in the case that there are two *Active* VMs. We decided to use *Active* VMs because this will allow us to perform measurements in a very progressive and controlled environment.

With respect to TPC-H, we analyze the effects of simultaneously running VMs with different amount of users, we stress the server by running a minimum of four users (between two VMs) and a maximum of 21 users (between three VMs). We investigate if varying the amount of users when having one or two *Idle* VMs, besides the ones running in an $N\ Users$ state, will have any repercussion in power consumption.

INQ 2. *Is there a performance degradation due to Idle VMs in a server? If so, is it constant regardless of the percentage of utilization in the Active VM?*

In order to solve this question we calculated the performance values while running the same set of experiments from INQ 1 and compared them against the *baseline*. The state of the VMs as well as the number of users for solving this question is identical as the one described in INQ 1.

3.4 Finding the maximum number of operations

We intend to empirically find the maximum number of operations that a server can handle and to test if the VMs will not experience performance degradation as long as this maximum number of operations is not reached. Below we present three inquiries that explain the effects in performance and energy-efficiency whenever the sum of utilization of the VMs gets closer to the maximum number of operations that a server can handle.

INQ 3. *Is it possible to find the maximum number of operations that a server can perform regardless of the number of VMs hosted in that server without performance degradation?*

The main goal of this inquiry is to establish a rule of thumb for maximum levels of utilization without performance degradation for both benchmarks (i.e. SPECpower and TPC-H).

In the case of SPECpower we first set two *Active* VMs with targets of utilization between 60 and 100%. Afterwards, when having three *Active* VMs we set the targets between 40 and 70%. This range of values was selected after noticing in INQ 2 that it is likely to find performance degradation between those levels of utilization.

INQ 4. *Does the state of the VM influence the maximum number of operations until the performance degrades?*

The purpose of this investigation is to observe whether depending on the state of the VM, that particular VM will experience performance degradation faster than when the server reaches its maximum levels of utilization. In the case that this is true the rule of thumb will be modified accordingly. For SPECpower, we first set a VM to a high (between

80 and 100% of utilization) *Constant* state, and then we set another VM to an *Active* state, and study if the VM that is running at a high *Constant* state experiences degradation in performance faster than the other. In another scenario, we set two VMs at a relatively low *Active* state (between 10 and 60%) and a third VM in an *Constant* state, to corroborate if the performance degradation is due to high levels of utilization or to constant levels of utilization.

INQ 5. *Is there a relationship between the results of INQ 3, INQ 4 and energy-efficiency?*

This inquiry aims to discover whether the average power consumption values from the maximum levels of utilization presented in the previous inquiries are equal to the maximum power consumption of the server. In particular, we want to explore whether it is possible to achieve the maximum energy-efficiency with no performance degradation.

3.5 Mixing different types of workloads

Following we present a set of experiments in which we combine SPECpower and TPC-H at different VM states and with different amounts of users in order to corroborate the effects of mixing different types of workload in the same server. As we already mentioned in the introduction of this work, one of the benefits of virtualization is that we could potentially allocate workloads that use different hardware resources in order to maximize the utilization of the server. It is not often the case, especially for enterprises, to find workloads that use only one particular hardware resource. More often is the case that a type of workload uses more than one hardware resource at the same time, however using more intensively one of the resources. For instance, TPC-H heavily uses CPU and Disk. However, Disk is used more intensively than CPU. SPECpower, as we already explained in section 2, it uses mostly CPU.

We primarily want to evaluate if the maximum number of operations found in the previous inquiries is the same when we substitute an *Idle* VM by a VM running TPC-H and to determine if we achieve better energy-efficiency levels when combining different types of workloads.

INQ 6. *Is it possible to combine different types of workload on the same server in a way that they do not interfere with each other and that there is no performance degradation?*

The main goal of this question is to show the effects on performance when running complementary workloads (i.e. they utilize different hardware resources). We performed several experiments with different number of users (in the case of TPC-H) and different levels of utilization for *Constant* states (in the case of SPECpower) in order to analyze the effect of workload mixes. For this inquiry, we set the experiments for SPECpower to have the same length in duration as TPC-H depending on the number of users. For instance, as $T_2 < T_7$ SPECpower will run for less time when evaluating TPC-H with two users than with seven users. We are mostly interested in finding whether the rule of thumb applies in this scenario.

INQ 7. *Is there a relationship between the results of INQ 6 and energy-efficiency?*

This inquiry presents the increment in the average power consumption values when running different types of workloads simultaneously. We have performed experiments both with two VMs and three VMs that combine the two different types of workload.

4. EVALUATION

In this section, we present the results of the inquiries defined in section 3. Furthermore, we first introduce the environment used for the experiments, along with the experimental setup.

4.1 Environment

We present below a description of the hypervisor, the characteristics of the server under test (SUT), and the characteristics of the VMs.

In order to measure the balance between energy-efficiency and performance it is necessary to establish a reliable virtualized environment. The hardware attributes of the SUT: the CPU processor is an Intel Core i7-2600 CPU at 3.40GHz. The processor has a total of 4 cores and 8 logical cores when hyper-threading is ON, as in our experiments. The SUT has 16GB of memory, a Hard Disk of 500GB, and runs a Linux Kernel version 3.2.0-4-amd64. We employ a HAMEG HM8115-2 power meter [2] for our power measurements.

When running the experiments it is necessary to emphasize that the commands were sent via ssh, meaning that no graphical interface was used in order to avoid unnecessary usage of hardware resources. Moreover, we monitored the server and VM's utilization with collectd [1].

4.1.1 KVM/QEMU

KVM (Kernel Virtual Machine)[3] is a Linux kernel module that allows a user space program to utilize the hardware virtualization features of various processors (e.g. Intel VT or AMD-V). QEMU is a generic and open source machine emulator and virtualizer [3]. KVM uses QEMU for I/O hardware emulation. KVM lets a program like QEMU safely execute guest code directly on the host CPU. This is only possible when the target architecture is supported by the host CPU; (currently is limited to x86-on-x86 virtualization only).

The main responsibilities of the KVM/QEMU package are:

- Set up the VM and I/O devices.
- Execute the operating system guest code via KVM kernel module.
- I/O emulation and live migration. Note that we enabled paravirtualized devices (i.e., virtio) to improve the I/O performance.

4.1.2 Virtual Machine Attributes

Three identical VMs were created and configured. The VM attributes are: CPU - 2 virtual CPUs ($VCPUs$) (out of a theoretical maximum of 8); RAM - 4GB; Disk - 20 GB. All the virtual machines have the same OS, namely, Ubuntu 12.04.2 LTS.

In order to run the benchmarks we installed the following software: MySQL 5.5.32 for the TPC-H benchmark (i.e., each VM has its own database), and Java 1.7.0_25 openJDK for the SPECpower benchmark. Finally, we created a database of 1GB in each of the VMs.

4.2 Experimental Setup

Next we describe the criteria used for our evaluation.

- A fixed target load was defined from averaging the results of 10 SPECpower calibrations.
- The SUT began in an idle state prior to the launch of each experiment and was allowed to return to this idle state for at least 20 minutes between runs. This allows the machine to 'cool down' and return to its idle energy consumption levels.
- An *Active* run consisted of at least seven possible target loads. Target loads do not necessarily have to be unique from each other.
- In the majority of the cases in which we used SPECpower, each target load was given 4 minutes, excluding pre and post measurements. We also established a delay of 10 seconds between loads.
- We collected utilization data from the hardware resources and the power meter every 2 seconds.
- In all our results, we report the average power consumption and hardware resource's utilization values over six runs.
- Before each experiments we force the kernel to drop clean caches, dentries and inodes from memory.

4.3 Metrics

For the purpose of this work, we define performance degradation, based on each workload as follows:

Definition 1 (Performance degradation for SPECpower)**.**

$$PD_S = \frac{ssj\_ops - no\_ops}{ssj\_ops}$$

where ssj_ops represents the number of operations targeted, and no_ops the number of operations actually performed.

Definition 2 (Performance degradation for TPC-H)**.**

$$PD_T = \frac{baseline - Throughput@Size}{baseline}$$

where the baseline represents the average between six Throughput@Size in a single VM with all the other VMs powered off and Throughput@Size represents the result achieved in the new scenario.

Finally, as we plan to determine the energy-efficiency level for a server, we measure the power consumption in Watts and we calculate the average power consumption per number of users (in the case of TPC-H) or per target of utilization (in the case of SPECpower).

4.4 Results

Below we present the results corresponding to the inquiries introduced in section 3.

4.4.1 Effects of having idle VMs

INQ 1: Figure 2 shows the variations in power consumption between a VM running as a single *Active* VM in a server, and an *Active* VM running with one or two VMs in an *Idle* state when using the SPECpower benchmark.

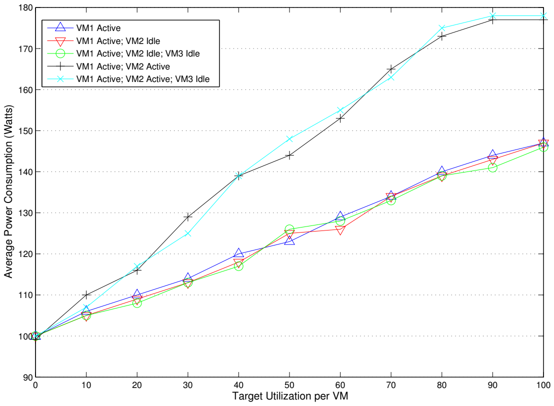


Figure 2: Average power consumption for different VM states (SPECpower)

The figure also presents the relation between two VMs in *Active* state and two *Active* VMs running with a third *Idle* VM. The reader can notice that the difference in the average power consumption at every level of utilization is almost negligible. Thus, the differences between having or not having *Idle* VMs with regards to the average power consumption is 3 Watts maximum, and -1 Watt minimum. In terms of percentage the difference is never more than 1.81%.

In the case of TPC-H, we notice that increasing the amount of users will not increase the average power consumption (see figure 3). However, Table 1 shows that the execution time for two users is significantly lower than for seven users, thus, the total amount of energy consumed by two users is lower than the amount of energy consumed by seven users. Figure 3 also presents the relation between two VMs in *N Users* state and two *N Users* VMs running with a third *Idle* VM. We notice how the average power consumption between the first scenario and the second is of no more than 25 Watts. Here we can also refer to Table 1 and observe that the average execution time for two VMs with two users is much lower than the average execution time for two VMs with seven users, namely, the total average consumption is higher for the former.

Number of Users	2	3	4	5	6	7
T_s one VM	650	980	1329	1766	2004	2386
T_s two VMs	720	1020	1705	1990	2521	2900

Table 1: Execution Time For *Throughput@Size* with different amount of users

It is important to mention that even though in the figures we present the averages, the maximum value over the six runs was not more that 2% higher than such averages, and the minimum value was just 1% lower than the averages presented. In fact, we discovered that the maximum standard deviation for the graphs presented in this paper is 1.8. Furthermore, in the majority of the cases the standard deviation was always below 0.5.

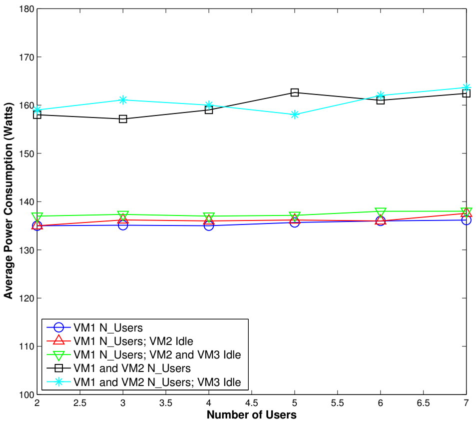


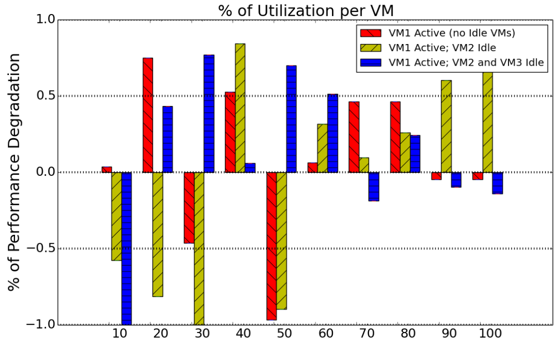
Figure 3: Average power consumption for different number of users (TPC-H)

INQ 2: In order to solve this question we calculated the performance values (i.e. PD_T and PD_S) while running the experiments for the previous inquiry.

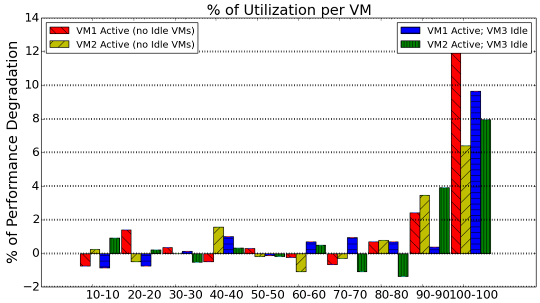
The values for PD_S are presented in figure 4. In figure 4(a) the reader can notice that the performance degradations are almost identical between the experiments in which there were no *Idle* VMs and the ones in which there were one or two *Idle* VMs. Furthermore, it is important to mention that during our experiments we sometimes noticed a PD_S of $\sim \pm 1.5\%$. The reasons for this degradation are that we use two warehouses per VM and we set the maximum number of operations to $107,500$. We assume that these conditions together with the policy that SPECpower follows to dispatch its batches of ssj_ops (refer to section 2.1) are the causes of this acceptable performance degradation.

Figure 4(b) shows that the performance degradation is reached at the same time for both scenarios (i.e. when there was an *Idle* VM and when there was no *Idle* VM). For instance, there is performance degradation when both VMs are running at 90% of utilization. Based on the observe results, we deduce that having *Idle* VMs will neither generate more performance degradation nor will it make either one of the VMs experience performance degradation at lower levels of utilization than when not having *Idle* VMs.

In the case of PD_T, the results are presented in Figure 5. In figure 5(a) we can observe that the difference in performance between the baseline and when having one or two *Idle* VMs is never higher than 1%. Thus, since the best *Throughput@Size* value for any amount of user is of 251.16 queries per hour, this represents a worst case scenario of approximately performing 2 queries less per hour when there are *Idle* VMs. Figure 5(b) presents the results when having two VMs with *N Users*. We can notice that there is a fluctuation in the performance degradation per VM. However, if we sum the degradation for each VM, we find that regardless of the number of users, the results remain almost identical. This concludes that even when the VMs are close to their peak capacity, the effects of *Idle* VMs remains the same, namely a maximum difference between having or not having *Idle* VMs of 1%.



(a) 1 Active VM



(b) 2 Active VMs

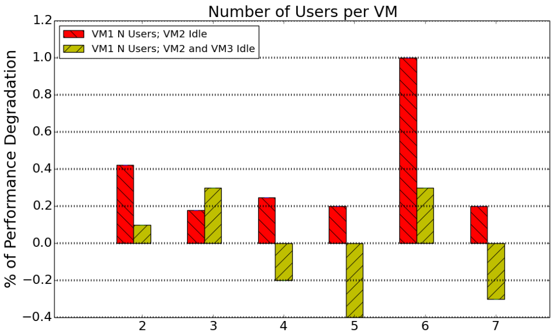
Figure 4: Performance degradation with *Idle* VMs running SPECpower

Similar to the previous inquiry, the minimum and maximum values over the six runs are very close to the average presented.

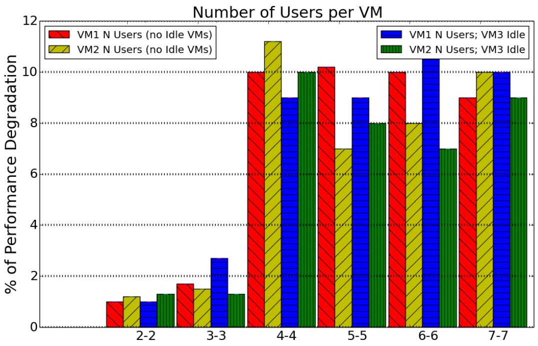
4.4.2 *Effects of increasing the total number of operations*

INQ 3: Figure 6 shows the maximum levels of utilization for two and three VMs. In the figure the reader can recognize that we gradually increased the levels of utilization in each *Active* VM, maintaining each VM almost at the same level of utilization, while measuring the effects of each increment on performance. Figure 6(a) presents for two *Active* VMs the maximum number of operations until we find a performance degradation, which is when both VMs are running at 80% of utilization. This represents $172,000$ ssj_ops. Furthermore, we performed another set of experiments in order to find the maximum number of operations when having three *Active* VMs. Figure 6(b), presents results for three VMs, the result shows that the maximum utilization levels without performance degradation is achieved when the three VMs summed 160% of utilization. This represents exactly $172,000$ ssj_ops, which is the same amount that we found when running two VMs. We conclude that in the case of SPECpower, we can precisely find the maximum number of operations that can be performed without experiencing performance degradation regardless of the number of VMs.

In the case of TPC-H, we have previously found in *INQ 2* that there is performance degradation even in a simple scenario of three VMs running queries for two users (see figure



(a) TPC-H VM1 N Users



(b) TPC-H VM1 and VM2 with N Users

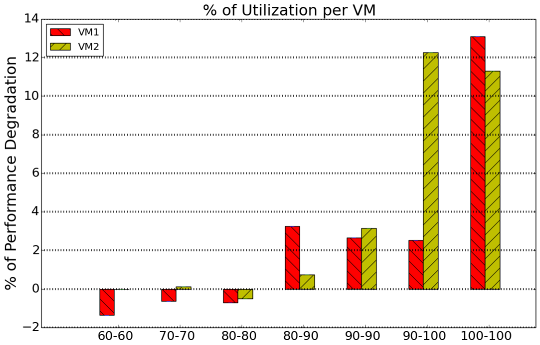
Figure 5: Performance degradation with *Idle* VMs running TPC-H

5(b)). Thus, this inquiry is not applicable to TPC-H, since we observed that it is not possible to not encounter at least a small percentage degradation when running multiple queries for decision support systems. Furthermore, examining the values from the monitoring tool we discovered that the reason is that this type of complex queries fully utilize the VM's available resources (mostly CPU and Disk). Thus, even for a small amount of users the resource utilization is near the maximum capacity, and consequently, increasing the number of users will simply increase the time required to finish the query execution.

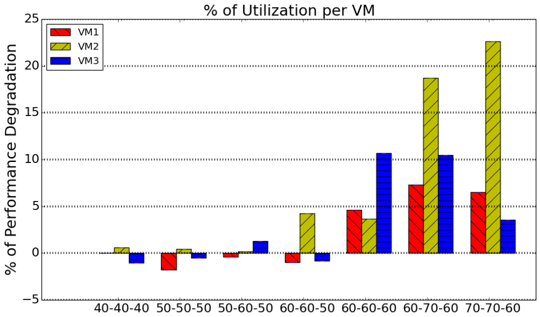
Finally, as with the previous inquiry, the minimum and maximum values over the six runs are very close to the average presented.

We can now partially define the following rule of thumb for no performance degradations in the case of SPECpower:

Rule of Thumb 1. *(preliminary) There are no performance degradations for a virtualized server s in an enterprise environment, when using only SPECpower, when:* $\sum_{i=1}^{m} u_i \leq 160\%$, *where* u_i *represents the level of utilization of VM* v_i *in s and m represents the number of VMs in s.*



(a) Determining the maximum levels of utilization with two *Active* VMs



(b) Determining the maximum levels of utilization with three *Active* VMs

Figure 6: Determining the maximum levels of utilization before experiencing performance degradation

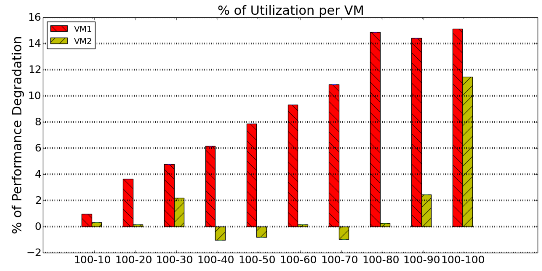
INQ 4: Contrary to what we expected, results show that for the studied environment there is no guarantee of not experiencing performance degradation even if the maximum number of operations is not reached. Figure 7, portraits the performance degradation when one VM is running at high *Constant* levels of utilization. In the figure we can observe that if one of the VMs is demanding most of the available resources (i.e. 100%) then a degradation in performance will occur faster for that particular VM. We performed the same experiments with two VMs in an *Active* state and one VM in a *Constant* state and we found a very similar behavior than with two VMs (see figure 8). Therefore, we determined that if a VM is running at more than 80% of its peak utilization levels, it is likely that the VM will experience performance degradation before the server reaches its maximum capacity.

We can now update and conclude our rule of thumb for no performance degradations in the case of SPECpower:

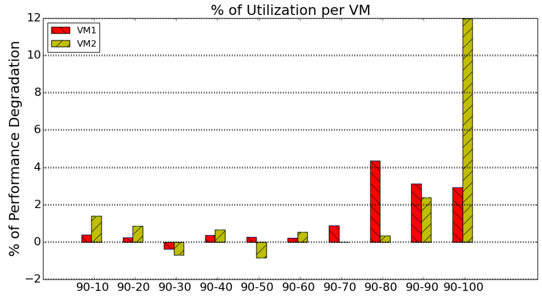
Rule of Thumb 1. *(final) There are no performance degradations for a virtualized server s in an enterprise environment, when using only SPECpower, when:* $\forall \boldsymbol{i} \in [\mathbf{1}, \boldsymbol{m}]\ \boldsymbol{u_i} \leq \mathbf{80\%} \wedge \sum_{i=1}^{m} u_i \leq 160\%$, *where* u_i *represents the level of utilization of VM* v_i *in s and m represents the number of VMs in s.*

We present the average over six runs, but as with the previous inquiries, the minimum and maximum values over the six runs are very close to the average presented.

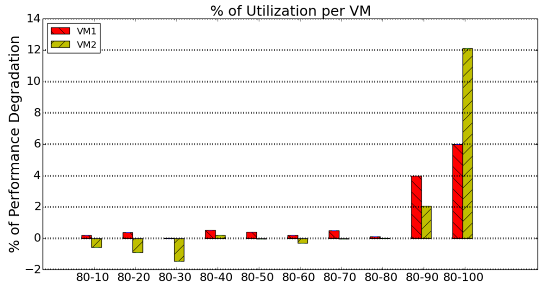
INQ 5: We analyze how the results from Figure 6 (i.e. SPECpower) relate to power consumption and ultimately to energy efficiency. As it can be seen, in figure 9 we marked with a dashed line the last step before the performance



(a) VM1 at *Constant* 100%; VM2 *Active*

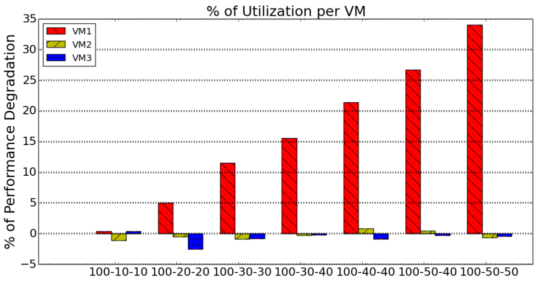


(b) VM1 at *Constant* 90%; VM2 *Active*

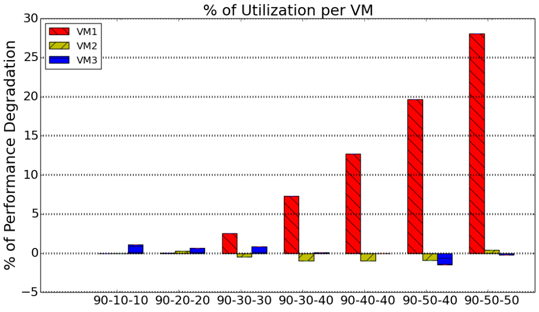


(c) VM1 at *Constant* 80%; VM2 *Active*

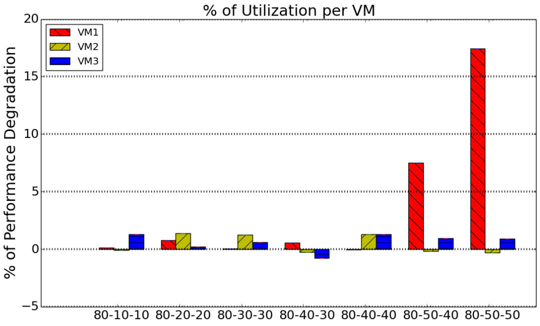
Figure 7: Performance degradation when one VM is running at high *Constant* levels of utilization



(a) VM1 at *Constant* 100%; VM2 and VM3 *Active*



(b) VM1 at *Constant* 90%; VM2 and VM3 *Active*



(c) VM1 at *Constant* 80%; VM2 and VM3 *Active*

Figure 8: Performance degradation when one VM is running at high *Constant* levels of utilization

degradation occurs. The reader can notice that the maximum average power consumption is 180 Watts and that it is possible to reach an average power consumption of 178 Watts, when there are two *Active* VMs, and 179 Watts when there are three *Active* VMs, before experiencing any type of performance degradation. In conclusion, it is possible to arrive near peak power consumption without experiencing any type of performance degradation.

Furthermore, figure 10 presents the average power consumption when running one VM at high *Constant* state (refer to figure 8). We marked with a dashed line the moment in which at least one of the VMs start experiencing performance degradation. We used a blue line in the case that one of the VMs was running at *Constant* 100% of utilization, a red line for the case in which one VM was running at a *Constant* 90% of utilization, and a black line in the case that one of the VMs was running at *Constant* 80%). We can observe that in the first two scenarios, the sever is using on average 160 and 165 Watts when experiencing performance degradation, on the contrary to figure 9 or when a VM is running at a *Constant* 80%, where the server reached at least 176 Watts of average power consumption before experiencing any performance degradation.

As with the previous inquiry, the minimum and maximum values over the six runs are very close to the average presented.

4.4.3 *Effects on performance and power consumption when mixing different types of workloads*

INQ 6: In Table 2, we present the values for the different permutations in the levels of utilization for SPECpower and in the number of users for TPC-H. Evaluating this table we can conclude that in the case of SPECpower there is no performance degradation for the VM running at less than 100% of utilization. However, when the VM is running at 100% of utilization, we observe a performance degradation of approximately 11% in the case of SPECpower and

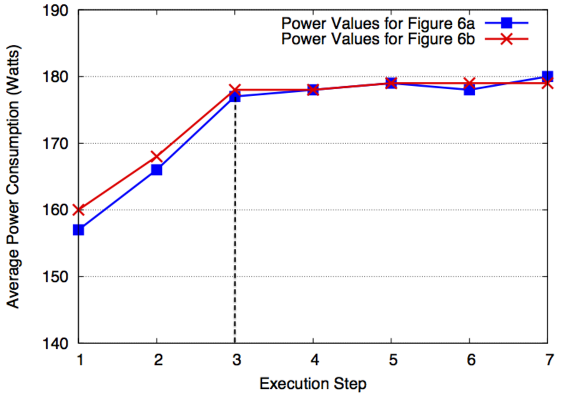


Figure 9: Average power consumption for utilization levels of Figure 6

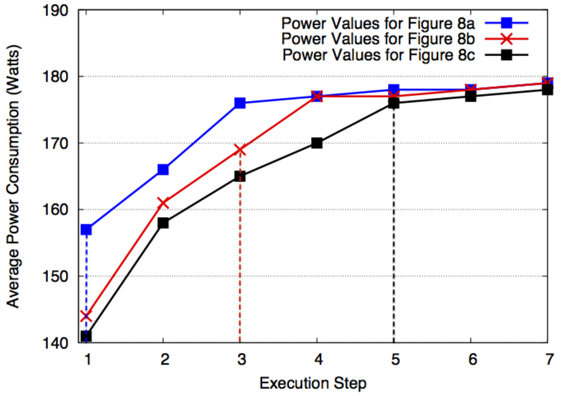


Figure 10: Average power consumption when there is a VM running at a high *Constant* state

between 12 and 13% in the case of TPC-H, depending on the number of users. In the case of TPC-H we found very small performance degradation in almost every case except when the SPECpower was running at 100% of utilization. We observe that the number of users has little or no impact in performance degradation in the case of two VMs with two different types of workloads. In this scenario, the performance degradation are smaller, as generally the VMs are complementary in using mainly different resources (TPC-H-Disk, SPECpower- CPU).

We can now partially formulate our rule of thumb in the case of mixed of workloads:

Rule of Thumb 2. *(preliminary) There are no performance degradations for a virtualized server s in an enterprise environment, when mixing SPECpower and TPC-H, regardless of the number of users when:* $\forall i \in [1, m], v_i$ *running SPECpower,* $u_i \leq 90\%$*, where* u_i *represents the level of utilization of VM* v_i *in s and m represents the number of VMs running SPECpower in s.*

Table 3 shows the effects of mixed workloads using three VMs: two running SPECpower and one running TPC-H. The table shows that in general if there is a degradation in performance in one of the VMs running SPECpower, then there will also be a degradation in the VM running TPC-H.

% of Target Load	PD_S (%)	Number of Users	PD_T (%)	Average Watts
10	0.13	3	0.23	∼149
10	0.27	7	0.24	∼151
80	0.3	3	0.41	∼169
80	0.51	7	0.37	∼172
90	**1.22**	**3**	**0.51**	**∼175**
90	**0.51**	**7**	**0.62**	**∼174**
100	10.7	3	12.43	∼178
100	11.4	7	13.27	∼179
50	0.37	5	0.4	∼162

Table 2: Mixing workloads for two VMs

% of Target Load	PD_S (%)	% of Target Load	PD_S (%)	Num of Users	PD_T (%)	AVG Watts
100	7.3	10	0	3	12.2	∼177
90	3.6	10	0	3	6.7	∼177
80	1.9	20	0	3	3.7	∼176
80	**0**	**10**	**0**	**3**	**0.87**	**∼ 176**
80	**0**	**10**	**0**	**7**	**1.2**	**∼ 176**
70	0	20	0	7	0.33	∼176
60	0	30	0	7	0.32	∼176
50	0	40	0	7	0.41	∼176

Table 3: Mixing workloads for three VMs

We noticed that if the sum of utilization for two VMs with SPECpower is not higher than 90% then there will be no performance degradation regardless of the number of users on the VM running TPC-H. We identified that the threshold in utilization level for each VM running SPECpower is 80%(i.e. lower than 90% as in the previous scenario). Based on the results from these two tables, we observe that mixing different types of workloads brings some benefits, such as running TPC-H without encountering performance degradation. This situation does not occur in an environment running only TPC-H (see *INQ 3*). We observe that in addition to running TPC-H, we can perform 56.25% of the maximum total SPECpower utilization level with no performance degradation in the server (i.e. from 160% to 90%). The reason for this decrement is the fact that both workloads are CPU intensive.

We can now finalize the formulation of our rule of thumb in the case of mixed of workloads:

Rule of Thumb 2. *(final) There are no performance degradations for a virtualized server s in an enterprise environment, when mixing SPECpower and TPC-H, regardless of the number of users when:* $\forall i \in [1, m], v_i$ *running SPECpower,* $u_i \leq \mathbf{80\%} \wedge \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{u_i} \leq \mathbf{90\%}$*, where* u_i *represents the level of utilization of VM* v_i *in s and m represents the number of VMs running SPECpower in s.*

As with the previous inquiries, the minimum and maximum values over the six runs are very close to the average presented.

INQ 7: We analyzed how the results from Tables 2 and 3 relate to power consumption and ultimately to energy efficiency. The reader can observe that in both cases (i.e. when mixing workloads for two and three VMs) the average values for power consumption are very close to the peak value (i.e. between 174 and 176 Watts). The main reasons for the high values in power consumption are: (i) the server reaches very high levels of CPU utilization, and (ii) multiple hardware resources utilized at the same time (i.e. TPC-H uses mainly Disk and CPU, and SPECpower uses mainly CPU).

Finally, as with the previous inquiries, the minimum and maximum values over the six runs are very close to the average presented.

5. RELATED WORK

In this paper we perform a thorough analysis between performance and energy efficiency in a virtualized environment using two enterprise benchmarks (i.e. SPECpower_ssj2008 and TPC-H).

Reviewing the literature we found several techniques for improving one of these two variables. For instance, a general view for improving energy-consumption is to use server consolidation. However, the impact of consolidating several VMs in the same server is not properly studied. Examples of such approaches are: Srikantaiah, et al in [27] discuss consolidating applications or tasks on a lower number of physical machines. However, they do not consider the impact of a virtualized environment. Moreover, there is also a vast literature available about the use of virtual machine placement for server consolidation ([19, 22, 26]). However, such approaches tend to place VMs based on their average resource utilizations, disregarding the effects of peaks utilization in one of the VMs or if such consolidation makes the server reaches energy-efficiency levels.

Other researchers go further and present their approaches to address the problem of achieving energy efficiency and its impact in performance. In this regard, Kephart et al. [18] proposed a coordination strategy based on utility functions for managing power and performance using two separate managers. However, the applications studied are very simple, as they consume just one resource, making them unlikely to be used in a typical enterprise. Furthermore, Gao et al. [15] propose a model for predicting the performance and energy consumption of a server. However, they simplify the power consumption measurements to a linear progression between the 'base' utilization and the maximum CPU utilization, omitting the increments in power consumption by using other hardware resources. Moreover, their work is focus on dynamically resizing the size of the VMs depending on their levels of utilization in order to consolidate them in the minimum amount of servers. Leite et al. [20] developed a coordinated technique for controlling end-to-end performance and minimizing power consumption using Dynamic Voltage Scaling (DVS) in a three-tier web application environment. Moreover, Brihi et al. [11] studied the effect in performance of varying power states in modern computing servers. However, such techniques do not properly address the problem of energy-efficiency, which can only be solved by increasing the utilization of the computing server to its maximum capacity. Furthermore, Rong et al [16] developed an analytical model for investigating energy-performance for parallel workloads. They investigate how to identify an optimal system configuration for running a given parallel workload. The difference is that our work is focused on enterprise type of application running in a virtualized environment.

There have been a number of recent efforts to understand the relationship between performance and power consumption in a virtualized environment. For instance, Smith et al [25] propose a technique for assigning tasks to compute nodes with the aim of balancing the trade-off between energy consumption and the application's performance. However, they use benchmarks that perform very simple tasks (e.g. compress files in memory or read and write a 1024MB test file to and from the hard disk). In addition, they do not study if it is possible to reach energy-efficiency without performance degradation, but rather they aim to spread the workloads in order to consume just the 'base' power consumption. Moreover, Chen et al [12] propose to characterize and profile the energy consumption for different type of tasks and study how the throughput for different type of workloads relates to power consumption. However, our work differs in the sense that we present a measure for performance degradation and how such performance is associated to power consumption and utilization. Finally, with regards to the benchmarks used in this work, Poess et al [24] studied the trends in performance and power consumption for the TPC-H benchmark and Subramaniam et al [28] utilized SPECpower to investigate the feasibility of achieving energy-proportional operations. To our best knowledge, this is the first comprehensive analysis in which there is a set of scenarios that could be found in any enterprise environment. Moreover, we answer a set of inquiries that aim to clarify some limitations of virtualized environments and to discover if performance and energy-efficiency can be achieved simultaneously.

6. CONCLUSION AND FUTURE WORK

The main contribution of this work is a methodology for detecting performance degradations for two typical workloads, namely SPECpower and TPC-H, in an enterprise environment. In order to achieve this, we introduced the concept of VM states based on their utilization levels. The methodology identifies the performance degradations based on predefined scenarios that combine VM states. The scenarios we used include up to three virtual machines, however, the methodology can be adapted for multiple machines by using the same patterns of VM states. Moreover, we evaluate how performance degradation is related to power consumption in a virtualized environment with the objective of reaching energy efficiency. The methodology has been verified on a specific type of server and yielded two rules of thumb to enable the optimum trade off point to be found. It is expected that the methodology could be used to similar effects on other types of servers.

We demonstrated that it is possible to achieve energy-efficiency without any performance degradation in the scenarios studied. Furthermore, we identified the maximum level of utilization with no performance degradation both for a server and a VM and defined accordingly a rule of thumb. Through our analysis we concluded that a server does not experience performance degradation as long as any of the virtual machines in the server under study does not reach high levels of utilization and the servers' utilization doesn't reach its peak capacity. Moreover, we investigated the case of a decision support system. In this case, the system performs complicated queries that require hardware resources for a significant amount of time. Due to this fact, the VMs are constantly required to be at high levels of utilization, leading to performance degradation for the VMs running that type of workload. Finally, we investigated the case of mixed workloads and defined a rule of thumb such that the server reaches high levels of utilization and none of the VMs experience performance degradation.

As future work, we plan to create an automated model that performs the analysis provided in this paper. The model will take as input the workload type, and a server power model, to subsequently investigate the feasibility of achieving energy-efficiency without performance degradation in the particular server. If energy-efficiency is reachable, the model will output the conditions for balancing performance and energy-consumption (e.g. maximum utilization level).

7. ACKNOWLEDGMENTS

We thank Anthony Ventresque and Ulrich Dangel for their valuable support and feedback. This work was supported by the Science Foundation Ireland under grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Center (www.lero.ie), and by Enterprise Ireland Innovation Partnership in cooperation with IBM and University College Dublin under grant IP/2010/0061.

8. REFERENCES

[1] collectd - the system statistics collection daemon. `https://collectd.org`.

[2] HAMEG HM8115-2 power meter. `http://www.hameg.com/0.147.0.html`.

[3] KVM - Kernel Based Virtual Machine. `www.linux-kvm.org`.

[4] SPECpower_ssj2008. `http://www.spec.org/power_ssj2008/`.

[5] SPECpower_ssj2008 - Design Document CCS. `http://www.spec.org/power/docs/SPECpower_ssj2008-Design_ccs.pdf`.

[6] SPECpower_ssj2008 - Design Document SSJ Workload. `http://www.spec.org/power/docs/SPECpower_ssj2008-Design_ssj.pdf`.

[7] SPECpower_ssj2008 - Run and Reporting Rulest. `http://www.spec.org/power/docs/SPECpower_ssj2008-Run_Reporting_Rules.html`.

[8] TPC-H. `http://www.tpc.org/tpch/`.

[9] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.

[10] A. Beloglazov and R. Buyya. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In *8th International Workshop on Middleware for Grids, Clouds and e-Science (MGC)*, pages 4:1–4:6, 2010.

[11] A. Brihi and W. Dargie. Dynamic voltage and frequency scaling in multimedia servers. In *27th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 374–380, 2013.

[12] F. Chen, J. Grundy, Y. Yang, J.-G. Schneider, and Q. He. Experimental analysis of task-based energy consumption in cloud computing systems. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, ICPE '13, pages 295–306. ACM, 2013.

[13] G. Dasgupta, A. Sharma, A. Verma, A. Neogi, and R. Kothari. Workload management for power efficiency in virtualized data centers. *Communications of the ACM*, 54(7):131–141, 2011.

[14] U. Drepper. The cost of virtualization. *Queue*, 6(1):28–35, Jan. 2008.

[15] Y. Gao, Z. Qi, Y. Wu, R. Wang, L. Liu, J. Xu, and H. Guan. A power and performance management framework for virtualized server clusters. In *2011 IEEE/ACM International Conference on Green Computing and Communications (GREENCOM)*, pages 170–175, 2011.

[16] R. Ge, X. Feng, and K. Cameron. Modeling and evaluating energy-performance efficiency of parallel processing on multicore based power aware systems. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8, 2009.

[17] A. G. Janopoulou. Ad-hoc decision-supports. Master's thesis, National Technical University of Athens, 2010.

[18] J. O. Kephart, H. Chan, R. Das, D. W. Levine, G. Tesauro, F. Rawson, and C. Lefurgy. Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs. In *4th International Conference on Autonomic Computing (ICAC)*, pages 24–, 2007.

[19] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. D. Nguyen. Reducing electricity cost through virtual machine placement in high performance computing clouds. In *2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 22:1–22:12, 2011.

[20] J. C. Leite, D. M. Kusic, D. Mossé, and L. Bertini. Stochastic approximation control of power and tardiness in a three-tier web-hosting cluster. In *7th international conference on Autonomic computing (ICAC)*, pages 41–50, 2010.

[21] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis. Efficient resource provisioning in compute clouds via vm multiplexing. In *7th international conference on Autonomic computing (ICAC)*, pages 11–20, 2010.

[22] K. Mills, J. Filliben, and C. Dabrowski. Comparing vm-placement algorithms for on-demand clouds. In *3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 91–98, 2011.

[23] J. Omana Iglesias, P. Perry, N. Stokes, J. Thorburn, and L. Murphy. A cost-capacity analysis for assessing the efficiency of heterogeneous computing assets in an enterprise cloud. In *6th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, page to appear.

[24] M. Poess and R. Othayoth Nambiar. A power consumption analysis of decision support systems. In *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*, WOSP/SIPEW '10, pages 147–152. ACM, 2010.

[25] J. W. Smith and I. Sommerville. Understanding tradeoffs between power usage and performance in a virtualized environment. In *6th International Conference on Cloud Computing (CLOUD)*, pages 725–731, 2013.

[26] B. Speitkamp and M. Bichler. A mathematical programming approach for server consolidation problems in virtualized data centers. *IEEE Transactions on Services Computing*, 3(4):266–278, Oct. 2010.

[27] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing. In *2008 conference on Power aware computing and systems (HotPower)*, pages 10–10, 2008.

[28] B. Subramaniam and W.-c. Feng. Towards energy-proportional computing for enterprise-class server workloads. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, ICPE '13, pages 15–26. ACM, 2013.