A Workload-Aware Energy Model for Virtual Machine Migration

Vincenzo De Maio*, Gabor Kecskemeti†, Radu Prodan*
*Institute of Computer Science, University of Innsbruck,
†MTA SZTAKI, Laboratory of Parallel and Distributed Systems
Email: {vincenzo,radu}@dps.uibk.ac.at, kecskemeti.gabor@sztaki.mta.hu

***Abstract*—Energy consumption has become a significant issue for data centres. Assessing their consumption requires precise and detailed models. In the latter years, many models have been proposed, but most of them either do not consider energy consumption related to virtual machine migration or do not consider the variation of the workload on (1) the virtual machines (VM) and (2) the physical machines hosting the VMs. In this paper, we show that omitting migration and workload variation from the models could lead to misleading consumption estimates. Then, we propose a new model for data centre energy consumption that takes into account the previously omitted model parameters and provides accurate energy consumption predictions for paravirtualised virtual machines running on homogeneous hosts. The new model's accuracy is evaluated with a comprehensive set of operational scenarios. With the use of these scenarios we present a comparative analysis of our model with similar state-of-the-art models for energy consumption of VM Migration, showing an improvement up to 24% in accuracy of prediction.**

I. Introduction

Recently, Cloud computing has emerged as a new paradigm by which computational power is hosted in data centres of specialised providers and rented on-demand to the users based on their occasional needs. Since power consumption significantly impacts the profits of these providers [1], they are inclined to maximise energy efficiency within their data centres. However, physical machines – the main constituents of data centres – are often under-utilised [2] leading to inefficient power utilisation. For this reason, a technique called *workload consolidation* is applied. This technique increases energy efficiency by mapping computational tasks on a subset of the data centre's machines and shut down the rest. Since in modern data centres computations are running within virtual machines (VMs), such mappings refer to running VMs on physical machines.

In order to assess whether a new mapping of VMs is beneficial energy-wise, prediction models are needed for their energy consumption. Such models should take into account all actors (e.g., VMs, physical hosts, network hardware) and activities (e.g., VM migration, powering off physical hosts) of workload consolidation. Among all activities, *VM migration* is one of the most widely used, because it provides the capability of moving the state of running VMs between physical machines, thus it allows to dynamically adjust the workload. Despite having a considerable impact on energy consumption [3], this activity has usually not been taken into account when building energy models for data centres and consolidation.

In recent years, several works modelled the energy impacts of VM migration. For example, [4] proposed a model based on network traffic generated by VM migration, while [3] focused on VM's CPU utilisation. However, past works focused on VMs only and have not considered other relevant actors involved in the migration process. For this reason, there is significant room for improvement in prediction models.

In this paper, we introduce a workload-aware energy consumption model for VM migration called `WAVM`^{3}. We aim at improving model precision by considering several previously neglected actors involved in the migration process. We also study the impact of different workload types on energy consumption related to VM migration. First, we identify the actors mostly involved in migrations. Then, we analyse their possible impact on data centre level energy consumption while considering different workloads. Our analysis uses CPU and memory-intensive workloads representing the most common and energy-impacting loads in data centres. As a result of our analysis, we identified several phases of VM migration from the point of view of energy consumption. Finally, we provided the model for the consumption of each relevant actor over each identified phase.

We target the Xen virtual machine monitor used by many commercial Clouds today such as Amazon EC2. Therefore, our model is restricted to scenarios with homogeneous source and target hosts, as Xen prevents execution of VM migration between machines with incompatible architectures. We limit our work to CPU and memory-intensive applications, as our experiments showed negligible energy impacts caused by network-intensive workloads during migration.

We built our model on measurements taken on a set of heterogeneous machines from a private Cloud. These measurements allowed us the experimental evaluation of the impact of different workloads on energy consumption by measuring the instantaneous power drawn of each involved actor while they were running benchmarks purposely designed to stress different physical machine components (e.g. CPU, memory). Based on the collected measurements for each benchmark, a new model was derived with linear regression. The regression's acceptance criterion was set as its normalised root mean square error (NRMSE) being notably lower than that of the other state-of-art models. Finally, we compared our model to other state-of-the-art models and we also performed preliminary investigations on a different set of machines showing the model's applicability for diverse configurations.

The paper is organised as follows. First, we review related work in Section II. Then we analyse the power characteristics of the migration process in Section III. From our analysis, we derive the theoretical model in Section IV and build a methodology for its validation in Section V. We present the results of our experimental validation in Section VI, followed by a comparison with other state-of-art models in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

Live VM migration has been proposed by [5] for the Xen hypervisor. Since then, it has been implemented in many popular hypervisors, such as Xen, KVM and VMWare. Many works like [6], [7], [8], [9] exploit live VM migration to perform energy-aware VM consolidation. However, energy consumption of VM migration is not taken into account in these works. Other works like [10], [11], [12] focused on the cost of live migration for Cloud data centres, but considered only performance and did not take energy consumption into account. Further works like [13] implemented a model for VM migration in a Cloud simulator, but do not provide models for its energy consumption. Recent works like [14] consider the time of live migration, but this study consider only CPU-intensive workloads and does not take energy into account. Other works like [15] propose a probabilistic approach to quantify the cost of VM live migration, but this cost does not take energy into account. First investigations about energy consumption of VM migration have been done by [16]. One of the first works that modelled time, energy and performance of live migration at the same time is [4], which identified a relationship between network bandwidth and energy consumption of Xen live migration. This work, however, considers only the load running on the migrating VM and makes the simplistic assumption that source and target host have the same energy consumption for VM migration. A similar work has been done for KVM live migration by [17]. Another model has been proposed by [18], but this model considers only CPU load.

In our work, we consider the workload of each actor involved in the migration process and extract a more accurate model for both live and non-live VM migration.

III. Power Characteristics of VM Migration

In this section we provide an overview of the power characteristics of VM migration. First, we describe the VM migration process and then the actors involved in this process. Afterwards, we investigate the workloads impacting the energy consumption of VM migration and finally, we identify the phases that occur during a migration.

A. VM migration

Although VM migration can be realised in different ways, we focus here on the most used approaches: non-live migration and live migration.

Non-live migration (sometimes referred as *suspend-resume migration*) approach consists of: (1) suspending the VM to be migrated, (2) transferring its state to the target host, and (3) resuming the VM on the target host.

Live migration has been proposed to reduce the downtime of the VM during migration. It consists of five steps: (1) moving the VM state from source to target host while the VM operates normally; (2) updating the state of target host with the modifications occurred on the source during state transfer; (3) repeating step (2) until a predefined termination criteria is reached (e.g., the size of the VM state difference reaches under a given threshold or maximum number of updates reached); (4) suspending the VM and transferring its last state changes to the target; (5) resuming the VM on the target when its state is consistent with the source; and (6) destructing the suspended VM on source.

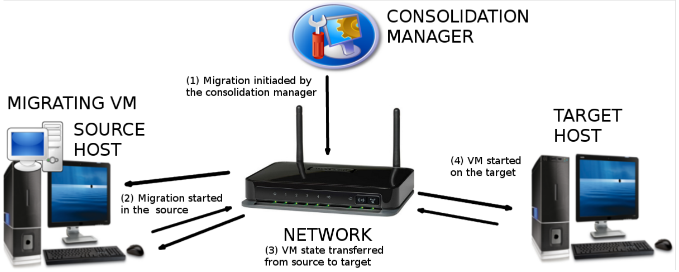


Fig. 1: Summary of the migration process.

B. Actors

In this section we identify the actors involved in the VM migration process, as detailed in Figure 1.

a) Consolidation Manager: constantly monitors the load of the data centre, selects the VM to be migrated and the target host, and finally initiates the migration. Afterwards, it returns to its previous operation.

b) Migrating VM: is the virtual machine to be transferred from the source to the target host, while the VM is also expected to be running services used by the customers of the data centre or the service provider utilising the VM.

c) Source host: is the physical machine that runs the migrating VM at the start of the migration process. The source establishes the initial connection with the target to communicate the intention of starting a VM migration.

d) Target host: is the physical machine designated by the consolidation manager as the destination for the migrating VM (i.e., the host that will execute the VM after the migration process completes).

e) Network: refers to the underlying communication infrastructure responsible for connecting the other actors and for supporting the VM state transfer.

In the rest of the paper we focus only on three of these actors: migrating VM, source host, and target host. We do not consider the consolidation manager because it does not further interact with the migration after initiating it. We also ignore the network infrastructure because it will affect the VM migration only at its maximum utilisation and it can be safely assumed that a VM migration will never be issued when the bandwidth between two hosts is fully utilised. Moreover, according to [19], the energy consumption of network switches is not proportional to the amount of traffic, therefore we can safely assume that is constant in this scenario.

Workload	*Migration type*	*Migrating VM*	*Source host*	*Target host*
CPU intensive	LIVE	Source/target load-dependant	Slowdown for state transfer	Slowdown for VM start/state transfer
	NON-LIVE			
MEMORY intensive	LIVE	Multiple transfers of VM state	Slight performance degradation	Slight performance degradation
	NON-LIVE	No influence		

TABLE I: Workload impact on VM migration according to the hosting actor.

C. Workloads

The three selected actors can influence the energy consumption of VM migration in different ways, especially depending on the workloads they are running. We analyse this aspect in the following paragraphs.

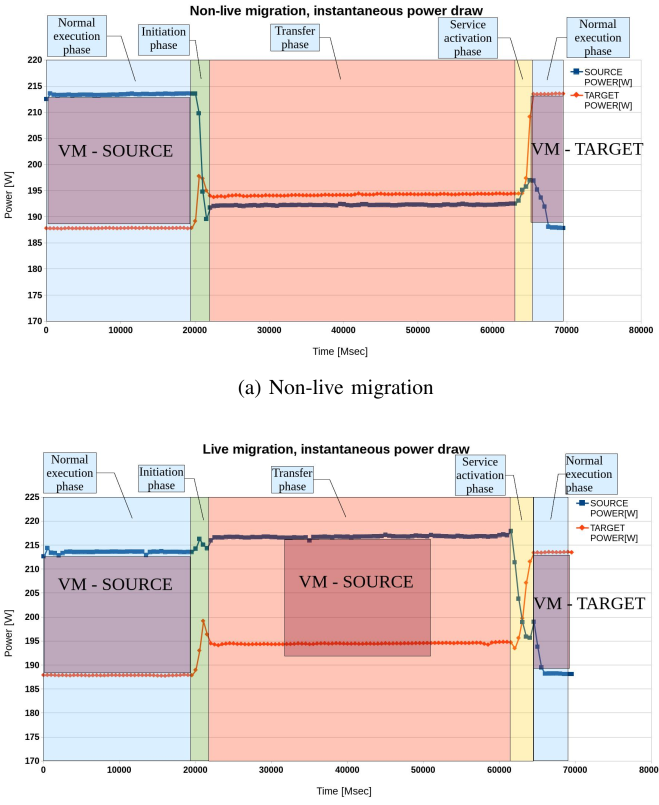
Although there may be different kind of workloads running in a data centre (e.g. CPU-intensive, memory-intensive, network-intensive, or mixed), in the following, we focus on the CPU and memory-intensive ones because they impact the VM migration process the most. Table I summarises the workloads' impact on VM migration. When the migrating VM is running a CPU-intensive workload, a performance drop may be experienced if the source and/or target hosts are fully loaded because the host's CPU must be shared between the workload of the hosts and the newly initiated migration process. If the migrating VM is running a memory-intensive workload that continuously updates RAM locations, it will highly impact the performance of the live migration approach since several state updates are needed to achieve a consistent VM state between the source and the target hosts. For these reasons, we only consider in this work (1) CPU intensive workloads running on source, target and migrating VM, and (2) memory-intensive workloads running on the migrating VM. We consider as memory-intensive workloads: (1) workloads using at least 90% of the memory allocated to the VM and (2) workloads with a high memory dirty ratio (i.e. a high percentage of memory pages marked as dirty over a given amount of time).

D. Migration energy phases

As we discussed in the previous sections, both live and non-live migration go through different phases that could lead to different energy-wise behaviour for each actor. In this section, we identify those phases of VM migration that differ from an energy point of view by collecting and analysing instantaneous power draw traces of a VM migration (see the traces and phases in Figure 2).

a) Normal execution: During this phase, every actor performs its normal operation as there was no migration decision taken so far. For the sake of simplicity, we ensure a constant energy consumption over this phase so the acquired power traces cannot be accounted for anything else but the migration process discussed in the later phases.

b) Initiation: This phase starts when the migration is requested by the consolidation manager and ends when the target host is prepared to receive the VM state. In case of non-live migration, the source host experiences a strong decrease in power consumption because the migrating VM is suspended in the beginning of this phase. In contrast during live migration, the source host will reach a new peak for energy consumption because of the preparation tasks necessary for sending the migrating VM to the target. The target host will show independent behaviour from the applied migration approach. It experiences peaks in its power draw due to checking of resource availability and acknowledging to the source that the migration can start.



(a) Non-live migration

(b) Live migration

Fig. 2: Energy consumption phases of non-live and live migration.

c) Transfer: During this phase, all the state information of the VM is transferred over the network from the source to the target host. Compared to the initiation phase, there is an increase of power drawn introduced by the exchanged VM state data in both the live and non-live migration approaches. For live migration, an additional consumption is recorded for the source because it needs to keep track of the modifications to the VM state.

d) Service activation: This phase starts after the VM state is transferred and ends when the VM is running on the target host. In this phase, the source host frees the resources that previously belonged to the migrating VM (please note, before freeing up the resources the source host must shut-down the migrating VM in case of live migration). The target host will instead run the VM machine. Finally, each actor returns to the normal execution phase.

IV. Model

In this section, we introduce our model for the energy consumption of each previously described migration phase. The energy consumption of the complete VM migration process will be the sum of the energy consumption of each phase.

A. Migration model

In this section we formally define VM migration as transferring the state of a migrating VM v from a source host $\texttt{S}$ to a target host $\texttt{T}$. As we saw in Section III-D, VM migration goes through different energy consumption phases. To delimit these phases, for each migration, we define m_s as the moment when the migration starts; t_s and t_e the time instances when the transfer phase of the migration starts and ends; and m_e as the instant when the migration ends. Thus the following time intervals define our phases: (1) between m_s and t_s is the initiation phase, (2) between t_s and t_e is the transfer phase and (3) between t_e and m_e is the activation phase.

B. Resource utilisation model

According to our analysis in Section III, the most impacting actors for VM migration are the source host ($\texttt{S}$), the target host ($\texttt{T}$) and the migrating VM (v). In this section, for each of these actors, we present a model for their resource utilisation to which their energy consumption is directly correlated. Both hosts and the VM have different types of resource use (e.g. CPU, memory, network), but according to our analysis in Table I, the most impacting parameters on migration are: (1) CPU utilisation of the source $\texttt{CPU}(\texttt{S}, t)$, target $\texttt{CPU}(\texttt{T}, t)$, and migrating VM $\texttt{CPU}(v, t)$ at time instance t, (2) memory dirtying ratio $\texttt{DR}(v, t)$ of the VM v expressed in the percentage of pages marked as dirty at time instance t , (3) the amount of memory $\texttt{MEM}(v)$ allocated to the migrating VM v, and (4) the available network bandwidth $\texttt{BW}(\texttt{S}, \texttt{T}, t)$ between the source and target hosts for transferring the state of the migrating VM.

If the VM is idle or suspended, then $\texttt{CPU}(v, t) = 0$ and $\texttt{DR}(v, t) = 0$. Otherwise, we define the memory dirtying ratio $\texttt{DR}(v, t)$ as:

$$\texttt{DR}(v,t) = \frac{\texttt{DIRTYPAGES}(v,t)}{\texttt{MEM}(v)}, \qquad (1)$$

where $\texttt{DIRTYPAGES}(v, t)$ is the number of pages marked as dirty at the time instance t in the memory of VM v and $\texttt{MEM}(v)$ is the VM memory size in pages.

The parameters $\texttt{CPU}(\texttt{S}, t)$ and $\texttt{CPU}(\texttt{T}, t)$ mainly depend on three terms: (1) CPU utilisation $\texttt{CPU}_{\texttt{VMM}}$ for arbitrating the hardware resources shared among the VMs, (2) CPU utilisation $\texttt{CPU}(v, t)$ of each VM v executed on the host h at the instant t and (3) CPU load $\texttt{CPU}_{\texttt{migr}}$ added by migration on both source and target:

$$\texttt{CPU}(h,t) = \texttt{CPU}_{\texttt{VMM}}(\mathcal{V}(h,t)) + \sum_{v \in \mathcal{V}(h,t)} \texttt{CPU}(v,t) + \texttt{CPU}_{\texttt{migr}}(h,t), \qquad (2)$$

where $\mathcal{V}(h, t)$ is the complete set of VMs running on the host $h \in \{\texttt{S}, \texttt{T}\}$ at time instance t other than the migrating VM v.

C. Energy model

For each physical host $h \in \{\texttt{S}, \texttt{T}\}$, the energy consumption of the migration is the integral of the instantaneous power drawn caused by the migration process throughout its duration $[m_s, m_e]$:

$$E_{\texttt{migr}}(h,v) = \int_{m_s}^{m_e} P_{(\texttt{i})}(h,v,t) + P_{(\texttt{t})}(h,v,t) + P_{(\texttt{a})}(h,v,t)dt, \qquad (3)$$

where the power drawn is represented as the sum of the power consumed over the three phases – identified in Section III-D –, initiation $P_{(\texttt{i})}$, transfer $P_{(\texttt{t})}$, and activation $P_{(\texttt{a})}$. The following subsections discuss our model for each of these power drawn functions. Integrating these values over the migration time, we obtain the energy consumption over each phase, $E_{(i)}(h, v)$, $E_{(t)}(h, v)$ and $E_{(a)}(h, v)$, respectively. By adding these values, we obtain energy consumption of VM migration $E_{\texttt{migr}}$:

$$E_{\texttt{migr}}(h,v) = E_{(i)}(h,v) + E_{(t)}(h,v) + E_{(a)}(h,v). \qquad (4)$$

1) Initiation phase: In this phase, we expect the power consumption on both hosts to depend on: (1) the increase in CPU usage for initiating VM migration and (2) the additional CPU usage for suspending the VM on the source host. On the source host, we also need to consider the resource usage of the VM that will still be running over this phase:

$$\begin{aligned} P_{(\texttt{i})}(h,v,t) = \alpha_{(\texttt{i})}(h) \cdot \texttt{CPU}(h,t) + \\ \beta_{(\texttt{i})}(h) \cdot \texttt{CPU}(v,t) + \mathcal{C}_{(\texttt{i})}(h) \end{aligned} \qquad (5)$$

where $\alpha_{(\texttt{i})}(h)$ and $\beta_{(\texttt{i})}(h)$ model the relationship between the CPU usage of the two hosts and of the migrating VM to the power consumption, and $\mathcal{C}_{(\texttt{i})}(h)$ include the power consumption the for establishing a connection between the two hosts. We approximate the power consumption with a linear function, as done in [20]. On the source host, it also includes the power consumption for suspending the VM. As the target is not yet involved in the execution of the VM, the $\texttt{CPU}(v, t) = 0$.

2) Transfer phase: Since transferring the state of the VM from the source to the target host is a network-intensive process, its power consumption is mainly related to the network bandwidth. In this phase, we also consider the CPU usage on both hosts proportional to the power consumption, while we also expect a linear relationship between dirtying ratio and power consumption due to the increased contention on memory.

$$\begin{aligned} P_{(\texttt{t})}(h,v,t) = \alpha_{(\texttt{t})}(h) \cdot \texttt{CPU}(h,t) + \beta_{(\texttt{t})}(h) \cdot \texttt{BW}(\texttt{S},\texttt{T},t) + \\ + \gamma_{(\texttt{t})}(h) \cdot \texttt{DR}(v,t) + \delta_{(\texttt{t})}(h) \cdot \texttt{CPU}(v,t) + \mathcal{C}_{(\texttt{t})}(h), \end{aligned} \qquad (6)$$

where $\alpha_{(\texttt{t})}(h)$ models the linear relationship between power and CPU usage, $\beta_{(\texttt{t})}(h)$ the relationship between bandwidth and power, $\gamma_{(\texttt{t})}(h)$ the linear relationship between the dirtying ratio and power consumption, $\delta_{(\texttt{t})}(h)$ the linear relationship between the migrating VM's CPU usage and its power consumption and $\mathcal{C}_{\texttt{t}}(h)$ the power consumption for moving the state of the migrating VM to the target host. We expect the latter to be higher on the target host than on the source because it also needs to load the VM state in memory. The main difference between live and non-live migration is that during a live migration, the migrating VM is still running on the source host and, therefore, we need to consider the power consumption on the host due to its workload (i.e., $\texttt{DR}(v, t) \neq 0$ and $\texttt{CPU}(v, t) \neq 0$). As the VM is not yet on the target, both the dirtying ratio and the migrating VM's CPU utilisation becomes 0 while evaluating power consumption on the target host.

3) Activation phase: After the transfer phase is completed, there are two remaining actions to be performed: resuming the VM on the target host and deallocating the resources occupied on the source host. Afterwards, due to the release of the resources previously owned by the migrating VM, on the source host, we consider the CPU load and a constant power consumption $\mathcal{C}_{(\mathtt{a})}(\mathtt{S})$ only. Concerning the target host, we need to consider the power consumed by the migrating VM that starts its execution, as well as the constant power consumed by the hypervisor to start the VM $\mathcal{C}_{(\mathtt{a})}(\mathtt{T})$:

$$P_{(\mathtt{a})}(h,v,t) = \alpha_{(\mathtt{a})}(h) \cdot \mathtt{CPU}(h,t) + \beta_{(\mathtt{a})}(h) \cdot \mathtt{CPU}(v,t) + \mathcal{C}_{(\mathtt{a})}(h) \quad (7)$$

where $\alpha_{(\mathtt{a})}(h)$ models the linear relationship between CPU usage and power consumption, and $\beta_{(\mathtt{a})}(h)$ models the relationship between the CPU usage of the starting VM.

V. Experimental Methodology

After describing our model, we introduce the methodology to evaluate its accuracy. First, we describe the rationale behind the experimental design, then introduce the hardware and software configuration for conducting the measurements.

A. Experimental design

Our experimental settings are summarised in Table IIa, and the VM and hardware configurations in Tables IIb and IIc. We used the Xen virtual machine monitor version 4.2.5, including both `xm` and `xl` toolstacks configured to perform the live and non-live migrations between two physical machines as specified in Table IIc. The two machines were connected through a networking switch. We performed the experiments on two sets of machines (`m01-m02` and `o1-o2`) with different CPUs and network cards/switch, to allow the validation of our model on different hardware configurations. For each experiment, we employed paravirtualized VMs mostly encountered in modern data centres as they ensure near-native performance. For the migrating VMs, we chose 4 GBs of memory size to assure a long enough migration time for the clearly identification of the energy consumption phases.

According to our analysis in Table I, CPU-intensive workloads running on source/target hosts and memory-intensive workloads running on the migrating VM have the highest impacts on the energy consumption VM migration. Therefore, we designed two families of experiments: *CPULOAD* and *MEMLOAD*.

1) CPULOAD: We investigate the impact of VM workload on live and non-live migration using two types of experiments:

a) CPULOAD-SOURCE: investigates the impact of CPU-intensive workloads running on the source host by migrating a VM to an idle target host. The load of the source is progressively increased from idle to 100% CPU utilisation to quantify its impact on VM migration. We also consider the case in which the VMs require more CPUs than the host can offer, to ensure some multiplexing amongst them.

b) CPULOAD-TARGET: investigates the impact of CPU-intensive workloads running on the target host by migrating a VM from a source host running the migrating VM only. The load of the target is progressively increased from idle to 100% CPU utilisation to quantify its impact. Also in this experiment, we consider the effects of multiplexing on hardware resources.

For the CPU-intensive workload, we use an OpenMP C implementation of a matrix multiplication algorithm for two reasons: it is used by many scientific workloads running on data centres, and it can be easily parallelised allowing us to load all virtual CPUs of the VMs taking part in the experiments with while it introduces only small communication and synchronisation overheads. Concerning the VM configuration, we select the `load-cpu` and `migrating-cpu` type among the instances described in Table IIb. We employ the `load-cpu` VM instance to load the physical host while migrating an instance of `migrating-cpu` type. We assign as many CPUs to these instances as needed to increase the load by 25% increments.

2) MEMLOAD: experiments study the effect of varying dirtying ratio (see Equation 1) in the migrating VM on the migration process. To compare the impact of the memory-intensive workloads with the CPU-intensive ones, we designed experiments involving CPU-intensive workloads running on both source and target, as follows:

a) MEMLOAD-VM: studies the impact of memory-intensive workloads by increasing the percentage of memory pages dirtied in the migrating VM. The source host is only running the migrating VM and the target is idle. This experiment serves as the baseline for the rest of the memory intensive ones.

b) MEMLOAD-SOURCE: investigates how live migration is differently impacted by: (1) CPU-intensive workloads running on the source host and (2) memory-intensive workloads running on the migrating VM. We perform a live migration of a VM running a memory-intensive workload from a source host running a CPU-intensive workload with increasing utilisation to an idle target.

c) MEMLOAD-TARGET: investigates how live migration is differently impacted by: (1) CPU-intensive workloads running on the target host and (2) memory-intensive workloads running on the migrating VM. We perform a live migration of a VM running a memory-intensive workload to a target host running a CPU-intensive workload with increasing utilisation. The source host is running the migrating VM only.

These experiments employ live migrations only, since non-live migrations have $\mathtt{DR}(v,t) = 0$. For this category of experiments, we chose a memory-intensive workload called *pagedirtier* implemented in ANSI C that continuously writes in memory pages in random order. We fixed the memory allocated to this application to 3.8 GB to avoid swapping effects incurring additional VM migration overheads, due to the continuous writing to the NFS storage and a consequent reduction of the available bandwidth. We employ again the `load-cpu` VM instances for generating load on the hosts and `migrating-mem` as the migrating VM (see Table IIb).

Experiment	*Configuration of source host*		*Configuration of target host*		*Configuration of migrating VM*		
	CPU	Memory	CPU	Memory	Instance	CPU	Memory
`CPULOAD-SOURCE`	$[0-100]\%$	5%	`idle`	5%	`migrating-cpu`	100%	5%
`CPULOAD-TARGET`	1×`migrating-cpu`	5%	$[0-100]\%$	5%	`migrating-cpu`	100%	5%
`MEMLOAD-VM`	`idle`	5%	`idle`	5%	`migrating-mem`	100%	$[5-95]\%$
`MEMLOAD-SOURCE`	$[0-100]\%$	5%	`idle`	5%	`migrating-mem`	100%	95%
`MEMLOAD-TARGET`	1×`migrating-mem`	5%	$[0-100]\%$	5%	`migrating-mem`	100%	95%

(a) Experimental design.

ID	*Number of virtual CPUs*	*Linux kernel*	*RAM*	*Workload*	*Storage size*
`load-cpu`	4	2.6.32	512MB	`matrixmult`	1GB
`migrating-cpu`	4	2.6.32	4GB	`matrixmult`	6GB
`migrating-mem`	1	2.6.32	4GB	`pagedirtier`	6GB
`dom-0`	1	3.11.4	512MB	VMM	115GB

(b) VM configurations.

Machine	*Available virtual cpus*	*Available RAM*	*Gigabit NIC*	*Gigabit switch*	*Xen version*
`m01`	32 (16×Opteron 8356, dual threaded)	32GB	Broadcom BCM5704	Cisco Catalyst 3750	4.2.5
`m02`					
`o1`	40 (20×Xeon E5-2690, dual threaded)	128GB	Intel 82574L	HP 1810-8G	4.2.5
`o2`					

(c) Hardware configuration.

TABLE II: Experimental setup.

B. *Energy measurement methodology*

We employ two Voltech PM1000+[1] power measurement devices connected to the AC side of the source and target hosts, measuring their instantaneous power drawn at a frequency of 2 Hz in order to capture the power consumption of a complete VM migration, including the pre- and post-migration execution phases. For each experimental run, we start measuring the hosts' power consumption and issue a VM migration only after the measured values stabilise. Similarly, we stop the measurements after the power consumption of the hosts stabilises too. We say that the power consumption of the host stabilises when we read twenty consecutive power measurements with a difference lower than 0.3%, that is below our measurement device's accuracy. Moreover, we repeat each experiment until the difference in variance between one run and the previous runs becomes less than 10%, resulting in at least ten runs for each experiment. From the power readings and the time intervals, we compute four energy metrics: initiation, transfer and activation energy of the corresponding VM migration phases (see Sections III-D and IV-C), and the total migration energy as the sum of the three metrics. In addition, we also measure the CPU and memory consumption during each migration using the `dstat` tool and average the values of all executions.

VI. Experimental Results

In this section, we show the results of our experiments described in Section V. For each experiment we report the instantaneous power consumption traced every 500 milliseconds (according to the resolution of our power measurement devices) which allows us to easily identify the migration phases. We extract the energy consumption for each phase by integrating the power over its length. We average each result over ten experimental runs to ensure statistical significance.

A. *CPULOAD-SOURCE*

The results for the CPULOAD-SOURCE experiment displayed in Figures 3a and 3b show that the instantaneous power consumption of a non-live migration follows the same trend for each CPU workload except the case with eight VMs, when we have multiplexing on the machine's CPUs. In this case, we clearly see that on the source host (Figure 3a) the power consumption trend follows a constant function, since it is proportional to the CPU usage that will never exceed its hardware-imposed limit beyond which the resources are shared between the VMs. In this case, the migrating VM is suspended when the migration starts and the load on the host drops when there is no multiplexing without affecting the power consumption.

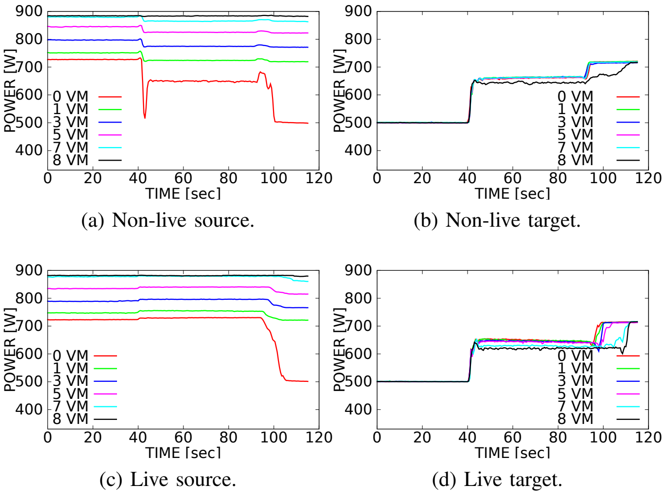
Concerning the target (Figure 3b), we notice a slightly lower power consumption from the beginning of the transfer phase when the source host has full CPU utilisation because of the reduced bandwidth to the target host (due to the 100% CPU load on the source host). A reduced bandwidth implies a lower power consumption and a longer transfer phase.

For live migration (Figures 3c and 3d), we observe an increased power consumption over the transfer phase due to the running VM because of: (1) the additional power consumption for network transfers and (2) the increased CPU usage of the virtualization software to handle the live migration. Concerning the source host, we notice a constant power consumption in case of CPU multiplexing, for the same reason as in Figure 3a.

Considering the power consumption on the target host (Figure 3d), we observe no significant differences compared to the non-live migration, except for a reduced consumption for the full CPU load with and without multiplexing. This is because the migrating VM is not suspended over the transfer phase and thus, it still uses CPU resources on the source host. Therefore, the source host is not able to exploit the full bandwidth available between the two hosts, leading to a scenario similar to the one observed in Figure 3b. We also notice a strong difference in power consumption before and after the migration in the 25% load scenario because the power drawn of the source host returns back to idle after the migration.

We conclude that CPU-intensive workloads have an impact on VM migration when running on the source, as bandwidth decreases when the CPU is fully loaded causing a longer transfer phase and a consequently, a higher energy consumption.

[1] http://www.voltech.com/products/poweranalyzers/PM1000.aspx



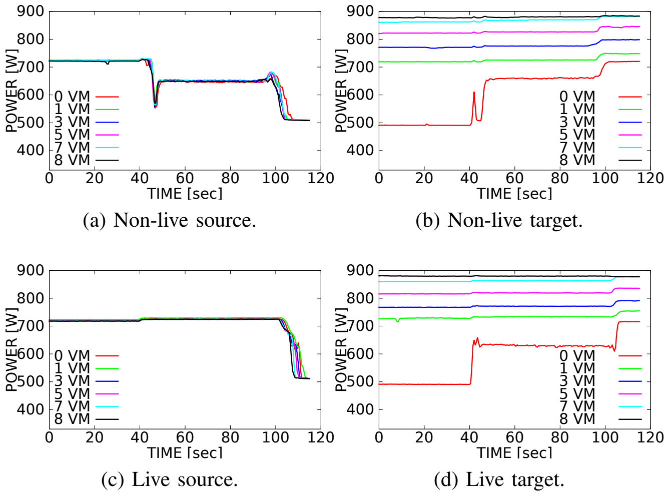
(a) Non-live source.

(b) Non-live target.

(c) Live source.

(d) Live target.

Fig. 3: CPULOAD-SOURCE results.



(a) Non-live source.

(b) Non-live target.

(c) Live source.

(d) Live target.

Fig. 4: CPULOAD-TARGET results.

B. CPULOAD-TARGET

For the CPULOAD-TARGET experiment, we observe fist in Figure 4a that the impact on the power consumption of source host is minimal when changing the load on the target. Concerning the target measurements in Figure 4b, we can notice (1) a small increase in power drawn due to the network transfer of the VM state and (2) a big increase in the power consumption when the migration is finished and the VM is up and running on the target. The impact of external load in this case is visible only when the target host is fully loaded, where the power resembles a constant trend since the host reached its CPU limit (see Equation 2).

For the live migration (Figure 4c), we notice for the source host a small increase in power consumption over the transfer phase due to: (1) the network transfer of the VM state and (2) the CPU increase for handling the migration. We do not notice any impact of the target load on this host except for the slight difference in case of multiplexing due to the additional load on the target host that prevents the VMM to use the full bandwidth. For the target host in Figure 4d, we see similar trends to the non-live migration except that: (1) the power drawn is slightly lower in the transfer phase and (2) the live migration takes at least 60 seconds longer. However, since this tendency is present also in the idle target case, it seems mostly related to hardware configuration than the host load.

C. MEMLOAD-VM

For the MEMLOAD-VM experiment, we observe in Figures 5a and 5b that the power consumption considerably changes with the dirtying ratio, with the difference that for the target host it does not go back to the idle level but slightly increases (since the VM is running on the target afterwards). On both hosts, the drop in power consumption during the transfer phase grows with the dirtying ratio because the VM experiences a longer suspension time to complete the migration by sending the more dirty memory pages from source to target.

Model	*Host*	*NRMSE* (non-live) (`m01 – m02`)	*NRMSE* (live) (`m01 – m02`)	*NRMSE* (non-live) (`o1 – o2`)	*NRMSE* (live) (`o1 – o2`)
`WAVM`^{3}	Source	11.8%	11.8%	12.5%	12.7%
	Target	12%	5%	16.3%	17.2%

TABLE V: Normalised root mean square error (NRMSE) of our model on the two datasets.

D. MEMLOAD-SOURCE

For the MEMLOAD-SOURCE experiment, we observe in Figure 6a that the transfer phase increases with the CPU load on the source host and the memory-intensive workload running on the VM. This slight increase is proportional to the decrease in bandwidth utilisation due to the increased CPU usage of the source. This tendency is better seen for high amount of loads for the target host (Figure 6b), when we notice a considerable increase in the transfer phase due to the reduced bandwidth. We also observe that that the CPU load on the source host has an impact on the energy consumption of migration even in case of memory-intensive workloads, for which reason we included it in Equation 6. Finally, we also notice on both hosts a considerable drop in power consumption towards the end of the transfer phase because of the VM suspension on the source due to the high dirtying ratio that transforms the live migration in a non-live one (i.e. the VMs are not accessible from the network during this time). The similarity with non-live migration is clear by looking at Figures 3a and 3b.

E. MEMLOAD-TARGET

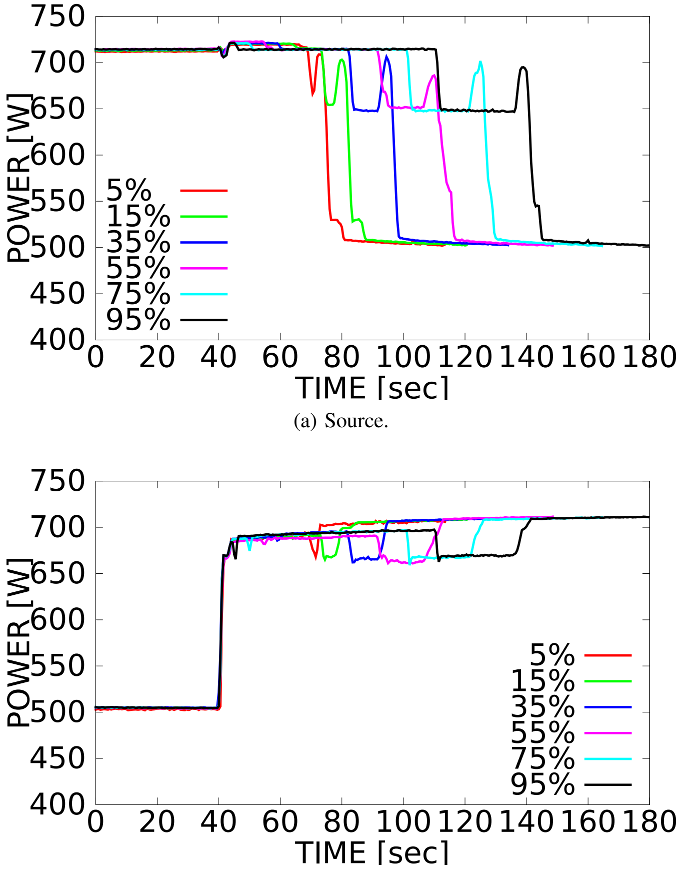
For the MEMLOAD-TARGET experiment, we see in Figure 7a that the transfer phase has a similar length on the source host, except for the slight difference in case of multiplexing due to bandwidth limitations on the target. The trends of the activation phase assume a different shape according to the amount of load. On the target host (Figure 7b), we observe a constant trend in power consumption except the idle case, when live migration becomes a non-live one as we can see by comparing the highlighted areas in Figures 4a and 4b.

Host	*Initiation*				*Transfer*				*Activation*			
	$\alpha_{(\mathtt{i})}$	$\beta_{(\mathtt{i})}$	$\mathcal{C}^1_{(\mathtt{i})}$	$\mathcal{C}^2_{(\mathtt{i})}$	$\alpha_{(\mathtt{t})}$	$\beta_{(\mathtt{t})}$	$\mathcal{C}^1_{(\mathtt{t})}$	$\mathcal{C}^2_{(\mathtt{t})}$	$\alpha_{(\mathtt{a})}$	$\beta_{(\mathtt{a})}$	$\mathcal{C}^1_{(\mathtt{a})}$	$\mathcal{C}^2_{(\mathtt{a})}$
Source	1.71	1.41	708.3	165	2.4	$1.08 \cdot 10^{-6}$	421.74	200	2.37	0	662.5	150
Target	3.18	0	596.06	162	2.56	$5.49 \cdot 10^{-7}$	520.214	210	1.88	17.01	499.56	100

TABLE III: Coefficients for non-live migration.

Host	*Initiation*				*Transfer*						*Activation*			
	$\alpha_{(\mathtt{i})}$	$\beta_{(\mathtt{i})}$	$\mathcal{C}^1_{(\mathtt{i})}$	$\mathcal{C}^2_{(\mathtt{i})}$	$\alpha_{(\mathtt{t})}$	$\beta_{(\mathtt{t})}$	$\gamma_{(\mathtt{t})}$	$\delta_{(\mathtt{t})}$	$\mathcal{C}^1_{(\mathtt{t})}$	$\mathcal{C}^2_{(\mathtt{t})}$	$\alpha_{(\mathtt{a})}$	$\beta_{(\mathtt{a})}$	$\mathcal{C}^1_{(\mathtt{a})}$	$\mathcal{C}^2_{(\mathtt{a})}$
Source	1.71	1.41	708.3	165	2.4	$1.52 \cdot 10^{-6}$	1.41	0.4	421.74	200	2.37	0	662.5	150
Target	3.18	0	596.06	162	2.56	$7.32 \cdot 10^{-7}$	0	0.4	520.214	200	1.88	17.01	499.56	100

TABLE IV: Coefficients for live migration.



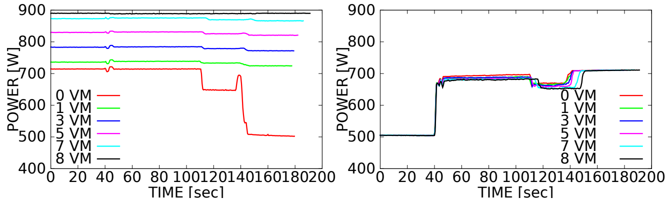
(a) Source.

(b) Target.

Fig. 5: MEMLOAD-VM results.

F. Regression analysis

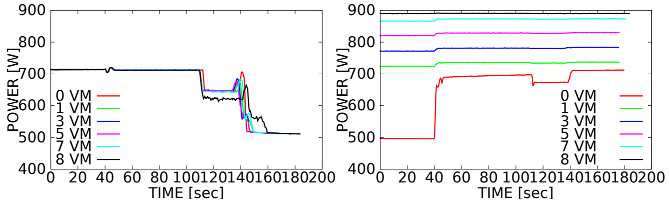
In this section we compute the model coefficients $\alpha, \beta, \gamma, \delta$ for each phase identified in the theoretical model using regression analysis based on the Non Linear Least Square algorithm. We select a training subset of the power readings from each phase to extract the model coefficients and use them afterwards as a model to predict the energy consumption. The training set used for this purpose is the 20% of the readings obtained by running our experiments on the machines $\mathtt{m01} - \mathtt{m02}$. The coefficients for non-live migration are summarised in Table III, while the coefficients for live migration are summarised in Table IV. To validate our model, we also used the same coefficients to predict the energy consumption of non-live and live migration on a different set of machines $(\mathtt{o1} - \mathtt{o2})$. When checking the results of our prediction on this new set, we observed that it was overestimating the measured values by a constant factor because the bias obtained from the training phase includes the idle power consumption of the physical machines. Therefore, we changed the bias by subtracting the difference in idle power between the two sets of machine. We will then use C^1 as bias for the prediction on $(\mathtt{m01} - \mathtt{m02})$ and C^2 for the prediction on $(\mathtt{o1} - \mathtt{o2})$. The error for our model in both datasets is shown in Table V. The discussion of our model's behaviour is presented in contrast to other state-of-the-art models in the next section.



(a) MEMLOAD-SOURCE source

(b) MEMLOAD-SOURCE target

Fig. 6: MEMLOAD-SOURCE results



(a) MEMLOAD-TARGET source

(b) MEMLOAD-TARGET target

Fig. 7: MEMLOAD-TARGET results

VII. Comparison

In this section, we compare the accuracy of our model with three other models available in the literature that take into account different parameters to model energy consumption of VM migration: HUANG [3], LIU [4] and STRUNK [17]. Next, we shortly describe each one of these models.

a) HUANG: The model of Huang et al. [3] is based on the assumption that the instantaneous power consumption P of each host is linear with the CPU utilisation of the VM v at the instant t $\texttt{CPU}(v,t)$ [20]:

$$P(t) = \alpha \cdot \texttt{CPU}(v,t) + \mathcal{C}, \tag{8}$$

where the istantaneous power $P(t)$ is linear by a factor of α and $\mathcal{C}$ is a hardware-related constant. We obtain the energy consumption by integrating P over the migration time $[m_s, m_e]$. This model perfectly suits scenarios when CPU utilisation has an impact on VM migration, but does not suit scenarios that involve other parameters (e.g. memory dirtying ratio, CPU load on migrating VM).

b) LIU: The model of Liu et al. [4] is based on the assumption that energy consumption of VM migration E_{mig} depends only on the amount of data $\texttt{DATA}$ exchanged by the two hosts during the VM migration:

$$E_{migr} = \alpha \cdot \texttt{DATA} + \mathcal{C}, \tag{9}$$

In their work, the authors compute the amount of data exchanged during migration as a function of VM memory size, memory transmission rate and memory dirtying ratio. Moreover, since they assume transfer is performed in several rounds, they compute the amount of data as the sum of the data sent in each round:

$$\texttt{DATA} = \sum_{r=0}^{n} \frac{(\texttt{MEM}(v) * \texttt{PAGESIZE})}{\texttt{BW}(\texttt{S},\texttt{T},r)} \cdot \texttt{DR}(v,t,r) \tag{10}$$

Where n is the number of rounds, $\texttt{BW}(\texttt{S},\texttt{T},r)$ is the bandwidth during the round r and $\texttt{DR}(v,t,r)$ the dirtying ratio over the round r. We use instead the amount of data transferred measured with our network instrumentation as the $\texttt{DATA}$ value. In this model, α models the linear relationship between the transferred data and energy consumption and $\mathcal{C}$ is a hardware-related constant. For this reason, the model is perfectly suitable for predicting the energy consumption of VMs workloads with high dirtying ratio. This model, however, does not consider the CPU load which generates modelling errors in case this has a high impact on the energy consumption. Moreover, it assumes that homogeneous hosts have the same consumption during migration. However, as stated also by [21], such an assumption could lead to inaccurate results.

c) STRUNK: The model of Strunk [17] considers the VM memory size $\texttt{MEM}(v)$ and the network bandwidth between source and target $\texttt{BW}(\texttt{S},\texttt{T})$ as parameters in a linear model:

$$E_{\texttt{migr}} = \alpha \cdot \texttt{MEM}(v) + \beta \cdot \texttt{BW}(\texttt{S},\texttt{T}) + \mathcal{C}, \tag{11}$$

where α and β model, the linear relationship between VM size and network bandwidth and $\mathcal{C}$ is a hardware-related constant. This model perfectly suits scenarios in which both hosts and the migrating VM are idle and does not take their load into account. Even though such conditions are very likely to happen in data centres [22], many works show the benefits of consolidating VMs executing tasks to/from hosts that are not idle [23]. Therefore, having a model able to predict the energy consumption of VM migration in different conditions can be helpful to decide whether this is beneficial energy-wise.

We train these models using the same training set used to train our model and the coefficients obtained for each model are summarised in Table VI. Afterwards, we compute three error metrics on the test set: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Normalized Root Mean Square Error (NRMSE). Each metric is summarised in Table VII. In the next subsections, we compare the results of our model named Workload-Aware Virtual Machine Migration Model ($\texttt{WAVM}^3$) with the other three.

Model	*Host*	α	β	$\mathcal{C}$
HUANG	Source	2.27	–	671.92
	Target	2.56	–	645.776
LIU	Source	2.43	–	494.2
	Target	2.19	–	508.2
STRUNK	Source	3.35	−3.47	201.1
	Target	5.04	−0.5	201.1

TABLE VI: Training phase coefficients for LIU, HUANG and STRUNK models.

Model	*Host*	*MAE* (non-live) [kJ]	*RMSE* (non-live)	*NRMSE* (non-live)	*MAE* (live) [kJ]	*RMSE* (live)	*NRMSE* (live)
$\texttt{WAVM}^3$	Source	1.8	2558	11.8%	6.3	8432	11.8%
	Target	1.7	1789	12%	3.6	4056	5%
HUANG	Source	1.8	2587	12%	5.5	9234	15.7%
	Target	1.8	2067	12.8%	7.1	9102	12.9%
LIU	Source	4.8	5812	26.9%	9.8	12117	36.3%
	Target	3.4	4121	25.3%	7	9622	29.4%
STRUNK	Source	0.026	3824	17.7%	0.028	4547	35.4%
	Target	0.058	5187	30%	0.019	4382	36.2%

TABLE VII: Comparison of $\texttt{WAVM}^3$ with other models on dataset m01-m02.

A. Non-live migration

By looking at Table VII, we observe that, among the analysed models, the one of Huang et al. provides the most accurate estimation for non-live migration. This is because non-live migration is mostly influenced by CPU usage which is the only parameter that this model takes into consideration. Since our model also takes CPU into account, we do not expect high variations in most of the scenarios. However, it can happen that one host is not able to use the full bandwidth if there is some multiplexing on the CPU. In such situations, network utilisation drops because CPU is not able to exploit all the network resources available and, therefore, network bandwidth cannot be ignored. Since our model also takes into account network bandwidth, it manages to have better estimations (-0.2% NRMSE for source host, -0.8% NRMSE for target host) when there is less network bandwidth available. Moreover, even though the MAE for the two models is very similar, we observe that the difference between RMSE and MAE is slightly higher for the model of Huang et al., showing that our model's estimation error has a lower variance too.

B. Live migration

The errors for the live migration are summarised in Table VII. Also in this case, the model of Huang et al. performs considerably better because it considers the CPU of source and target hosts, ignored by the other two, that has a considerable impact on energy consumption during VM migration. However, we notice an 18% increase in NRMSE versus the non-live migration error for the source host and a 16.2% increase in NRMSE for target host. This is because live migration should taken into account the CPU utilisation and the dirtying ratio of the migrating VM that is still running during the migration. Our model performs better because these parameters are instead considered, increasing the accuracy of prediction of Huang et al. by 3.9% (11.8% vs 15.7% NRMSE) for the source host and by 7.9% (5% vs 12.9%) for the target host.

VIII. Conclusion and Future Work

In this paper, we developed a new energy model for VM migration. We considered the impact of workloads running on different actors and identified how much their load impacts the energy consumption of VM migration. Then, we compared the accuracy of our model versus other state-of-the-art models that do not consider it. We quantify how much each actor's workload influences VM migration energy-wise. Our results show an improvement up to 24% in accuracy, showing that workload's impact on VM migration cannot be ignored when predicting its energy consumption. As a result, employing our model can significantly improve more energy efficiency focused VM consolidation decisions. For example, one may think not to consolidate a VM with an high dirtying ratio to a host that is running a lot of CPU intensive workloads since, as shown in Figure 7, this is going to increase the energy consumption of VM migration. The other models considered in this work do not take into account impact of the load running on the target host and therefore, may not be able to provide the same accuracy in predictions. Such a model could also be easily integrated in Cloud simulators to provide more accurate estimation of energy consumption in data centres. We plan to extend this work by also considering the impact of network-intensive workloads.

Acknowledgements

This work is supported by the joint Austrian Science Fund (FWF) project TRP 237-N23: Workflows on Manycore Processors, and by the European Union under the project 644179: dEcentralized repositories for traNsparent and efficienT vIrtual maChine opErations (ENTICE) and the COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS).

References

[1] Meikel Poess and Raghunath Othayoth Nambiar. Energy cost, the key challenge of today's data centers: A power consumption analysis of tpc-c results. *Proc. VLDB Endow.*, 1(2):1229–1240, August 2008.

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

[3] Qiang Huang, Fengqian Gao, Rui Wang, and Zhengwei Qi. Power consumption of virtual machine live migration in clouds. In *CMC '11*, pages 122 –125, april 2011.

[4] Haikun Liu, Cheng-Zhong Xu, Hai Jin, Jiayu Gong, and Xiaofei Liao. Performance and energy modeling for live migration of virtual machines. HPDC '11, pages 171–182. ACM, 2011.

[5] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. NSDI'05, pages 273–286. USENIX Association.

[6] Wei Deng, Hai Jin, Xiaofei Liao, Fangming Liu, Li Chen, and Haikun Liu. Lifetime or energy: Consolidating servers with reliability control in virtualized cloud datacenters. CLOUDCOM '12, pages 18–25. IEEE Computer Society, 2012.

[7] Jen-Cheng Huang, Hsien-Hsin S. Lee, and Mohammad M. Hossain. Migration energy-aware workload consolidation in enterprise clouds. CLOUDCOM '12, pages 405–410. IEEE Computer Society, 2012.

[8] Richa Sinha, Nidhi Purohit, and Hiteishi Diwanji. Energy efficient dynamic integration of thresholds for migration at cloud data centers. *IJCA Special Issue on Communication and Networks*, (1):44–49, 2011.

[9] Anton Beloglazov and Rajkumar Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurr. Comput. : Pract. Exper.*, 24(13):1397–1420, September 2012.

[10] William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya. Cost of virtual machine live migration in clouds: A performance evaluation. CloudCom '09, pages 254–265. Springer-Verlag, 2009.

[11] S. Akoush, R. Sohan, A Rice, AW. Moore, and A Hopper. Predicting the performance of virtual machine migration. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*, pages 37–46, Aug 2010.

[12] Akshat Verma, Gautam Kumar, Ricardo Koller, and Aritra Sen. Cosmig: Modeling the impact of reconfiguration in a cloud. In *Proceedings of the 2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, MASCOTS '11, pages 3–11. IEEE Computer Society, 2011.

[13] Takahiro Hirofuchi, Adrien Lèbre, and Laurent Pouilloux. Adding a Live Migration Model Into SimGrid, One More Step Toward the Simulation of Infrastructure-as-a-Service Concerns. In *IEEE CloudCom*, 2013.

[14] Kateryna Rybina, Abhinandan Patni, and Alexander Schill. Analysing the migration time of live migration of multiple virtual machines. In *CLOSER 2014 - Proceedings of the 4th International Conference on Cloud Computing and Services Science, Barcelona, Spain, April 3-5, 2014.*, pages 590–597, 2014.

[15] W. Dargie. Estimation of the cost of vm migration. In *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, pages 1–8, Aug 2014.

[16] A Strunk. Costs of virtual machine live migration: A survey. In *Services (SERVICES), 2012 IEEE Eighth World Congress on*, pages 323–329, June 2012.

[17] A. Strunk. A lightweight model for estimating energy cost of live migration of virtual machines. In *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, pages 510–517, June 2013.

[18] Qiang Huang, Fengqian Gao, Rui Wang, and Zhengwei Qi. Power consumption of virtual machine live migration in clouds. In *Communications and Mobile Computing (CMC), 2011 Third International Conference on*, pages 122–125, April 2011.

[19] H. Shirayanagi, H. Yamada, and K. Kono. Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks. In *Computers and Communications (ISCC), 2012 IEEE Symposium on*, pages 460–467, July 2012.

[20] Gong Chen, Wenbo He, Jie Liu, Suman Nath, Leonidas Rigas, Lin Xiao, and Feng Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *NSDI'08*, pages 337–350. USENIX Association, 2008.

[21] A-C. Orgerie, L. Lefevre, and J.-P. Gelas. Demystifying energy consumption in grids and clouds. In *Green Computing Conference, 2010 International*, pages 335–342, Aug 2010.

[22] Faraz Ahmad and T. N. Vijaykumar. Joint optimization of idle and cooling power in data centers while maintaining response time. *SIGARCH Comput. Archit. News*, 38(1):243–256, March 2010.

[23] Jyothi Sekhar, Getzi Jeba, and S. Durga. A survey on energy efficient server consolidation through vm live migration. *IJAET*, 5:515–525, November 2012.