# Modeling Service Execution on Data Centers for Energy Efficiency and Quality of Service Monitoring

Monica Vitali
DEIB
Politecnico di Milano
Milan, Italy
vitali@elet.polimi.it

Una-May O'Reilly, Kalyan Veeramachaneni
CSAIL
Massachusetts Institute of Technology
Cambridge (MA), USA
{unamay, kalyan}@csail.mit.edu

***Abstract*—This work provides a system modeling approach to describe a virtualized data center environment running a business process. This model allows the collection of simulation data at different workload rates that can be used as a dataset for reasoning about quality of service and energy efficiency issues related to the process. The model overcomes the issue of obtaining real data from data center administrators and collecting relevant information needed for studying the process behavior. The model is flexible and can be used to model data centers of different dimensions and characteristics. It can also be used for "what-if" analysis about the system configuration, predicting the outcome of a modification over energy efficiency and quality of service.**

***Index Terms*—Model, energy efficiency, quality of service, data center, estimation, business process, virtualization.**

## I. Introduction

Energy efficiency management and energy usage reduction are important topics in the data centers environment. Data centers consume a considerable percentage of the worldwide global energy and, most of the time, available resources are not properly used. In the context of Service Oriented Systems, energy efficiency is usually in conflict with quality of service (QoS). The tradeoff between these two aspects is difficult to manage in an efficient way.

We are ultimately interested in collecting data reports of energy usage and QoS from a real system. This allows us to analyze and adapt its design (e.g. resource allocation, VM-server-activity mapping, etc.) toward a dynamic balance between energy efficiency and adequate QoS. This goal presents multiple challenges. First, a predictive model of the system or some means of predicting the outcomes of a system design change is necessary. Further, online changes will be required and these must be monitored. The collection of real monitoring data for energy and QoS is not trivial. The first issue is the availability of such an observable environment, with a monitoring system installed. Data center owners are also reticent to give access to data even if the customers' data are not of a sensitive nature, and even more reticent to allow the access to their system and its modification. Yet another challenge arises because, with virtualization, a system which "appears" to function in isolation, actually does not. It is important to understand the behavior of the system under study while also incorporating relevant influences of the "neighboring" systems in the virtual environment. In well controlled data center settings, i.e. when virtual resource management is transparent and stable, this task is straightforward. In other scenarios, it is complex.

Given these premises, we propose a system modeling approach which helps addressing the aforementioned challenges by approximating the behavior of a real system. The model allows clear expression, calibration and prediction. It can be used for collecting simulated monitoring data and for testing different workload levels, difficult to test in a real environment. The model of the data center is also flexible. It can be enriched with more components and parametrized depending on the characteristics of the business process (BP) considered. The structure of the system, in terms of resources allocated, number of active servers, positions of the virtual machines, can be modified at any time. The simulation can be used to conduct a what-if analysis to evaluate the outcome of a modification over energy efficiency and QoS in an off-line modality. Even being an approximation of the real system, the proposed approach allows testing of configurations impossible to perform in a real environment, without affecting its performance.

The class of systems we aim to improve (and model) is a coherent set of servers composing in a single data center. The intended purpose of the system class is to support a BP. A BP can be described as a work-flow of activities. It is deployed over these servers using virtualization. In this class of system, each activity is on its own dedicated virtual machine (VM). The work flow has conditional branching which depends on the user interaction but which can be estimated from historical data. The system's monitoring sub-components will periodically sub-sample the system state to enable model calibration and validation as well as support real-time analysis. The number of variables to be monitored and the rate for data collection are dependent from the number of servers and virtual machines involved. A representation of one example of this class of system is shown in Fig. 1. It comprises 2 servers with 2 and 3 business activities, each executing in its own VM.

We proceed as follows: in Sec. II we introduce related works. In Sec. III we describe how the system has been modeled in all its components. In Sec. IV we validate our

978-1-4799-0652-9/13/$31.00 

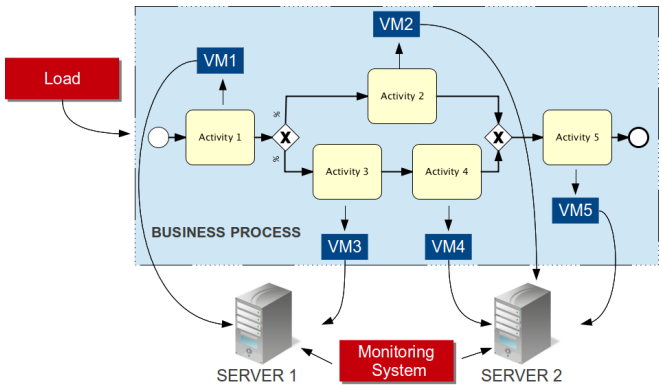



Fig. 1. A system class example

model by simulating an incoming load. Conclusions and future works are discussed in Sec V.

## II. State of the Art

Models are fundamental in a lot of different fields. Models can be used in an offline and online mode to evaluate new configurations and solutions to a problem both at design and run time.

In energy efficiency research, power models are used to improve the design of components and of the system. In [1], the authors provide a model for the CPU power consumption, while in [2] and [3] other components, such as memory, disks, and networks are modeled too, always with a focus on energy and power. In [4], a model for real-time power consumption prediction is provided. A collection of work related to power models is provided in [5].

Energy consumption measurements are hard to collect in a data center environment. The problem of estimating energy at a software level is still open. Some scholars try to estimate the energy consumption of an application from the knowledge of the amount of resources used by the application and the total execution time [6]. Resources considered are CPU, network, and disk. In [6] energy consumption of a laptop battery is considered. This does not extend to desktop or server systems. A similar approach is described in [7] where idle and activity states are distinct when estimating energy consumption.

Other approaches model energy consumption focusing on the virtual machine level. In [8], the account of energy consumption is estimated from performance monitoring counters (PMC) of CPU and memory. The authors define a power model, trained with data collected for different loads of each component. Another approach [9] builds a parametric model to compute energy based on resources usage. In the model only CPU, memory, and disk are considered. The contribution of each component is modeled as a linear function of its usage. The accounting of each VM can be made knowing the amount of each physical resource used by it. This information is available from the hypervisor. The parameters of the linear functions need to be learned using linear regression.

The definition of a model for energy estimation at the application or virtual machine level is still an open issue and all the presented techniques use approximation in defining the energy consumption model.

The cited works are only a small subset of available fields in which models have been proposed to describe the behavior of a system and to allow testing and off-line reasoning on its behavior. The system modeled in this paper focus attention on a different system class, virtualized data centers. Also the level of monitoring is different, incorporating application level information of the BP and its parameters.

TABLE I
Green Performance Indicators and Key Performance Indicators

| Indicator | Description |
|---|---|
| CPU Usage | The amount of CPU used as a fraction of CPU allocated. It is a number between 0 and 1. It can be measured both at the VM and at the server level |
| Response Time | Time between the start and the completion of a specific activity instance in the BP. It is measured in seconds. |
| Energy | The amount of energy used by a component in the system. It is measured in kilowatt-hour |
| Performance per kWh | The number of operations that each activity executes per energy unit, expressed in kilowatt-hour. |

## III. The Model of the Data Center

Energy efficiency and QoS of a process executed over a data center are not directly measurable, but can be assessed using a set of indicators belonging to two families: green and key performance indicators [10][11]. In this work we consider a small subset of these indicators. The list of indicators, with a description, is contained in Tab. I. Indicators values are computed from data collected through a monitoring system.

In this section, we present a system modeling approach to simulate the collection of monitoring data. Some assumptions are made about the system that we are going to model:

- we model a BP in which each activity is deployed on a dedicated virtual machine;
- resources are not over-allocated, the sum of the resources allocated through virtualization is no more than the available physical resources;
- resources that are not used can be momentarily used by other virtual machines if needed.

The complete list of variables monitored and computed in the system is contained in Tab. II.

In the rest of the section the modeling of all the interesting components is analyzed, including CPU utilization (Sec. III-A), response time (Sec. III-B) and performance per kWh (Sec. III-C).

### A. CPU Usage

CPU usage of the virtual machine is strictly dependent on the workload and on the activity executed on the virtual machine. Each activity $k$ has an average usage need $\lambda_k$. Given

TABLE II
VARIABLES IN THE SYSTEM

| Symbols | Description |
|---|---|
| $i : i = 1 \dots I$ | virtual machine $i$ of the system |
| $j : j = 1 \dots J$ | server $j$ of the system |
| $k : k = 1 \dots K$ | activity $k$ of the system |
| $d_k, d_i$ | need for CPU for activity $k$ or virtual machine $i$ |
| $L, l_i$ | incoming load rate for the whole process and for virtual machine $i$ |
| $U_j, U_i$ | CPU usage value for server $j$ or virtual machine $i$ |
| $V_i$ | CPU usage value for virtual machine $i$ expressed in terms of server resources |
| $R_i$ | response time of the activity $k$ running on virtual machine $i$ |
| $PE_i$ | performance per energy unit of the activity $k$ running on virtual machine $i$ |
| $P_j, P_i$ | instant power consumed by the server $j$ or virtual machine $i$ |
| $E_j^T, E_i^T$ | energy consumed by the server $j$ or virtual machine $i$ in the time period $T$ |
| $CPU_j, CPU_i$ | total CPU allocated to the server $j$ or virtual machine $i$ |

this value, in each moment, the resource demand of the activity can be sampled by a distribution. We want to sample from a distribution so that samples are around $\lambda_k$ but with some small deviation, in order to simulate small variations around the average usage rate. We decided to use a Poisson distribution, in which the parameter $\lambda_k$ represents the more probable value. Poisson has been chosen because it samples integer positive values around the more probable one. The demand for activity $k$ can be defined as:

$$d_k \sim Pois(\lambda_k) \tag{1}$$

with $d_k : [0...D_k]$ an absolute value representing the amount of resources needed.

Given the instantaneous load value $l_i$, the estimated resource need of virtual machine $i$ to execute activity $k$ is:

$$d_i = \begin{cases} \frac{d_k}{CPU_i} & IF\ l_i \leq 1 \\ \sum_{l_i} \frac{d_k}{CPU_i} + f(l_i) & IF\ l_i > 1 \end{cases} \tag{2}$$

with $d_i \in [0...D_i]$ a value representing the amount of resources needed expressed as a fraction of the resources on virtual machine $i$. In Eq. 2, the value $f(l_i)$ is an overload due to the management of the request queue when more than 1 request is received. In our tests we will consider this value as 0.

From Eq. 2 we can derive the usage of virtual machine $i$ as a direct consequence of its resource demand as follows:

$$U_i = min \langle d_i, 1 \rangle \tag{3}$$

with $U_i \in [0...1]$ expressed as a ratio of the VM resources.

This model derives the CPU usage so that all the available resources will be used if needed, without exceeding the maximum available which is 1.

Starting from sampled values from the defined distributions for each VM in the example system, it is possible to compute the CPU usage at the server level. In this case, we expect the CPU usage of the server to be proportional to the CPU usage of the virtual machines running on it, plus a constant value $s$ representing the CPU used by the server itself. We also add a noise parameter $\eta$ to the equations. We represent the CPU usage model for the server in normal conditions in Eq. 4:

$$U_j^{min} = min \left\langle 1, \sum_{i \in j} w_i U_i + s + \eta \right\rangle \tag{4}$$

with $U_j^{min} \in [0...1]$ expressed as a ratio of the server resources. The weight $w_i$ represents the ratio of CPU on the server allocated to the Virtual Machine and can be expressed as:

$$w_i = \frac{CPU_i}{CPU_j} \tag{5}$$

The usage of each virtual machine is not isolated and when the demand of one of them is high, it can use more resources than the one directly assigned, by stealing unused CPU cycle if they are available or competing with other virtual machines for them. For this reason, when defining the CPU usage for a server, this can be bigger than the sum of the usage of the single virtual machines running on it. We assume that all the available resources will be allocated if needed and we define the server usage as:

$$U_j = min \left\langle 1, \sum_{i \in j} w_i d_i + s + \eta \right\rangle \tag{6}$$

with $U_j^{min} \in [0...1]$ expressed as a ratio of the server resources.

From Eq. 4 and Eq. 6 it is possible to derive the amount of resources on the server that are reallocated to one or more of the virtual machines running on it as:

$$U_j^* = U_j - U_j^{min} \tag{7}$$

with $U_j^* \in [0...1]$ expressed as a ratio of the server resources. The amount of extra resources used by a virtual machine will be a portion of this value proportional to the unsatisfied demands of all the virtual machines on the server $j$:

$$U_i^* = min \left\langle d_i - U_i, \frac{d_i - U_i}{\sum_{i:d_i > U_i} w_i (d_i - U_i)} * U_j^* \right\rangle \tag{8}$$

with $U_i$ expressed as a ratio of the virtual machine resources.

The effective CPU usage of the virtual machine derives from Eq. 3 and Eq. 8:

$$U_i' = U_i + U_i^* \tag{9}$$

with $U'_i$ expressed as a ratio of the virtual machine resources possibly bigger than 100%.

The resources used on the server j for the virtual machine i can be derived from Eq. 9 as:

$$V_i = w_i U'_i \tag{10}$$

with $V_i$ expressed as a ratio of the server resources.

### B. Response Time

Response time is modeled here as the time needed for executing an instance of an activity. Response time is dependent from the characteristics of the activity, but also from the CPU usage of the machine running the application. When CPU usage is low, the application can take easily all the resources needed and complete in a small amount of time. When the CPU usage is high, the application have to wait for resources, and response time value increases.

As defined in [12], response time can be expressed by the following relation:

$$R = \frac{c * usage}{d - usage} + e \tag{11}$$

with $c$, $d$ and $e$ parameters dependent from the application behavior. This expression is generic and does not fit properly a virtualized environment. If the server in which the virtual machine is deployed is overloaded, the response time will be longer because of the non isolation of the resources shared between virtual machines concurrently running on the same server.

In our model, we want to express the relation between the average response time of an activity running on a virtual machine taking into account that the real usage of the virtual machine can be bigger than one if it is using spare resources on the server. We also want to model the response time so that an exponential growth can be observed when the resources available for the virtual machine are less than the one demanded. Hence, we modify Eq. 11 as follows:

$$R_i = t_k + \frac{t_k 10^{-2} U_i}{[1 - (d_i - U'_i)]^\gamma} \tag{12}$$

with $k$ deployed on virtual machine $i$.

According to this model, defined $t_k$ as the minimum amount of time to complete the activity $k$ on virtual machine $i$, response time will be only slightly increased with the usage of the virtual machine, with an exponential increasing when not enough resources are available.

### C. Performance per kWh

The Performance per kWh metric gives a measurement of the number of operations the activity can complete using an energy unit. The mathematical definition is as follows:

$$PE_i = \frac{N_i^T}{E_i^T} \tag{13}$$

where $N_i^T$ is the number of operations completed in the period of time $T$ and $E_i^T$ is the energy used in the same period.

The numerator of Eq. 13 is strictly dependent on response time. We can consider the total number of operations executed by an instance of the activity as a constant value $N_k$. Knowing the response time of the activity, which is the time to complete all its operations, the number of operations completed in the period $T$ can be expressed as:

$$N_i^T = \frac{N_k}{R_i} * T \tag{14}$$

where the activity $k$ is the one deployed on virtual machine $i$.

To derive the second term of equation 13 we have to define the power consumed by the server. According to [13], power consumed by a server can be computed as the combination of the idle power consumption plus a component dependent on the CPU usage. The idle power is a portion of the power consumed in peak usage condition, expressed as $P_j^{peak}$. From this value we can derive a formula for modeling the server power consumption as follows:

$$P_j = \underbrace{\alpha P_j^{peak}}_{idle} + \underbrace{\beta P_j^{peak} * U_j}_{usage\ dependent} \tag{15}$$

where $\alpha$ and $\beta$ are equal to $0.66$ and $0.34$ according to [13]. Similar values are provided in [14] where authors assign to idle server the 60% of the peak power consumption.

Another model is provided in [12] similar to Eq. 15.

The power consumed by the whole server has to be split for all the virtual machines running on it. If we consider Eq. 15, the only portion of energy that can be ascribed to the virtual machines is the non idle part. Assuming the server is not doing anything except running the virtual machines, power can be split on the basis of CPU utilization.

$$P_i = \beta P_j^{peak} * w_i U'_i \tag{16}$$

where $w_i$ is defined as in Eq. 5 and $U'_i$ is defined as in Eq. 9.

Energy can be computed integrating the power over the time period considered, both for the server:

$$E_j^T = \int_T P_{j,t} \, \mathrm{d}t \tag{17}$$

and for the virtual machine:

$$E_i^T = \int_T P_{i,t} \, \mathrm{d}t = \int_T \beta P_{j,t}^{peak} * w_i U'_{i,t} \, \mathrm{d}t \tag{18}$$

All the parameters of the model are described in Tab. III.

The model defined to this point must next be calibrated to express the physical equipment used, the activities running on the virtual machines, the structure of the data center, and the mapping between activities, virtual machines, and servers.

## IV. Model Validation

Once the model has been defined, it can be validated for relational correctness by simulating an incoming load into the system. Validation is used to check whether the model expresses the key relational properties of the system, reflecting how this class of systems reacts to different load regimes.

TABLE III
PARAMETERS IN THE SYSTEM

| Parameter | Description |
|---|---|
| $s$ | server CPU usage that is independent from the virtual machines deployed on it |
| $\eta$ | noise added to the server usage computation |
| $\gamma$ | parameter for the exponential behavior of the response time |
| $\alpha, \beta$ | parameters for the computation of the power consumption knowing the peak power usage |
| $t_k$ | minimum time for completing activity $k$ |
| $\lambda_k$ | average resource need of activity $k$ |

Samples values for all the metrics of the system can be obtained giving a load value as input of the model. The load represents the number of requests that has to be handled at a given time by the process. Since each activity has a probability to be executed due to the branches probabilities, the general load has to be redistributed according to these values. As an example, in the BP in Fig. 1, Activity1 and Activity5 are always executed, so they will have to handle the 100% of the incoming load, while Activity2, Activity3 and Activity4 will handle only a portion of it depending from the branch execution percentage. This value is expressed as a parameter of the activity.

Given a load value $L$, different activities can require different CPU resources, This feature has been modeled using the activity parameter $\lambda_k$, representing the average CPU usage of activity $k$. Eq. 2, allows us to derive the resource demand $d_i$ of a Virtual Machine as:

$$d_i = \sum_{l_i} \frac{Pois(\lambda_k)}{CPU_i} \tag{19}$$

where $l_i = L * splitProb_k$.

Substituting this value in the formulas described in Sec. III results in the derivation of simulated values of the monitoring system.

We evaluated a model of a system with two servers S1 and S2 and three virtual machines V1, V2 and V3, which is a part of the system shown in Fig. 1. To observe the load impact, an incremental load was used to test the model, starting from 0 and increasing to 20 for the first virtual machine while keeping the other two stable at a constant load of 5, and the same was repeated for each VM. Values obtained for the system are represented in Fig. 2, where the first line contains the plots of all the values monitored on server 1 and the second all the values monitored on server 2. The CPU usage values of different VMs has to be independent by each other, but they influences the CPU usage of the server where they are deployed. This relation is shown in Fig. 2(a) and Fig. 2(d). Even response time values are independent from each other, but they are dependent from the usage of the VM's CPU and response time values increase exponentially when CPU gets saturated (see Fig. 2(b) and Fig. 2(e)). Powers of different

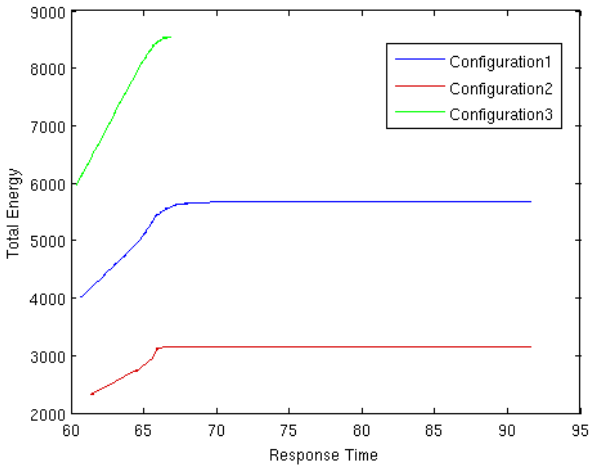



Fig. 3. What-if analysis with three different configuration of the data center

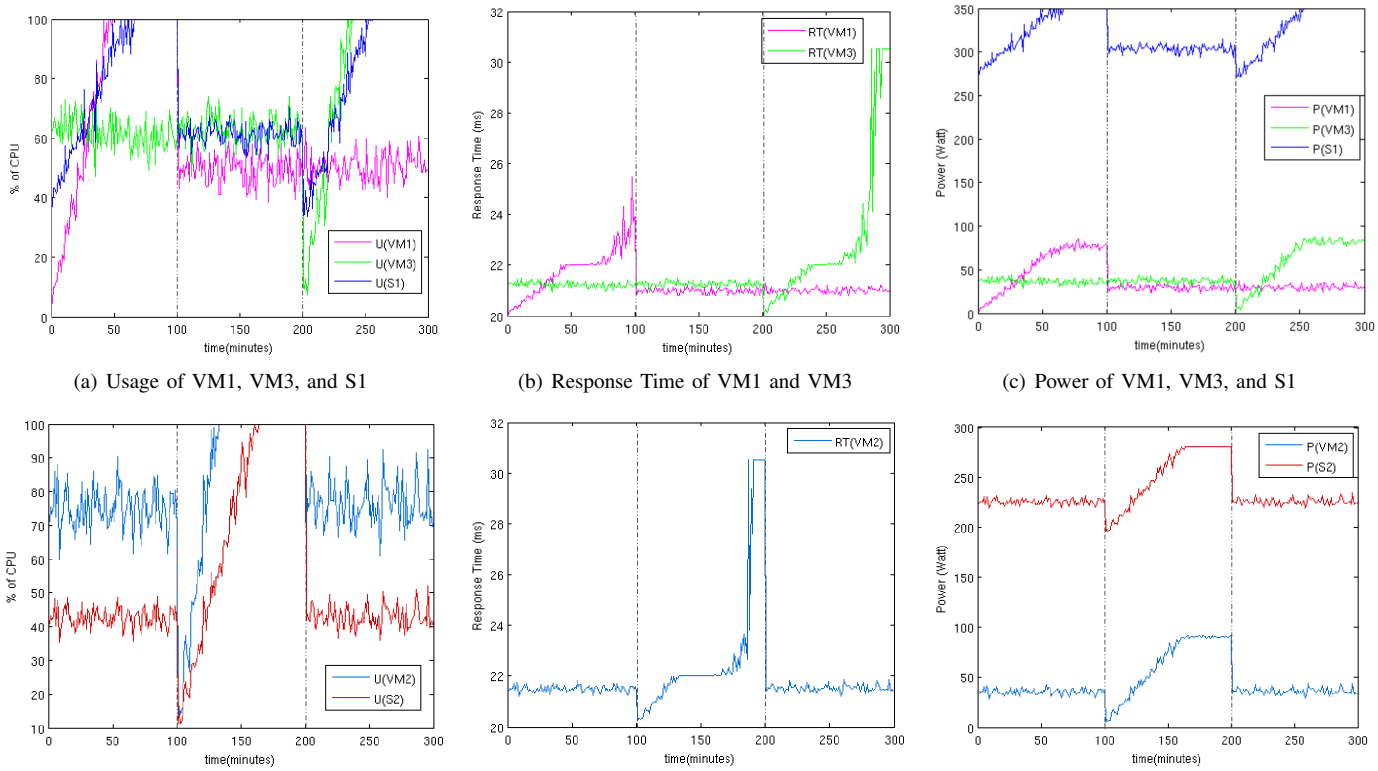
VMs are independent, but they are related to the power of the server which hosts them and they are also dependent from the CPU usage of the VM. This behavior is reflected in plots of Fig. 2(c) and Fig. 2(f).

Finally, we used the model for what-if analysis, using three configurations and evaluating quality of service and energy efficiency for each of them. QoS has been expressed as the total response time, while for energy efficiency we considered the total energy consumed at the server level. In "Configuration1" the system is the same as the previous tests, with three virtual machines on two servers. In "Configuration2" all the VMs are on S1 and S2 is turned off. In "Configuration3" a new server is added and VM3 is migrated on it. Tests have been conducted by increasing the load with the same rate for all the VMs. From Fig. 3 we can see that energy increases with the number of servers. On the contrary, QoS is better in "Configuration3", since more resources are available for each VM, and response time remains low even when the load is high.

The model has been implemented using MATLAB [15]. The code used for the experimental part presented in this section is freely available at [16].

## V. Conclusion

In this work, we presented a system modeling approach for describing the behavior of the components of a data center related to the deployment of a BP. Unlike other models for energy usage estimation, this work focus on the application level information and model a complex environment involving a whole data center using virtualization. Through this model, we were able to collect simulated monitoring data that can be used in the study of strategies for improving energy efficiency and quality of service of a BP. The model allows off line simulation of different workload levels and a what-if analysis, by changing the configuration of the data center and observing the generated data. The what-if analysis enables the evaluation of different system configurations, both in terms of energy efficiency and of quality of service, providing a tool for selecting the best solution given a load rate. Results discussed




(a) Usage of VM1, VM3, and S1

(b) Response Time of VM1 and VM3

(c) Power of VM1, VM3, and S1

(d) Usage of VM2 and S2

(e) Response Time of VM2

(f) Power of VM2 and S2

Fig. 2. Monitoring values on a system composed of two servers and three virtual machines

in Sec. IV showed that the monitored system behaves as expected, with an increasing critical level as the load rate increases. Even if with some approximation, the system allows some reasoning impossible to recreate in a real environment without affecting it.

In the future, the model will be used to collect data for testing and comparing different adaptation strategies to improve both quality of service and energy efficiency.

## REFERENCES


[1] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," *ACM SIGARCH Computer Architecture News*, vol. 28, no. 2, pp. 83–94, 2000.

[2] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "The design and use of simplepower: a cycle-accurate energy estimation tool," in *Proceedings of the 37th Annual Design Automation Conference*, 2000, pp. 340–345.

[3] S. Gurumurthi, A. Sivasubramaniam, M. J. Irwin, N. Vijaykrishnan, and M. Kandemir, "Using complete machine simulation for software power estimation: The softwatt approach," in *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*, 2002, pp. 141–150.

[4] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," in *In Proceedings of Workshop on Modeling, Benchmarking, and Simulation*, 2006, pp. 70–77.

[5] S. Rivoire, M. A. Shah, P. Ranganatban, C. Kozyrakis, and J. Meza, "Models and metrics to enable energy-efficiency optimizations," *Computer*, vol. 40, no. 12, pp. 39–48, 2007.

[6] T. Do, S. Rawshdeh, and W. Shi, "pTop: A Process-level Power Profiling Tool," in *Workshop on Power Aware Computing and Systems (HotPower'09)*, 2009.

[7] A. Kansal and F. Zhao, "Fine-grained energy profiling for power-aware application design," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 2, pp. 26–31, 2008.

[8] R. Bertran, Y. Becerra, D. Carrera, V. Beltran, M. Gonzalez, X. Martorell, J. Torres, and E. Ayguade, "Accurate Energy Accounting for Shared Virtualized Environments using PMC-based Power Modeling Techniques," in *International Conference on Grid Computing*, 2010.

[9] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proceedings of the 1st ACM symposium on Cloud computing*, 2010, pp. 39–50.

[10] A. Kipp, T. Jiang, M. Fugini, and I. Salomie, "Layered green performance indicators," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 478–489, 2012.

[11] A. Nowak, F. Leymann, D. Schumm, and B. Wetzstein, "An architecture and methodology for a four-phased approach to green business process reengineering," in *Proceedings of the 1st International Conference on ICT as Key Technology for the Fight against Global Warming, ICT-GLOW 2011, August 29 - September 2, 2011, Toulouse, France*, ser. Lecture Notes in Computer Science (LNCS), vol. 6868. Springer-Verlag, 2011, pp. 150–164.

[12] M. Pedram and I. Hwang, "Power and Performance Modeling in a Virtualized Server System," in *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, 2010, pp. 520–526.

[13] D. Kliazovich, P. Bouvry, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, pp. 1–21, 2010.

[14] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: eliminating server idle power," in *Proceedings of the 14th international conference on Architectural support for programming languages and operating systems*, ser. ASPLOS XIV. New York, NY, USA: ACM, 2009, pp. 205–216. [Online]. Available: http://doi.acm.org/10.1145/1508244.1508269

[15] MATLAB, *version 7.14.0 (R2012a)*. Natick, Massachusetts: The MathWorks Inc., 2012.

[16] M. Vitali, "dcSimulation," https://github.com/monicavit164/dcSimulation, 2013.