Improved Energy-Efficiency in Cloud Datacenters with Interference-Aware Virtual Machine Placement

Ismael Solis Moreno[1], Renyu Yang[2], Jie Xu[1, 2], Tianyu Wo[2]

School of Computing[1]
University of Leeds
Leeds, UK
{scism, J.Xu}@leeds.ac.uk

School of Computer Science and Engineering[2]
Beihang University
Beijing, China
{yangry, woty}@act.buaa.edu.cn

***Abstract* — Virtualization is one of the main technologies used for improving resource efficiency in datacenters; it allows the deployment of co-existing computing environments over the same hardware infrastructure. However, the co-existing of environments - along with management inefficiencies - often creates scenarios of high-competition for resources between running workloads, leading to performance degradation. This phenomenon is known as *Performance Interference*, and introduces a non-negligible overhead that affects both a datacenter's Quality of Service and its energy-efficiency. This paper introduces a novel approach to workload allocation that improves energy-efficiency in Cloud datacenters by taking into account their *workload heterogeneity*. We analyze the impact of performance interference on energy-efficiency using workload characteristics identified from a real Cloud environment, and develop a model that implements various decision-making techniques intelligently to select the best workload host according to its internal interference level. Our experimental results show reductions in interference by 27.5% and increased energy-efficiency up to 15% in contrast to current mechanisms for workload allocation.**

Keywords — Cloud computing, energy-efficiency, performance interference, virtual machine placement, workload heterogeneity

I. INTRODUCTION

Cloud Computing is "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction*" [1]. It is experiencing rapid growth as it promises to reduce maintenance and management costs in comparison with in-house infrastructure [2]. Despite its commercial advantage of reduced energy consumption on the client side, Cloud providers still need to address a number of key challenges, such as striking a balance between optimal energy efficiency and satisfying the increasing demand and high performance expectations of users.

The first generation of energy-efficient Cloud computing approaches have introduced mechanisms to dynamically resize the pool of servers based on actual demand [3, 4]. Additionally, others such as [5, 6] have proposed to extend these mechanisms with enhanced migration and server activation policies to reduce Service Level Agreement (SLA) violations. However, these approaches neglect potential inefficiencies at a fine-grained level such as the overhead produced by the high competition for resources in virtualized environments [7]. If the approaches do not take into account such inefficiencies, their claimed energy-efficiency and performance improvements may be drastically diminished under real conditions. Cloud computing datacenters are multi-tenant environments where diverse workload types live together. Normally encapsulated into Virtual Machines (VMs), these workloads are co-allocated into the same servers sharing the underlying physical infrastructure to maximize the datacenter utilization. Although virtualization offers environmental and fault isolation, it does not guarantee that the resource consumption of a VM will not affect the performance of other VMs running on the same server [8]. This condition creates scenarios of high-competition for resources that could negatively affect the Quality of Service (QoS) specified in SLAs. This phenomenon is known as *Performance Interference* and its effect on the QoS of workloads has been previously analyzed in [8-12]. However, current approaches have yet to consider the impact of such interference on a datacenter ́s energy-efficiency. An understanding of this phenomenon is critical if we are to design energy-efficient mechanisms that maintain performance under realistic environmental conditions.

In this paper, we analyze the impact of performance interference on energy-efficiency and propose a model to reduce energy waste by taking into account the workload heterogeneity that exists in Cloud environments. Our core idea is to co-allocate different types of workloads based on the level of interference that they create, in order to reduce the resultant overhead and consequently improve a datacenter ́s energy-efficiency. The proposed model classifies incoming workloads based on their resource usage patterns, pre-selects the hosting servers based on resources constraints, and makes the final allocation decision based on the current servers' performance interference level. In order to conduct this study we emulate the different workload types derived from the Google Cloud tracelog [13], and execute them on the iVIC Virtual Computing Infrastructure [14] to measure their interference and energy consumption. iVIC provides flexible access to virtual cluster computing environments on top of common resources, and enables users to dynamically create customized and scalable virtual computing environments. Additionally, we perform simulation experiments using the CloudSim framework [15] to evaluate the overall impact of our proposed model in a highly dynamic scenario, also modeled from the Google tracelog. Our experimentation shows that our proposed model reduces interference by 27.5% and improves energy-efficiency by up to 15% in contrast with current allocation mechanisms in the analyzed environment. In particular, the major contributions of this paper are:

- The first analysis conducted to determine the impact of performance interference on energy-efficiency in Cloud environments.
- A novel method that takes into account workload heterogeneity in order to reduce the overhead and energy waste produced by performance interference.

The remaining sections are structured as follows: Section 2 discusses workload heterogeneity and presents a characterization of workloads derived from a real scenario; Section 3 performs an analysis of performance interference impact on energy-efficiency; Section 4 describes the proposed model; Section 5 describes the experimental environment and results; Section 6 discusses related work; Section 7 presents our conclusions and discusses future work.

II. WORKLOAD HETEROGENEITY

A workload is a specific amount of work computed or processed within the datacenter with defined resource consumption patterns. In this context, Cloud computing datacenters can be defined as pools of computer resources that can host a variety of different workloads ranging from long-running scientific jobs to transactional operations [16]. In such heterogeneous environments, workloads are different among them not only because the amount of resources that they consume but also because the placement constraints that they impose [17]. While task resource requirements describe *how much* resources a task consume, task placement constraints specify *which* resources are consumed and their characteristics.

To capture this diversity, we have analyzed the Google Cloud tracelog [13] and derived a workload classification based on length and resource utilization patterns (CPU and Memory). The tracelog contains information about 930 different users submitting 25 million running-tasks on a cluster composed over 12,000 servers for a period of 1 month. The classification presented in Fig. 1 was obtained applying *k*-means algorithm [18] on the day 18th that has highest ratio between number of submissions and the work performed in the entire tracelog. From the cluster centroid analysis, 3 different types of workloads can be outlined. These have been labeled as "*Small*", "*Medium*", and "*Large*" due to the proportions *P* of their 3 dimensions as presented in Table I. Here, the values of Length, CPU, and Memory have been standardized based on the maximum and minimum values from the tracelog to avoid skewed results due to the use of different metric units.

Regarding to the task constraints, we have assumed the list of 21 constraints and their probabilities defined by Sharma et al. [17]. This list is composed by the most popular constraints in Google computer clusters such as the one from where we have derived the classification. Each task can be associated with zero or more constraints as represented in Fig. 2 and each constraint is defined by a triple of machine attribute, relational operator, and desired value. The list includes machine attributes such as architecture, number of cores, number of disks, number of CPUs, kernel version, clock speed, Ethernet speed, and platform family. In combination, task types and constraints create a highly heterogeneous workload environment that can be exploited to reduce the negative effects of performance interference [11].

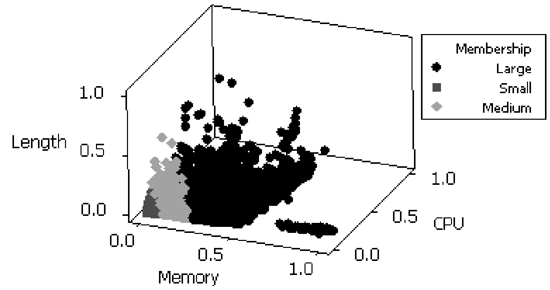


Figure 1. Google task classification plot.

TABLE I. TASK TYPE DIMENSIONS AND PROPORTIONS.

	Length	*P*	CPU	*P*	Memory	*P*
S	0.0007	1	0.0149	1	0.0089	1
M	0.0038	5	0.0810	5	0.0585	6
L	0.0107	15	0.2206	14	0.2556	28

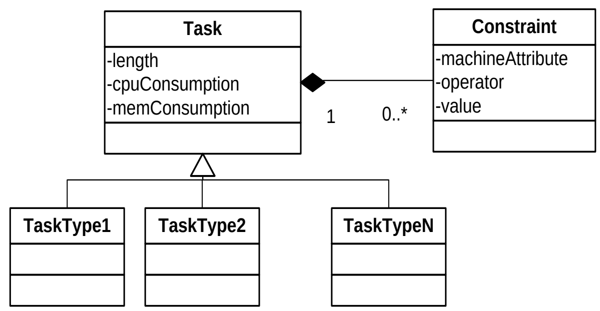
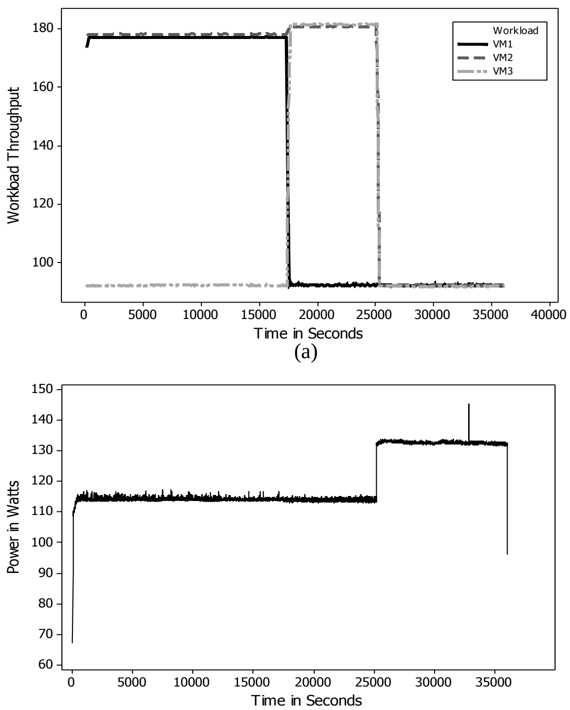


Figure 2. Task constraints model.

III. IMPACT OF PERFORMANCE INTERFERENCE ON ENERGY-EFFICIENCY

The impact of performance interference in virtualized environments has been typically measured in terms of QoS such as throughput, latency or response time. However, this phenomenon also affects other critical factors that include the energy-efficiency of the overall datacenter. When performance interference occurs, co-allocated workloads essentially fight for common resources creating overhead that increase the power consumption of individual servers. On the other hand, the remaining resources are mainly wasted until the overhead is dissolved. To provide an example, we have allocated 3 KVM VMs repeatedly running CPU-bounded workloads in the same virtualized server for 10 hours while the energy consumption is monitored. Each workload computes the 50th Fibonacci number using naive recursion to create a high competition for CPU time. Running alone, each workload requires in average 94.5 seconds to be completed but when running all together the performance for some of the VMs is reduced approximately 90% during some periods of time when interference occurs. In Fig. 3(a), it is observable that from time 0 to 25000 one VM primarily keeps the control of the resources greatly affecting the performance of the other two. During this period, the power consumption as observed in Fig. 3(b) steadily remains about 115 Watts in average. However, when the interference is reduced from time 25000 to 36000 the average power consumption increases up to 135 Watts. Although the increment of power is close to 17%, it is still small in comparison to the performance improvement of 90% for each affected VM. This suggests that when the interference increases, the energy-efficiency is reduced and vice versa.



(b)

Figure 3. Interference effects on (a) workload performance and (b) Server energy consumption.

In order to analyze the impact of performance interference on energy-efficiency in real cloud environments, we have emulated the 3 task types derived from the Google tracelog in Section 2. We have used Sysbench [19] "*memory-test*" to stress CPU and Memory based on the proportions *P* defined in Table I. Sysbench is a modular, cross-platform and multi-threaded benchmark tool for evaluating system parameters under intensive loads. In our emulation, each workload is defined by one or more Sysbench commands that execute a number of writing operations on pre-established memory blocks to create CPU and memory usage patterns according to the command in (1). Additionally, the configuration for each workload type is presented in Table II.

Sysbench --test=memory --memory-oper=write --num-threads=1 --memory-block-size=60M run (1)

Over a period of 12 hours, we deployed different pair combinations of these workloads in a virtualized environment using iVIC System [14] which is a KVM-based Virtual Computing Infrastructure. While the performance of each VM has been recorded using the libvirt API, the transient power and total energy consumption has been monitored through a Voltech PM3000 Ace power analyzer unit. The effects of

TABLE II. WORKLOAD CONFIGURATION.

Type	Length (Number of Operations)	Threads (Sysbench Commands)	Memory Allocation (MB)
Small	1707	1	60
Medium	8535	5	360
Large	23898	14	1680

performance interference in each pair has been measured by extending the Combined Score (CS) proposed in [11] to calculate the "*Combined Interference Score*" (CIS) as described in (2).

$$CIS(s) = \sum_{i=1}^{n} \frac{P_i - B_i}{B_i} \quad (2)$$

Where n is the total number of VMs co-allocated in the server s, P_i is the performance of the VM_i when combined with other, and B_i is the performance of the VM_i when running in isolation. Regarding to the energy-efficiency decrement, it is calculated as described in (3) where E is the expected energy-efficiency and A is the actual energy-efficiency obtained for each pair. In both cases energy-efficiency is defined as the ratio of work (performed or expected) by the total amount of energy consumed. The expected work is supposed to be the aggregated work of individual VM when running in isolation. Correspondingly, the actual work is calculated by the total work of each VM when is combined with others.

$$\Delta EE(s) = \frac{E - A}{E} \quad (3)$$

Table III describes the performance and energy measurements as long as the values obtained of *CIS(s)* and *ΔEE(s)* for each combination. Additionally, Fig. 4 illustrates the impact of performance interference on the energy-

TABLE III. PERFORMANCE AND ENERGY OBSERVATIONS.

Type	Perf Comb[a]		Energy (Whr)	CIS	ΔEE
	VM_1	VM_2			
SS	61.36	61.28	1558.92	0.482	0.241
SM	60.06	62.23	1555.80	0.519	0.259
SL	59.19	66.38	1535.88	0.516	0.257
MM	62.23	62.43	1563.60	0.522	0.261
ML	60.25	66.97	1541.64	0.527	0.263
LL	66.38	66.38	1538.40	0.496	0.248

a. The performance in isolation for S, M, L was measured in 80.9, 84.4, 88.3 operations/s respectively.

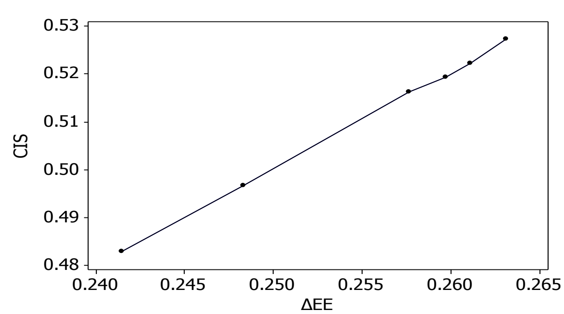
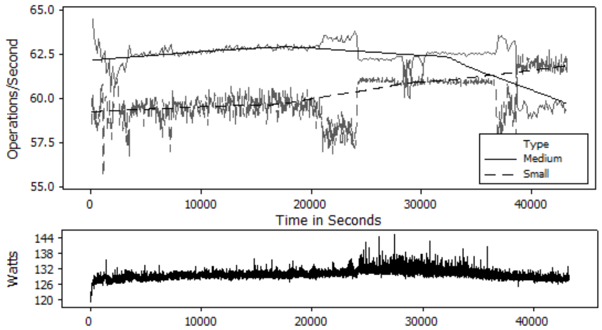
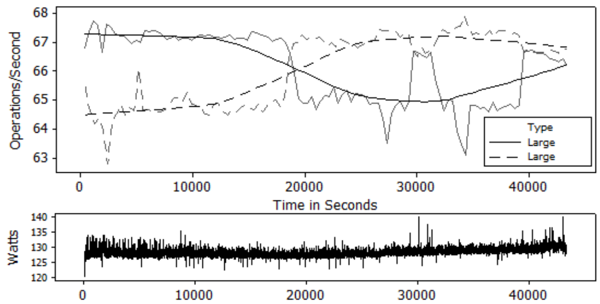


Figure 4. Impact of performance interference on energy-efficiency decrements for the evaluated workload pairs.

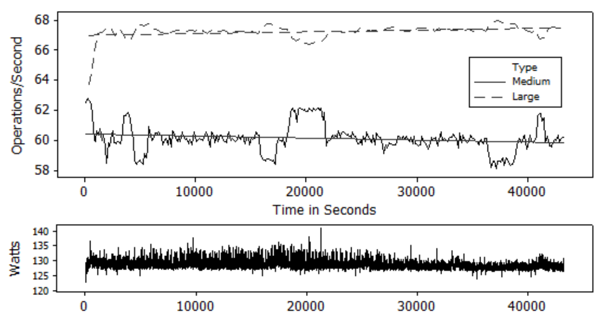
efficiency decrements for the analyzed workload combinations.



(a)



(b)



(c)

Figure 5. Performance Interference characterization (a) Medium Vs Small (b) Large Vs Large (c) Medium Vs Large.

As observed, the energy-efficiency almost linearly decreases when the interference increases. This is mainly caused by reduction of the number of operations executed per Watt consumed when interference occurs. This phenomenon is observable from Fig. 5 where the interference is characterized in terms of both performance and power consumption for a sample of combinations. For example, Fig. 5(a) illustrates the case of combining Small and Medium workloads. Here while the performance of each VM gets close to each other indicating the reduction of the interference from time 25,000 to 35,000, the power consumption is increased from 130 to 135W. However, this is a transient increment that slightly affects the overall energy consumption. This means that when the interference is reduced the power increment does not significantly affect the energy-efficiency. On the other hand, when the interference is strong the resulting performance degradation drastically affects the amount of work computed in contrast to average Watts utilized for such computation. The same phenomenon is also observable in Fig.5 (b) for the combination of Large workloads from time 29,500 to 31,200 and in Fig. (c) for the combination of Medium and Large workloads from time 5700 to 21,700.

IV. Proposed Model and Approach

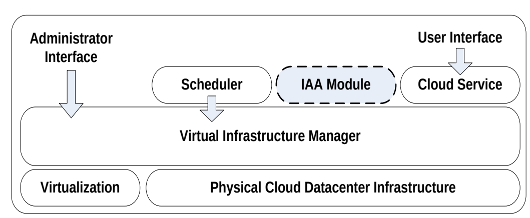
As observed in Fig. 6(a), the proposed model extends the traditional Cloud Management System (CMS) architecture with an Interference-Aware Allocation Module (IAA). It evaluates the incoming workloads and the datacenter servers to create a balanced mixture of workload types. The IAA module is integrated by four components: the Workload Classifier Service (WCS), the Resource Description Reasoner (RDR), the Dynamic Status Monitor (DSM), and the Matchmaker Service (MMS). The IAA module is also supported by the Resource Information Service (RIS) that provides the data collected from monitoring the resource utilization patterns. Fig. 6(b) illustrates the interaction of the listed components.

A. *Workload Classifier Service*

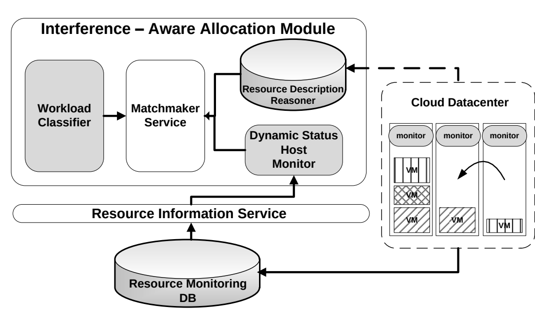
The WCS receives the characteristics of the incoming workload and determines its membership regarding to the classification presented in Section 2. The classifier is implemented as a Decision Tree (DT) which is automatically constructed based on the entropy measurement of historical data using the ID3 algorithm as defined in [20]. ID3 is a machine learning technique to construct classification trees based on the homogeneity level of the provided training dataset. In this context, entropy is used as a measurement to characterize the homogeneity of an arbitrary collection of examples as specified in (4).

$$Entropy(x) = -\sum_{i=1}^{n} p_i \log p_i \tag{4}$$

Where x is a random variable with n possible values and p_i is the mass probability function of the outcome i. The proposed learning module applies the ID3 algorithm on randomly selected cases to separate the data into targeted classes creating a workload classification tree. Later, the prediction module determines the membership of incoming workloads comparing



(a)



(b)

Figure 6. Proposed model (a) overview (b) components interaction.

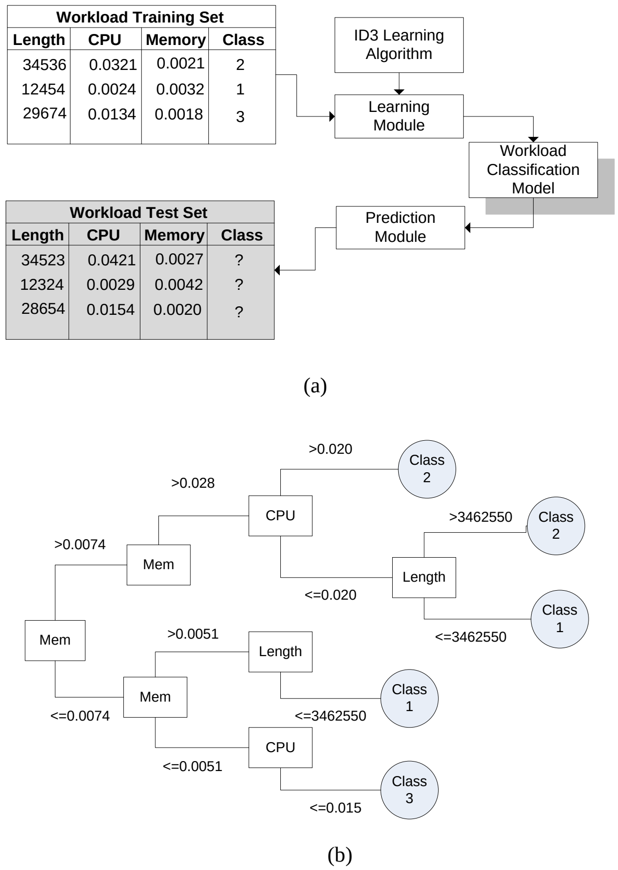


Figure 7. Decision tree classification mechanism (a) training process (b) schema.

each one their characteristics against the tree nodes previously defined. This process is illustrated in Fig. 7(a) and a fraction of the derived workload classification is presented in Fig. 7(b).

Decision Trees present high-classifying speed, strong learning ability, and simple construction with a neglectable overhead for large training sets [20, 21]. It takes less than 1 second to construct the classification tree for a training set of 1000 elements randomly selected from the Google tracelog. The tree was evaluated against 100 randomly created test sets of 1000 elements from the same dataset. It accurately determined the membership of incoming workloads in an average of 98.5% of the cases.

B. *Resource Description Reasoner*

The RDR is responsible to preselect a subset of servers that fulfill the constraints $C= \{c_1, c_2, c_3, ..., c_n\}$ imposed by the incoming workload. It maintains a case library that describes all the servers in the datacenter and their features $F= \{f_1, f_2, f_3, ..., f_m\}$. The reasoner takes the server features described in the library and the set of workload constraints and determines their similarity measuring the spatial distance between them as described in (5) and (6).

$$d(c_i, f_i) = d(f_i, c_i) = \sqrt{(c_i, f_i)^2} = |c_i - f_i| \qquad (5)$$

$$D(C,F) = \sum_{i=1}^{n} w_i d(c_i, f_i) \qquad (6)$$

Where $d(c_i, f_i)$ is the spatial distance between the constraint c and the server feature f, w is the constraint importance defined by the user, and $D(C,F)$ is the overall spatial distance between the workload with constraints C and the server with features F. The RDR returns to the MMS a list of "*unique identifiers*" (uids) of those servers that exactly match the requirements or those that have a minimum level of similarity defined by the Cloud administrator.

C. *Dynamic Status Monitor*

The DSM is responsible to maintain the status of each server in the datacenter. Every time a VM is deployed or removed from a specific server, the dynamic characteristics of that server including resources availability, energy-efficiency, and CIS are determined and stored by the DSM. When required, this information is passed to the MMS to select the server with available resources and less interference impact. The availability A for a server s is determined for each resource $r = \{CPU, memory, disk, and bandwidth\}$ based on the maximum server availability $Max(r,s)$ and the sum of current allocation for each deployed VM $Alloc(r, vm)$ as defined in (7).

$$A(r,s) = Max(r,s) - \sum_{i=1}^{n} Alloc(r, vm_i) \qquad (7)$$

The energy-efficiency EE for a server s is calculated as the ratio of the work being computed w measured in Millions of Instructions (MI) and the used power $P(u)$ in Watts as defined in (8) to (10).

$$EE(s) = \frac{w}{P(u)} \qquad (8)$$

$$P(u) = \Delta Pow.u + (P(\alpha) - \Delta Pow.\alpha) \qquad (9)$$

$$\Delta Pow = \frac{P(\beta) - P(\alpha)}{\beta - \alpha} \qquad (10)$$

Where u is the system utilization, α and β are the lower an upper utilization levels $\alpha \le u \le \beta$ derived from the server profiling process as presented by SpecPower [22]. The interference score $CIS(s)$ is calculated using (2) defined in Section 3. The dynamic status is stored using a Hash Map structure in order to perform indexed searches based on the servers' uids.

D. *Matchmaker Service*

The MMS is responsible for orchestrating the previously described modules to select the best server based on the workload constraints, resource availability, and interference impact. When the MMS receives the workload characteristics, it requests the WCS to determine the workload's membership. Then it sends the workload constraints to the RDR to obtain the unique identifiers of the servers that fulfill such constraints. Afterwards, the server identifiers are sending to the DSM to get their current dynamic status. To select the best servers from the retrieved subset, the MMS first discard those whose resource availability is not sufficient to host the new workload. The set of suitable servers S is ranked based on the current weighted energy-efficiency *WEE* of each server s as defined in (11).

$$WEE(s) = EE(s) * \left(\frac{1}{CIS(s)} \right) \tag{11}$$

Then the insertion of the new workload is simulated for each server in S until the maximum weighted energy-efficiency WEE' is found. The detailed algorithm for server selection is presented in Table IV.

TABLE IV. SERVER SELECTION ALGORITHM.

Input: $m \rightarrow$ New workload membership,
$S \rightarrow$ Set of ranked servers
Parameters: $w \rightarrow$ New workload to allocate
Output: $s_i \rightarrow$ Selected server
(1) *MaxWEE'* ← 0
(2) s_i ← NULL
(3) **For Each** s **in** S
(3.1) s ← **Insert** w
(3.2) *EE'* ← **Calculate** Energy-efficiency **of** s (Eq.7)
(3.3) *CIS'* ← **Calculate** Interference ratio **of** m **in** s (Eq.1)
(3.4) *WEE'* ← **Calculate** *WEE* **based on** *EE'* **and** *CIS'* (Eq.10)
(3.5) **If** *WEE'* **is greater than** *MaxWEE'*
MaxWEE' ← *WEE'*
$s_i \leftarrow s$
(3.6) **Else** break_loop
(4) **Return** s_i

V. EXPERIMENTAL ASSESSMENT

The objectives of the conducted experiment are assessing the performance interference reductions and energy-efficiency improvements obtained by the proposed approach in comparison to the current Google's allocation mechanism. For that reason we have configured a simulated environment based on CloudSim framework [15]. This allows us to reproduce the desired variability and dynamicity of Cloud datacenter in a controlled way. The entire workload and user models have been derived from the Google's tracelog while the server characteristics and energy models are depicted from SpecPower results [22]. The environment consists of users with 6 different behavioral profiles as listed in Table V submitting 5000 tasks with the characteristics described in Section 2. The cluster is composed by 100 heterogeneous servers from 3 different platforms with the characteristics described in Table VI. Additionally, the complete energy models are presented in Fig. 8.

TABLE V. USER SUBMISSION PATTERNS.

User Model	
Profile ID	***Avg. Submission Tasks/second***
1	0.00056
2	0.00005
3	0.00432
4	0.28420
5	0.17870
6	1.24700

TABLE VI. SERVER PLATFORM PROFILES.

Server Model			
Platform	***CPU Capacity ssj_ops@100%***	***Memory GB***	***Energy in Watts Idle/Max***
ProLiant DL365 G5	337,543	8	144 / 268
PRIMERGY RX200 S7	1,338,554	32	58.6 / 257
1022G-NTF	793,535	48	70.3 / 213

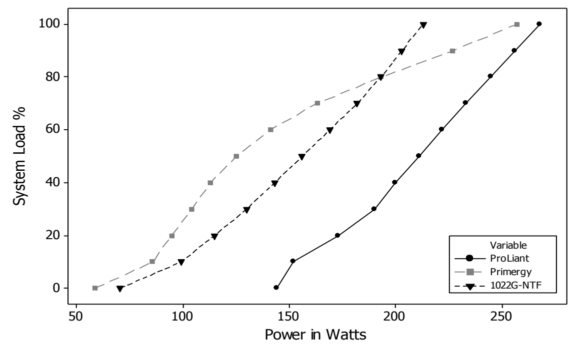
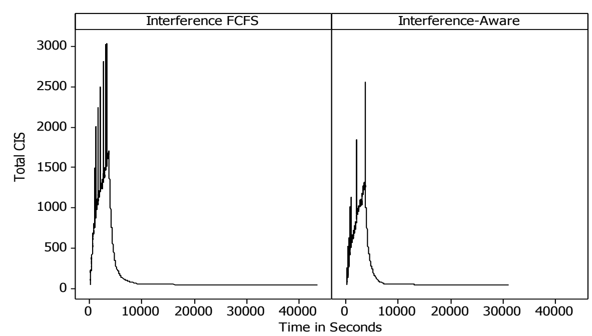


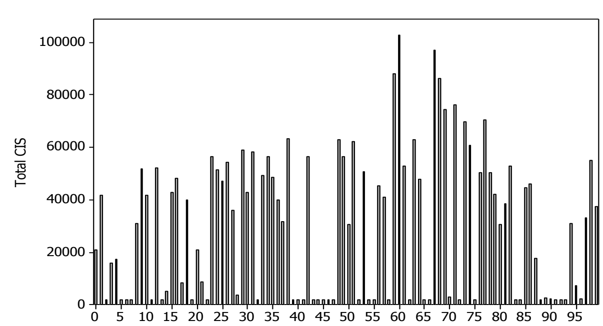
Figure 8. Energy models of the three selected platforms.

A. Experimental Results

The proposed interference-aware mechanism is contrasted against the prioritized First-Come, First-served (FCFS) algorithm according to the scheduling system described by Google in [23]. Initial results show a significant reduction in the levels of performance interference and improvement on the datacenter's energy-efficiency. The "*Combined Interference Score*" (*CIS*) is reduced in average from 1044.20 to 756.86 units which represents an improvement of approximately 27.5% in comparison to the current scheduling system. This can be

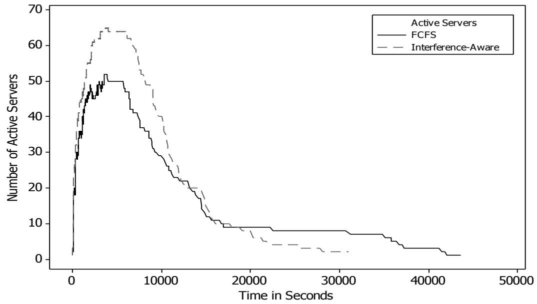


(a)

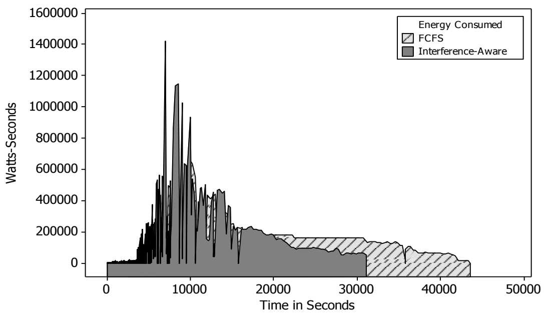


(b)

Figure 9. Performance Interference (a) per instant of time (b) per server.



(a)



(b)

Figure 10. (a)The number of active servers during the experimental time (b) Energy consumed by active servers during the experimental.

observed from Fig. 9(a) where is also evident that the peaks of interference have been reduced not only in number but also in proportion which is critical when the datacenter utilization is high. Furthermore, in Fig. 9(b) it is presented the total *CIS* accumulated per server. It is clear that specific servers produce more interference than others, for example servers 59, 60, and 67 accumulate a *CIS* over 80,000 units. This is caused by the great demand of the server characteristics that fulfill the constraints of a significant proportion of the submitted tasks.

In order to further reduce the effects of performance interference it is evident that providers need to balance the characteristic of offered infrastructure based on the dimensions of the demand. This requires reducing the number of servers that produce low interference and increasing those that are heavily used. To achieve this, providers need to clearly understand the overall datacenter workload and its resource utilization patterns as long as its specific placement constraints. As was expected, during peak utilization periods the interference-aware mechanism increases the number of working servers. In Fig. 10(a) we can observe that at time 5,000 when high utilization occurs the number of active servers was incremented from 50 to 63. This is the result the *WEE* evaluation performed in instruction (3.5) from Table IV. When the energy-efficiency on a specific server is drastically affected due to its current workload composition, the proposed scheduling mechanism deploys the newly incoming tasks on less utilized servers. Although this creates transient peaks on the energy consumption as observed in Fig. 10(b), it reduces the performance impact and therefore the completion time of the workload from 45,000 to 30,000 seconds. This time difference and its intrinsic savings on energy consumption represent an improvement of 15% on the datacenter's efficiency as observed also in Fig. 10(b). From these observations we can clearly depict a trade-off between performance interference and resource availability. That is when performance interference is reduced, the availability of servers is impacted and vice versa. The critical aspect is to find a balance between both dimensions to reduce the negative effects of performance interference while datacenter's resource availability is maintained. In this case the use of resources overallocation mechanisms [24] along to interference-aware approaches can help to improve the obtained results.

VI. RELATED WORK

The negative effect of performance interference in virtualized environments has been previously analyzed. This section describes and discusses the most relevant related work approaching the problem. Younggyun et al. [12], present an study that evaluates the performance impact of co-allocating pairs of different applications in virtualized servers by analyzing system-level characteristics including CPU, memory, and disk utilization. In this paper the authors proposed a model to predict the performance of a new incoming application based on previous observations. Gupta et al. [25], discuss the sources of interference at Xen's Virtual Machine Monitor (VMM) for I/O intensive workloads. They propose a set of primitives implemented at hypervisor-level to improve the resource sharing mechanisms and mitigate the performance impact caused by co-allocated VMs. Pu et al. [11], present a complete analysis of performance interference in Xen hypervisor. In this analysis they demonstrate that co-allocating different types of workloads reduces the performance interference in virtualized environments. Additionally, they present a set of performance metrics to outline points of conflict among the studied workloads. Govindan et al. [10], also analyze the phenomenon of performance interference at Low-level Cache (LLC). They propose a technique to predict the performance interference due to shared processor cache using synthetic cache loader benchmarks to profile the performance of mixed applications.

It is observable that the main related approaches have been focused on QoS aspects completely neglecting the impact on energy-efficiency produced by this phenomenon. If this is not considered, it can drastically diminish claimed energy-efficiency improvements by energy-aware mechanisms when applied under real conditions. Furthermore, they have only considered the amount of resources but not the placement constraints imposed by applications. These can also magnify the performance interference in real scenarios since they can increase the demand of certain servers in comparison to others. Finally, previous studies are largely based on unrealistic workload characteristics that can lead to misleading results in real operational environments. This is mainly caused by the lack of tracelogs and their consequently workload analysis from real and large-scale Cloud computing environments.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we have characterized workload heterogeneity derived from a real Cloud environment, and have provided the first approach to assess the impact of performance interference on a datacenter's energy-efficiency. Moreover, we have presented a mechanism to enhance energy-efficiency by exploiting the intrinsic workload heterogeneity that exists in Cloud environments. Experimental results show that our proposed mechanism reduces performance interference by 27.5% and increases energy-efficiency by up to 15% compared to current allocation mechanisms. From our presented study, the following conclusions can be drawn:

- *Interference not only affects the QoS of individual VMs but can also affect the energy-efficiency of the overall datacenter if not properly handled.* This is mainly produced by a drastic reduction in the work processed per Watt consumed, in comparision to the expected efficiency in dedicated servers.
- *Exploiting the inherent workload heterogeneity that exists in Cloud environments provides an excellent mechanism to improve both the performance of running tasks and energy-efficiency.* Combining specific workload types can reduce the performance impact introduced by virtualization as well as its negative effect on energy-efficiency.
- *Task constraints also play an important role, and can create bottlenecks that dramatically increase interference in specific servers.* When specific server characteristics are highly popular, but the population of these is low in the datacenter, a strong overhead is introduced that affects not only performance but also overall energy-efficiency.
- *Relying on real data is critical to understanding the real challenges in Cloud Computing and formulating assumptions under realistic operational circumstances.* This is especially true in very dynamic environments such as Cloud datacenters, where precise behavioural modeling is required to improve environmental efficiency while maintainng the QoS offered to customers.

As future work, we are planning to perform more experimentation to determine what other factors affect performance and energy-efficiency in Clouds, such as different hypervisors and hardware architectures. Additionally, a deeper study about the exposed interference impact on energy-efficiency needs to be conducted in order to formulate holistic models considering hardware, software, and workload patterns. Finally, we are also interested in evaluating the impact of performance interference on energy-efficiency when resources in the Cloud datacenter are over-allocated, in order to improve server availability whilst reducing interference effects.

Acknowledgments

The work in this paper has been supported in part by the National Basic Research Program of China (973) (No. 2011CB302602), the Mexican Council of Science and Technology CONACyT (No. 213247), the UK EPSRC WRG platform project (No. EP/F057644/1), and the Major Program of the National Natural Science Foundation of China (No. 90818028).

References

[1] D. Amrhein, *et al.*, "Cloud Computing Use Case," Cloud Computing Use Case Discussion Group, White paper, 2010.

[2] B. Gain. (2010, January 1) Cloud Computing & SaaS In 2010 *Processor Mag.* 12.

[3] R. Nathuji, *et al.*, "Exploiting Platform Heterogeneity for Power Efficient Data Centers," in *Proc. of the IEEE International Conference on Autonomic Computing* Washington, DC, USA 2007, pp. 5-15.

[4] K. Ley, *et al.*, "Cost- and Energy-Aware Load Distribution Across Data Centers," presented at the 22nd ACM Symposium on Operating Systems Principles, Montana, USA, 2009.

[5] R. Buyya, *et al.*, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges," in *Proc. of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, NV, USA, 2010, pp. 1-12.

[6] J. L. Berral, *et al.*, "Towards energy-aware scheduling in data centers using machine learning," in *Proc. of the 1st International Conference on Energy-Efficient Computing and Networking*, Passau, Germany, 2010, pp. 215-224.

[7] M. Hauck, *et al.*, "Towards Performance Prediction for Cloud Computing Environments based on Goal-oriented Measurements," in *in Proc. CLOSER*, 2011, pp. 616-622.

[8] R. Nathuji, *et al.*, "Q-clouds: managing performance interference effects for QoS-aware clouds," presented at the Proceedings of the 5th European conference on Computer systems, Paris, France, 2010.

[9] G. Casale, *et al.*, "A Model of Storage I/O Performance Interference in Virtualized Systems," presented at the Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops, 2011.

[10] S. Govindan, *et al.*, "Cuanta: quantifying effects of shared on-chip resource interference for consolidated virtual machines," presented at the Proceedings of the 2nd ACM Symposium on Cloud Computing, Cascais, Portugal, 2011.

[11] X. Pu, *et al.*, "Who is Your Neighbor: Net I/O Performance Interference in Virtualized Clouds," *Services Computing, IEEE Transactions on*, vol. PP, pp. 1-1, 2012.

[12] K. Younggyun, *et al.*, "An Analysis of Performance Interference Effects in Virtual Environments," in *Performance Analysis of Systems & Software, 2007. ISPASS 2007. IEEE International Symposium on*, 2007.

[13] Google. *Google Cluster Data V2*. Available: http://code.google.com/p/googleclusterdata/wiki/ClusterData2011_1

[14] J. Huai, *et al.*, "CIVIC: a Hypervisor based Virtual Computing Environment," presented at the International Conference on Parallel Processing Workshops, Xi'an, China, 2007.

[15] R. N. Calheiros, *et al.*, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software: Practice and Experience*, 2010.

[16] H. S. Abdelsalam, *et al.*, "Analysis of Energy Efficiency in Clouds," in *Proc. of Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, 2009, pp. 416-421.

[17] B. Sharma, *et al.*, "Modeling and synthesizing task placement constraints in Google compute clusters," presented at the Proceedings of the 2nd ACM Symposium on Cloud Computing, Cascais, Portugal, 2011.

[18] D. T. Pham, *et al.*, "Selection of K in K-means clustering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 219, pp. 103-119, January 1, 2005.

[19] A. Kopytov. (2012, July). *Sysbench Manual*. Available: http://sysbench.sourceforge.net/docs/

[20] Y. Liu and N. Xie, "Improved ID3 algorithm," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 2010, pp. 465-468.

[21] P. Geurts, *et al.*, "A machine learning approach to improve congestion control over wireless computer networks," in *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, 2004, pp. 383-386.

[22] Standard Performance Evaluation Corporation. (2012, July 7). *SPECpower_ssj2008 Results*. Available: http://www.spec.org/power_ssj2008/results/

[23] C. Reiss, *et al.*, "Google Cluster-Usage Traces: Format + Schema," Google Inc., White Paper, 2011.

[24] I. Solis Moreno and X. Jie, "Neural Network-Based Overallocation for Improved Energy-Efficiency in Real-Time Cloud Environments," in *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2012 IEEE 15th International Symposium on*, 2012.

[25] D. Gupta, *et al.*, "Enforcing performance isolation across virtual machines in Xen," presented at the Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware, Melbourne, Australia, 2006.