Modelling Energy Consumption of Network Transfers and Virtual Machine Migration[☆]

Vincenzo De Maio, Radu Prodan

Institute of Computer Science, University of Innsbruck, Austria

Shajulin Benedict

St.Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, India

Gabor Kecskemeti

Institute for Computer Science and Control, MTA SZTAKI, Hungary

Abstract

Reducing energy consumption has become a key issue for data centres, not only because of economical benefits but also for environmental and marketing reasons. Therefore, assessing their energy consumption requires precise models. In the past years, many models targeting different hardware components, such as CPU, storage and network interface cards (NIC) have been proposed. However, most of them neglect energy consumption related to VM migration. Since VM migration is a network-intensive process, to accurately model its energy consumption we also need energy models for network transfers, comprising their complete software stacks with different energy characteristics.

In this work, we present a comparative analysis of the energy consumption of the software stack of two of today's most used NICs in data centres, Ethernet and Infiniband. We carefully design for this purpose a set of benchmark experiments to assess the impact of different traffic patterns and interface settings on energy consumption. Using our benchmark results, we derive an energy consumption model for network transfers.

[☆]This work is partially supported by the Indo-Austrian Project funded by the Austrian Science Fund and the Indian Department of Science and Technology.

Based on this model, we propose an energy consumption model for VM migration providing accurate predictions for paravirtualised VMs running on homogeneous hosts. We present a comprehensive analysis of our model on different machine sets and compare it with other models for energy consumption of VM migration, showing an improvement of up to 24% in accuracy, according to the NRMSE error metric.
Keywords: Energy consumption, benchmarking, network interface card, virtual machine migration

---

1. Introduction

Recently, Cloud computing has emerged as a new paradigm by which computational power is hosted on data centres by specialised providers and rented on-demand to the users based on their occasional needs. In doing this, providers are interested in maximising their profit. Since nowadays energy consumption has a big impact on their budget [1], they are inclined to maximise energy efficiency within their data centres. However, physical machines in data centres are often underutilised [2]. For this reason, one of the ways to increase energy efficiency is to increase their utilisation by mapping tasks on a subset of the data centre's machines and shut down the rest, a technique called *workload consolidation*. 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, we need prediction models for their energy consumption. Such models should take into account all the actors (e.g. VMs, physical hosts, network hardware) and activities (e.g. VM migration, powering down/off physical hosts) involved in the consolidation. Among all activities, *VM migration* is widely used when performing consolidation, because it provides the capability to move the state of running VMs between physical machines 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 con-

solidation. In recent years, several works modelled energy consumption of VM migration. For example, [3] developed a model considering VM CPU utilisation, [4] proposed a model based on VM dirtying rate and [5] a model based on VM size. However, both works either did not consider all the actors involved in the VM migration (e.g., source, target host, network infrastructure) or do not consider energy consumption due to network transfers using different software stacks. Since VM migration moves the state of a VM between two hosts over the network, we need an accurate energy model for network transfers, on top of which we will build a model for VM migration.

Many works [6, 7, 8] address specific hardware components such as CPUs, storage, and memory, but few of them focus on network transfers. In the networking area, existing works investigate energy-saving techniques like sleeping and rate adaptation [9] with focus on routers and switches [10] or on MPI parallel scientific applications [11]. Several works like [12] focused on the energy consumption of network transfers in message passing models, but few investigated it at the *software level*, comprising their complete stacks with different power characteristics and their impact on the application. Since data centres often install multiple Network Interface Cards (NICs) on each node, we believe that investigating and comparing them at the software level has high potential to enhance the energy efficiency of applications on Cloud infrastructures.

In this paper, we first investigate the main factors influencing the energy consumption of the software stack of the two mostly used networks in data centres: *Ethernet* and *Infiniband*. Our goal is to model their energy consumption at the application software level (not at the hardware level), considering all components involved in the network transfers (CPU, RAM, I/O, and NIC). For this purpose, we design a set of network-intensive benchmarks that emulate a wide spectrum of possible application parameters such as transfer size, number of simultaneous transfers, payload size, communication time, and traffic patterns. We focus on homogeneous nodes and on data transfers running over the TCP transfer protocol, because it is the most pervasive one according to [13]. We execute the benchmarks on machines equipped with NICs belonging to the

two families and compare their energy consumption. We do not consider energy consumption of routers and switches, because it has been proven to be constant regardless of the traffic amount [14]. Finally, we derive network transfer's energy consumption models for each network software stack.

Based on the developed network energy model, we introduce an energy consumption model for VM migration. We aim to improve the precision of existing models by (1) employing our model for network transfer and (2) considering a wider number of actors involved in this activity. We also consider the impact of different types of workload on energy consumption of VM migration. First of all, we identify the actors mostly involved in this activity. Then, we analyse the impact of VM migration on energy consumption of each actor considering different workloads. We focus on CPU and memory-intensive workloads that represent the most common and energy-impacting loads in data centres. In doing this, we identify the different phases of VM migration from an energy point of view and model the consumption of each actor over each phase. We target the Xen virtualization platform used by many commercial Clouds today such as Amazon EC2[1]. 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, since our measurements show that network-intensive workloads do not have a big impact on VM migration. We build our model on measurements taken on two different sets of machines with different architectures from a private Cloud. Then, we experimentally evaluate the impact of different workloads on energy consumption by measuring the energy consumption on each actor involved in VM migration while running benchmarks purposely designed to stress different components (e.g. CPU, memory). Based on the collected energy measurements for each selected component, we set as acceptance criterion for our model a normalised root mean square error (NRMSE) lower than in other other state-of-art models. Finally, we compare our results

[1] `http://www.citrix.com/global-partners/amazon-web-services/overview.html`

with three state-of-art models and perform investigations on a different subset of machines showing its applicability for diverse configurations.

1.1. Our contribution

The paper is an extension of our work [15], published in the UCC 2014 proceedings. The paper is organised as follows: first, we analyse related work in Section 2. Then, in Section 4.4 we introduce our network model. First, we describe the employed network hardware, then we identify the energy impacting factors for network transfers. Once identified the most energy-impacting factors, we design network-intensive benchmarks to evaluate the energy efficiency of the selected network software stacks. Finally, we use the benchmarks results to develop a model of energy consumption of network transfers.

As an extension of our previous work, we use the previously developed network model to develop a model of energy consumption of VM migration in Section 5. Since VM migration involves also CPU and memory, we extend the previously developed network model in order to consider also these other parameters. First, we identify the actors involved in VM migration and the migration energy phases. Then, we develop a model for VM migration by extending the previously developed network transfer model. Finally, we compare the obtained model with other existing models, showing that our model can reach an higher accuracy in different scenarios, and we conclude our paper in Section 6.

2. Related Work

Energy aware networking. Many works exploit network awareness to save energy, with focus on routing equipment and algorithms: [16, 17, 18]investigate energy-aware allocation of resources in Clouds considering network topology. Works like [19, 20] address the problem from the routing point of view. Techniques exploiting rate adaptation are explored for data centre networks in [21, 22]. Complementary to these works, we focus on the energy consumption from the perspective of software application, including not only the NICs, but also the other components involved in network transfers.

Network energy modelling. The first studies on network energy consumption [23, 10] focuses on energy consumption of routers, switches and hubs and do not take into account energy consumed by the application for data transfer. In [24] a energy consumption model for network equipment and transfers for large-scale networks, based on transfer time and bandwidth, is introduced. We propose here a complementary model for network transfers considering different NICs and more parameters. Works like [9] consider only transfer time when building a model for network transfers. In our work, we consider additional factors.

VM migration. One of the first works about VM migration is [25], but it does not take into account the energy consumption of this process. Other works such as [26, 27] investigate the advantages of using VM migration to achieve energy savings in data centers, but do not consider its own energy consumption. However, it focuses on the total energy consumed and does not highlight which consumption is related just to the network transfer. Moreover, this model makes a simplistic assumption that two nodes involved in a network operation consume the same energy, which may not be true for some NICs. Live VM migration has been proposed by [25] for Xen hypervisor. Since then, it has been implemented in many popular hypervisors, such as Xen, KVM and VMWare. Many works like [28, 29] 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 [30, 31] focused on the cost of live migration for cloud data centres, but they considered only performance and did not take energy consumption into account. Further works like [32] implement model for VM migration in Cloud simulator, but they do not provide models for its energy consumption. First investigations about energy consumption of VM migration have been done by [33]. One of the first works who modelled at the same time energy and performance of live migration is [4], that 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. Moreover, it 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 [5]. Another model has been proposed by [34], but it considers only CPU load. In this work, we consider the workload of each actor involved in the migration process and extract a more accurate model for VM migration. In this work we also consider energy consumption for non-live migration.

3. Preliminaries

In this section we are going to explain the basics concepts regarding network and VM migration.

3.1. Network hardware

We choose in our work the Ethernet and Infiniband NICs because they are to the best of our knowledge the most used interconnection technologies used in data centres. While communications running on Ethernet use the implementation of TCP/IP provided by the operating system, Infiniband software stack relies on kernel-bypass mechanisms and on RDMA-based capabilities. Such capabilities have a different impact on energy consumption. Therefore, comparing these two software stacks may give interesting insights about energy consumption of network transfers.

3.1.1. Ethernet

Ethernet is the most popular local-area network technology, defining several protocols which refer to the *IEEE 802.3* standard using four data rates: (1) 10 Mbps for 10Base-T Ethernet, in IEEE 802.3, (2) 100 Mbps, also called Fast Ethernet, in IEEE 802.3u, (3) 1000 Mbps, also called Gigabit Ethernet, in standard IEEE 802.3z, and (4) 10-Gigabit, also called 10 Gbps Ethernet, in standard IEEE 802.3ae. We focus on Gigabit Ethernet because, along with the newer 10-Gigabit, it is the most used interconnection technology in data centres. The minimum frame size for Gigabit Ethernet (1000Base-T standard) is 520 bytes, while the Maximum Transmission Unit (MTU) is 1500 bytes.

3.1.2. Infiniband

Infiniband is a popular switch-based point-to-point interconnection architecture that defines a layered hardware protocol (physical, link, network, transport), and a software layer to manage the initialisation and the communication between devices. Each link can support multiple transport services for reliability and multiple virtual communication channels. The links are bidirectional point-to-point communication channels that can be used in parallel to achieve higher bandwidth. Infiniband offers a bandwidth of 2.5Gbps in its single data rate version used in our work for comparison with Gigabit Ethernet. TCP/IP communications are mapped to the Infiniband transport services through IP over Infiniband (IPoIB) drivers provided by the operating system. An Infiniband NIC can be configured to work in two operational modes.

Datagram is the default operational mode of IPoIB described in RFC 4391 [35]. It offers an unacknowledged and connectionless service based on the unreliable datagram service of Infiniband that best matches the needs of IP as a best effort protocol. The minimum MTU is 2044 bytes, while the maximum is 4096 bytes.

Connected mode described in RFC 4755 [36] offers a connection-oriented service with a maximum MTU of 2GB. Using the connected mode can lead to significant benefits by supporting large MTUs, especially for large data transfers.

Setting Infiniband in one of these two modes will result in mapping a TCP communication on a different transport service. For this reason, we will measure the energy consumption of an Infiniband network transfer in both modes.

3.1.3. 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 in: (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 down time of VMs during migration in four steps: (1) moving the VM state from source to target host

while the VM performs its normal operations, (2) updating the state of the target VM with the modifications made on the source while transferring the state, (3) repeating step (2) until a termination criteria is reached (e.g. modifications under a given threshold or maximum number of copies reached), (4) suspending the VM and transferring the last modifications to the target, and (5) resuming the VM on the target when its state is consistent with the source.

3.1.4. Actors

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

- *Consolidation manager* constantly monitors the load of the data centres, selects the VM to be migrated and the target host, and finally initiates the migration. Afterwards, it returns to its previous operation;
- *Migrating VM* from the source to the target host, which also runs the services used by the customers of the data centre;
- *Source host* running the migrating VM, establishes first a connection with the target to communicate the intention of starting a VM migration;
- *Target host* designated as destination for the migrating VM. It provides the resources necessary for running the migrating VM;
- *Network* refers to the underlying communication infrastructure responsible for connecting the actors and for supporting the VM state transfers.

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. Concerning networking, we consider only the energy consumption consumed by the hosts for transferring the VM state. We do not consider the energy consumption of routers, switches and the underlying network infrastructure because according to our studies they are going to affect VM migration energy only when the

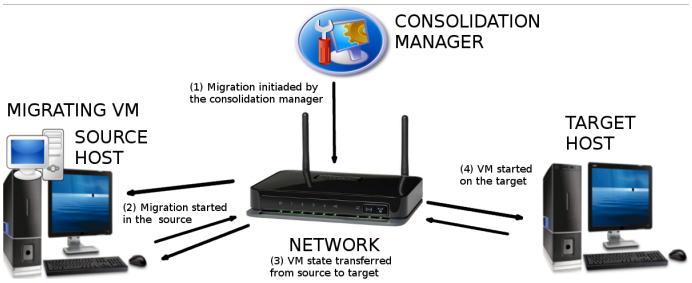


Figure 1: Overview of the migration process.

network is highly loaded. However, when there is a lot of network traffic between two hosts, it is unlikely that the consolidation manager will issue a migration between the two hosts, due to its big drawbacks on VMs performance.

4. Network Energy Modelling

4.1. Benchmarking Methodology

In this section we describe the benchmarking methodology for evaluating the energy consumption of the NIC software stacks. We first outline the impacting factors and then present the benchmarks and the evaluation metrics.

4.1.1. Energy-impacting factors

We describe the main factors affecting the energy consumption of network transfers according to our studies.

Time this parameter must be considered since the longer a network transfer, the more energy it consumes.

Transport protocol affects energy consumption because it defines the way in which transfers are performed. It defines how application layer's effective data are encapsulated. Such encapsulation inherently affects the NIC's operational mode and the amount of transferred data. While there exist many transport protocols (e.g. TCP, UDP, RSVP, SCTP), we only focus our analysis on TCP (the most pervasive one) due to space limitations.

Per-packet payload size is the real data transmitted with a single packet, juxtaposed to a header that makes the communication possible. The payload size depends on many factors such as protocol configuration, physical layer MTU, maximum segment size (MSS, representing the largest amount of data that can be sent in a single packet) on TCP, and other application characteristics (e.g. some applications require frequent exchange of small packets). Payload size has an impact on time, since a smaller payload size implies a higher number of packets and thus, more headers to process.

Number of connections to the NIC, typically shared among multiple applications that simultaneously send and receive data. With an increasing number of connections, one could experience a higher energy consumption due to the overhead introduced by their arbitration.

Traffic patterns of different types generated by network-centric applications as showed in [37], characterised by the inter-arrival time of packets.

4.1.2. Benchmarks

We investigate each factor through six benchmarks, all running on TCP.

BASE investigates the impact of network transfers on energy consumption by transferring a fixed amount of data using sockets without any specific tuning.

PSIZE investigates whether the NIC energy consumption is related to the payload size under two premises: (1) *PSIZE-DATA* determines the impact of payload size on energy efficiency independent of the data size by repeatedly transferring a fixed amount of data while varying the maximum payload size, and (2) *PSIZE-TIME* performs a maximum payload size evaluation with a fixed transfer time by continuously transferring data until a timeout is reached.

n-UPLEX evaluates the energy consumption of NICs in full duplex (FD) mode, while handling multiple concurrent connections. We transfer a fixed amount of data using a varying number of FD connections on each machine.

PATTERN evaluates the effects of traffic patterns on energy consumption. We configure the data transmissions to be a succession of *burst* and *throttle* intervals, representing fixed time intervals in which the NICs are continuously

Tool	*Transfer data size*	*Transfer timeout*	*MSS setting*	*Disable buffering*[1]	*FD/HD connections*	*Concurrent connections*	*Variable burst*	*Variable throttle*
`ttcp` (v1.12)	✓	✗	✗	✓	✗	✗	✗	✗
`netperf` (v2.4)	✓	✓	✗	✓	✗	✗	✗	✗
`iperf` (v2.05)	✓	✓	✓	✓	✓	✓	✗	✗

[1] e.g. an option for setting the TCP_NODELAY

Table 1: Comparison of networking benchmarking/diagnosis tools.

communicating and idle, as depicted in Figure 2. For *PATTERN-B* we keep the throttle size constant and vary the burst size, while *PATTERN-T* we vary the throttle size keeping a constant burst size.

For the PSIZE benchmarks, we need to successively set the transferred data size and a transmission timeout, and to strictly control the packet size. This can be achieved by altering the MSS and by disabling any buffering algorithms. For the n-UPLEX benchmark, we need to configure the type of (FD/HD) connections and the number of simultaneous connections. Finally, the PATTERN benchmark requires the possibility to shape the communication patterns through variable burst and throttle intervals. In the next section, we are going to see how we implemented our benchmarks.

4.1.3. Nimble NEtwork Traffic Shaper

To configure the metrics of our study based on transfer data size and timeout, payload size, FD/half-duplex (HD) connections, connection concurrency, and transmission patterns, we analysed three of the most popular open-source network diagnosis and benchmarking tools: `ttcp`[2], `netperf`[3] and `iperf`[4]. Table 1 presents a comparison of the flexibility of these tools focused on the provided configuration options for the metrics relevant to our study. Since none of the analysed tools covers all configuration parameters needed, we designed the *Nimble NEtwork Traffic Shaper (NNETS)*, a versatile network traffic shaping

[2] `http://www.pcausa.com/Utilities/pcattcp.htm`
[3] `http://www.netperf.org/netperf/`
[4] `http://iperf.sourceforge.net/`

tool implemented in Python 2.7 using the standard socket API, publicly available under GNU GPL v3 license[5]. In addition to the custom design required for accommodating all studied configurations, the tool allows a proper instrumentation of network and energy metrics. We implemented it with a clear separation between data processing and networking operations in order to instrument only the relevant regions of code, excluding data staging and pre-/post- processing operations and ensuring that the measured energy consumption is strictly related to the network transfer.

4.1.4. Metrics

To evaluate software stacks' energy efficiency we employ five metrics:

- *Machine energy consumption* in Kilojoules (kJ) for each experiment;
- *Network energy consumption* in Kilojoules (kJ), computed as the difference between the machine's energy consumption during benchmarks' execution and its idle consumption. This metric includes the energy consumed by all the components of the network stacks involved in the network transfer, that we purposely include to have a more realistic metric related to the software application;
- *Average power* in Watts (W), defined as the ratio between network energy consumption and its execution time;
- *Energy per byte* in Nanojoules (nJ), defined as the ratio between the network energy consumption and the number of bytes transferred, which indicates how energy varies in relation to the size of data transfer;
- *Energy per packet* in Millijoules (mJ), defined as the ratio between the network energy consumption and the amount of packets transferred.

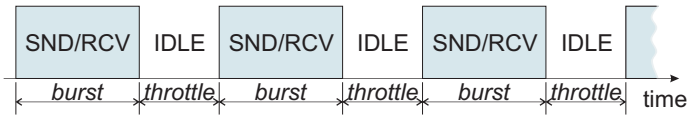


Figure 2: PATTERN benchmark (burst/throttle intervals).

Host	*CPU*	*Kernel*	*Gigabit NIC*	*Infiniband NIC*	*Gigabit switch*	*Infiniband switch*	dom0 kernel	Xen version
k12	4×	Linux	Broadcom	SDR Mellanox	Cisco	Mellanox MT47396	3.0.4	4.2
k13	Opteron 880	2.6.9-67	BCM5704	MT23108	Catalyst 3750	Infiniscale-III		

Table 2: Experimental hardware.

4.2. Experimental Setup

We employ two machines, both equipped with Infiniband and Gigabit Ethernet NICs, as specified in Table 7c. We set the MTU on all machines to 16382 bytes for the Infiniband NICs in connected mode, to 2044 bytes in datagram mode, and to 1500 bytes for the Gigabit Ethernet NICs. The machines are connected through two dedicated server-grade network switches to exclude the impact of external network traffic. For each NIC and connectivity mode, we run the benchmarks in three configurations (send, receive and n-uplex), namely: (1) ETH-SND/RCV, ETH for Gigabit Ethernet in send, receive and n-uplex; (2) IBC-SND/RCV, IBC for Infiniband connected in send, receive and n-uplex; and (3) IBD-SND/RCV, IBD for Infiniband datagram in send, receive and n-uplex. For the energy measurements, we use Voltech PM1000+ power analysers (with 0.2% accuracy) connected to the machines' AC side and capable of reading the power twice per second. For each benchmark, we select the input parameters to produce an execution time of at least 50 seconds, which allows us to have at least 100 readings in each execution. Table 3 summarises the experimental parameters. The *data* and *time* columns denote the termination condition of each benchmark experiment. When the data size is set, the experiment terminates after transferring the indicated amount of data (i.e. the session and transport overheads), while when the timeout is set, the experiment is termi-

[5]To be published at: `http://code.google.com/p/nnets/`

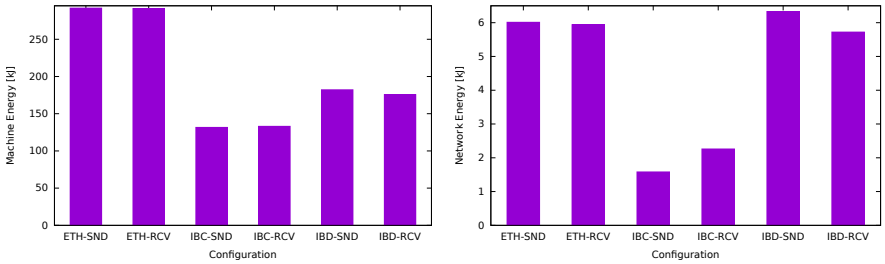
Benchmark	*Size* [GB]	*Time* [min]	*Payload* [% MTU]	*Connections*	*Burst* [ms]	*Throttle* [ms]
PSIZE-DATA	75	–	**30 − 100**	1 HD	–	–
PSIZE-TIME	–	5	**30 − 100**	1 HD	–	–
n-UPLEX	150	–	100	**1 − 8 FD**	–	–
PATTERN-B	11	–	100	1 HD	**1 − 10**	10
PATTERN-T	11	–	100	1 HD	10	**1 − 10**

Table 3: Benchmark summary with focus metric in bold.

nated after the indicated time. The *payload* indicates the size of the useful data in each packet, computed as a percentage of MTU minus 40 bytes (the size of IP and TCP headers), but for simplicity we denote it as "a percentage of MTU". The *connections* column indicates the number of concurrent connections through which the transfer is made. Finally, the *burst* and *throttle* represent the concrete time intervals of continuous activity and inactivity of the NICs. For the PSIZE benchmarks, we vary the maximum payload between 30% and 100% of the NICs' MTU. We also set the `TCP_NODELAY` flag to prevent packets smaller than MTU from being buffered. For PSIZE-DATA we set the data size to 75GB, while for PSIZE-TIME we set a timeout of 5 minutes. For the n-UPLEX benchmark, we transmit a fixed amount of data of 150GB (sending 75GB and receiving 75GB) over n FD connections. For both PATTERN benchmarks, we set the data size to only 11GB, as the studied traffic patterns considerably increase the transfer times. In the PATTERN-B benchmark, we keep the throttle size constant to 10 msand vary the burst size to 2, 4, 6, 8, and 10 ms. Conversely, for the PATTERN-T benchmark, we vary the throttle to 2, 4, 6, 8, and 10 ms with a constant burst size of 10 ms. We run each experiment for ten times, which ensures an average coefficient of variation of 0.053, and present the average of the results.

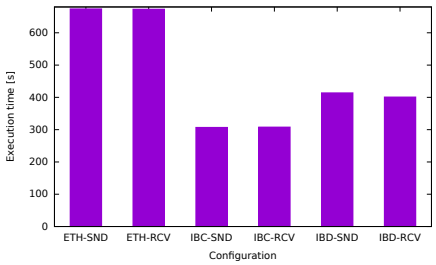
4.3. Experimental Results

In this section we present the results of our experiments.

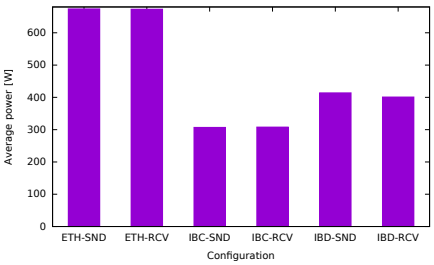


(a) Machine energy

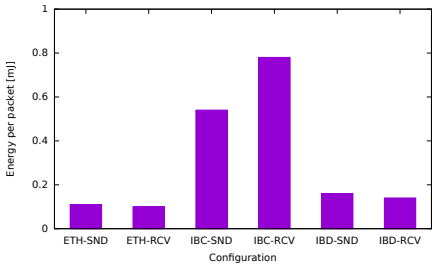
(b) Network energy



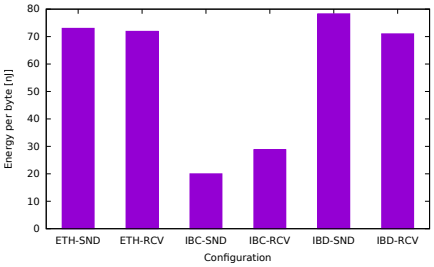
(c) Execution time



(d) Average power



(e) Energy per packet



(f) Energy per byte

Figure 3: Base benchmarks result

4.3.1. BASE

We observe in Figure 3 a considerable difference in energy consumption for running the BASE benchmark. The immediate finding is that transferring the same quantity of data over Infiniband in connected mode is more efficient in terms of energy and time. We can also observe that Infiniband's energy consumption significantly differs between sending and receiving operations: 30%

less energy for sending than receiving in connected mode, and 10% less energy for receiving compared to sending. It is also noteworthy that, even in this simple benchmark, the network energy consumption is between 1.58 and 6.33 kJ, that can be a significant percentage of energy consumption in a node with lower idle power consumption. The other metrics provide supplementary insight into these NICs' energy efficiency. Although it might appear that the Infiniband in connected mode is more energy efficient with the lowest average power in operation, this only holds true when the two communicating parties require large amounts of on-hand data to be transferred. When the communication is message centric and the volume of effective data is low, resulting in a high number of packets being transmitted, the Gigabit Ethernet NIC is the more energy-efficient choice, closely followed by Infiniband in datagram mode.

These preliminary findings hint that an energy efficient network communication depends on the nature of the traffic generated by the application. For data intensive traffic in applications such as data warehousing and content streaming or delivery, the more energy-efficient network is Infiniband configured in connected mode. On the other hand, for finer-grained traffic and real-time message exchanges such as low traffic databases and online games, Gigabit Ethernet is more efficient. The following experiments give further assessment of the energy consumption of the two networks with respect to traffic characteristics.

4.3.2. PSIZE

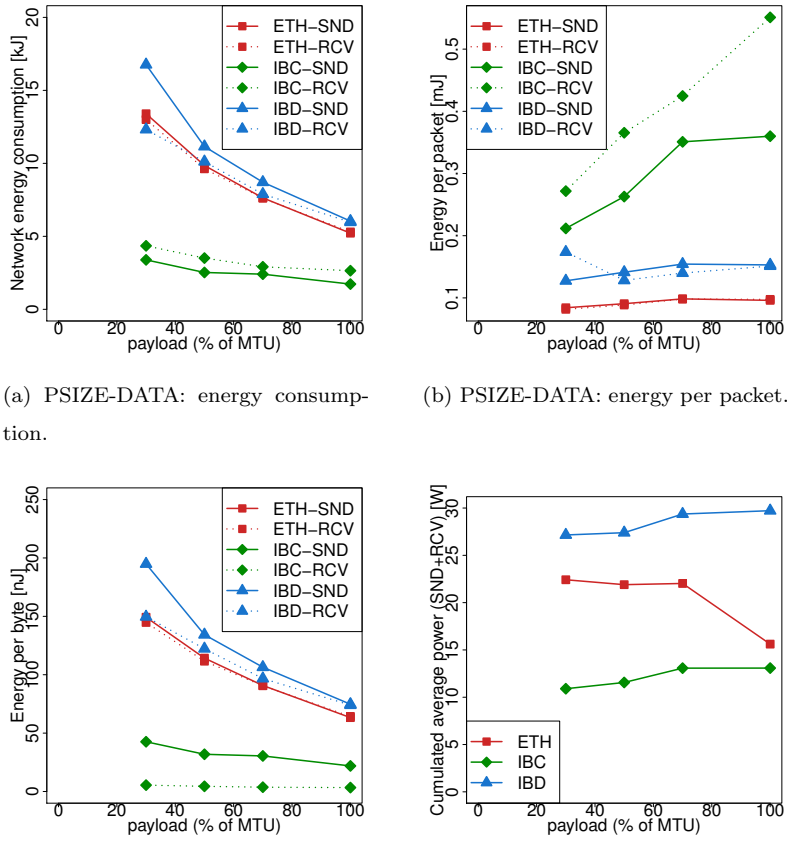
We begin with the PSIZE benchmark, focused on the influence of the payload size on networks' energy efficiency.

PSIZE-DATA. The results in Figure 4a show that the energy consumption of the software stacks of the studied NICs is inversely proportional to payload, the most efficient operational point being reached for the maximum payload. Also noteworthy is the significantly better scalability in terms of energy when employing Infiniband NIC in connected mode: 36% energy consumption increase for a 50% decrease in payload, versus 84% for Gigabit Ethernet and 79% increase

for Infiniband in datagram mode. Analysing the other metrics presented in Figure 4, we can identify in detail the energy-to-payload relation. Figure 4b suggests that, while for Infiniband in connected mode the energy consumption per transferred packet is proportional to its payload, it is relatively constant in the case of Infiniband in datagram mode and Gigabit Ethernet. Conversely, Figure 4c reveals a stronger inverse correlation between the payload and the energy consumption per transferred effective byte. The Infiniband in datagram mode and the Gigabit Ethernet NICs are affected in terms of energy efficiency by a payload decrease, the energy consumption per effective byte nearly tripling at a 30% of MTU payload. This behaviour is less severe for Infiniband in connected mode, the energy per byte doubling for a payload of 30% of MTU.

PSIZE-TIME. We present the resulting average power consumption in Figure 4d, each point representing the cumulated power for send and receive operations. The main finding of this experiment is that the energy consumption of both Infiniband and Gigabit Ethernet NICs is not exclusively correlated with running time. We observe that while Infiniband (regardless of its operational mode) consumes in average less power with lower payloads, Gigabit Ethernet is more power efficient at higher payloads. Further investigation revealed that Gigabit Ethernet's high power efficiency for larger payloads is likely due to driver optimisations, as we noticed a 32% decrease in CPU utilisation between the transfers with payloads set at 30%, respectively 100% of MTU. The CPU utilisation was constant for all Infiniband transfers in both modes.

To conclude, energy consumption of the networks is inversely proportional to the maximum payload size. Second, Gigabit Ethernet and Infiniband in datagram mode are better suited for lightweight, mixed traffic (with varying payload sizes), while Infiniband connected is by far the most energy efficient for non-fragmented traffic. Finally, network energy consumption is not exclusively time-related, thus one cannot optimise for time and expect proportional savings.



(a) PSIZE-DATA: energy consumption.

(b) PSIZE-DATA: energy per packet.

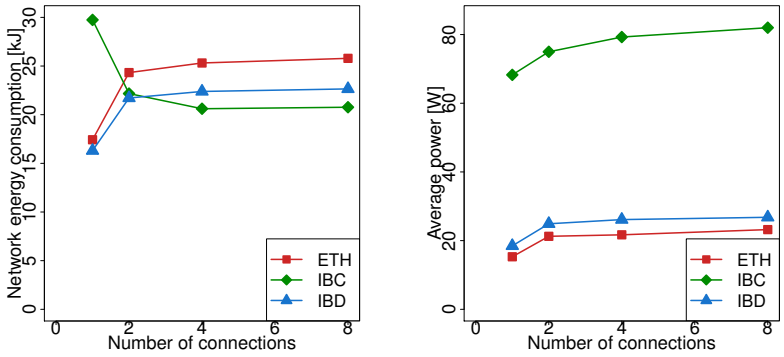
(c) PSIZE-DATA: energy per byte.

(d) PSIZE-TIME: Cumulated send and receive average power.

Figure 4: PSIZE benchmark results.

4.3.3. n-UPLEX

We observe in Figure 5a a considerable increase in the energy consumption of Gigabit Ethernet and Infiniband in datagram mode with more concurrent connections. The trend has a piecewise linear shape and is relatively similar for the power traces shown in Figure 5b. In contrast, Infiniband in connected mode shows a decreasing energy consumption with the increase in concurrent connections. Moreover, although Infiniband in connected mode consumes the least energy for transferring the fixed data amount for multiple connections,



(a) Network energy.

(b) Average power.

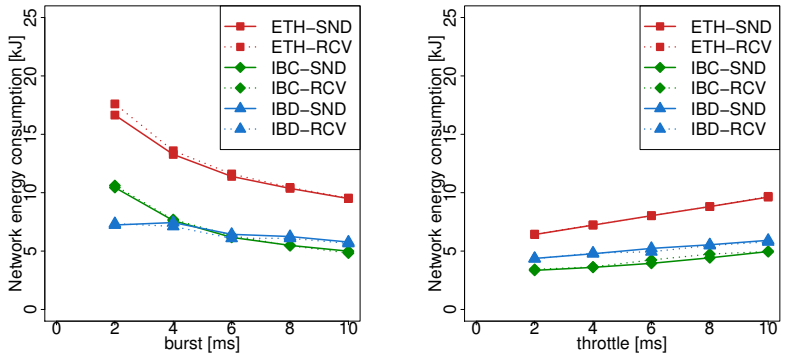
Figure 5: n-UPLEX benchmark results.

it is clearly exhibiting the highest average power consumption. This raises a question regarding the NICs' performance in terms of transfer bandwidth in this contention scenario. We present in Table 4 a comparison between the variation of the achieved bandwidth, consumed energy, and CPU utilisation between the two extreme cases studied: (1) the network contention case with eight concurrent FD connections and (2) the single FD connection. The results reveal a significant increase of 72% in bandwidth for the Infiniband connected, with a 19.1% average power increase. This variation of its power state with performance (in terms of bandwidth), is the reason of its energy efficiency. At the other end, Gigabit Ethernet exhibits the highest increase in energy consumption of almost 50% with only a marginal 2.5% increase in bandwidth. The considerable average power consumption increase in all cases stems from both (1) NICs requiring more power to handle the increased load and (2) increasing CPU overheads for managing multiple simultaneous connections. This observation is supported by the non-proportional energy consumption versus the CPU utilisation increase shown in Table 4. Finally, the increase of CPU utilisation for Infiniband in connected mode is 130.15% higher than the other two configurations due to the increased bandwidth requiring faster data preprocessing.

In summary, in a connection concurrency environment significant power con-

Metric	*Variation [%] (8 vs 1 connections)*		
	ETH	*IBD*	*IBC*
Bandwidth	+2.49	+4.39	+72.03
Energy	+45.80	+37.33	−31.03
Power	+49.43	+43.37	+19.11
CPU	+38.62	+38.23	+130.15

Table 4: Variation of relevant metrics with number of concurrent connections.



(a) PATTERN-B.

(b) PATTERN-T.

Figure 6: PATTERN benchmark results.

sumption penalties occur, the Infiniband in connected mode being the best choice in terms of energy efficiency. The increased power consumption is due to a higher NICs' power state and to processing overheads.

4.3.4. PATTERN

These two experiments study the energy consumption of the NIC software stacks for different communication patterns.

PATTERN-B. Figure 6a shows that Gigabit Ethernet is the least energy efficient for all studied burst intervals. For short burst intervals ($2 - 4$ms), Infiniband datagram is surprisingly more efficient consuming up to 44% less energy than in connected mode. For longer burst intervals, connected mode becomes better consuming 17% less energy.

PATTERN-T. Figure 6b shows a stable, monotonously increasing energy consumption with increasing throttle intervals. It is noteworthy that the energy consumption increases at different rates for the different NICs and operational modes: Gigabit Ethernet's consumption increases by 110J per ms of throttle, while Infiniband by 49J in datagram mode, and by 55J in connected mode. Although Infiniband connected is more energy efficient for the studied configurations, a basic extrapolation shows that for traffic patterns with throttle intervals higher than 50ms the datagram mode becomes the more energy efficient choice.

In conclusion, Infiniband in datagram mode shows the least variation in energy consumption with different (mixed or undetermined) transmission patterns, while Infiniband in connected mode exhibits a very good energy efficiency in a few particular cases (for long transmission bursts).

4.4. Network Energy Consumption Model

We model in this section the factors analysed in Section 4.1.1 that affect the network energy consumption. We believe that such a model would help scientists in more accurately predicting the energy consumption of network-intensive applications, as required for example by resource managers and schedulers. We decided to use regression analysis, that has been successfully used in previous energy prediction and modelling works [38]. We employ the NLLS regression algorithm. For extracting model parameters, we employ the data gathered from ten experimental runs. We assess the accuracy of our models using two metrics: (1) mean absolute error (MAE) and (2) root mean squared error (RMSE) that is also an absolute deviation metric, but more sensitive to large deviations. The difference between the two metrics is a measure of the variance in the individual deviations for all samples. We will also present a normalised value of RMSE (NRMSE) for metric-independent comparisons.

We model the energy consumption of a network transfer as:

$$E = \sum_{x \in \{\texttt{send},\texttt{receive}\}} \left(E_x(\texttt{DATA}_x, p_x, b_x, t_x) + O\left(c_x\right)\right), \tag{1}$$

	$\alpha_{\texttt{send}}$[µJ]	$\alpha_{\texttt{receive}}$[µJ]	$\beta_{\texttt{send}}$	$\beta_{\texttt{receive}}$	$\gamma_{\texttt{send}}$	$\gamma_{\texttt{receive}}$	$\mathcal{K}_{\texttt{send}}$[kJ]	$\mathcal{K}_{\texttt{receive}}$[kJ]	ϵ	ζ	MAE [kJ]	$RMSE$	$NRMSE$
ETH	73.5	71.3	19.71	21.57	0.59	0.58	0.35	0.35	733.14	-685.56	0.44	0.9	0.03
IBC	137.1	181.4	13.93	14.23	0.23	0.19	0.58	0.80	12.59	-0.21	0.82	2.62	0.09
IBD	97.9	69.0	4.13	3.96	0.22	0.16	2.37	2.16	99.52	-82.13	0.83	0.98	0.05

Table 5: Model parameters and error.

where $\texttt{DATA}_x$ is the number of bytes transferred, p_x the payload per packet, c_x the number of additional connections (for FD transfers), and b_x and t_x the size of burst and throttle intervals in ms. We calculate E_x as:

$$E_x = \alpha_x \cdot \frac{\texttt{DATA}_\texttt{x}}{p_x} + \frac{\beta_x}{b_x} + \gamma_x \cdot t_x + \mathcal{K}_x, \tag{2}$$

where $x \in \{\texttt{send}, \texttt{receive}\}$, a_x can be interpreted as the cost for sending, respectively receiving a packet, β_x and γ_x are the model parameters, and $\mathcal{K}_x$ is a hardware and driver-related constant for setting up a sending, respectively receiving connection. Regarding the overhead of multiple connections, since Gigabit and Infiniband datagram use the NICs in a different way compared to Infiniband connected, their arbitration of multiple connections will be different too. For this reason, we employ Equation 3 for both Gigabit and Infiniband datagram and Equation 4 for Infiniband connected:

$$O_{\texttt{datagram}}(c_x) = \log(\epsilon \cdot c_x + \zeta); \tag{3}$$

$$O_{\texttt{connected}}(c_x) = \epsilon \cdot c_x^{\zeta}, \tag{4}$$

where ϵ and ζ are the model parameters and $x \in \{\texttt{send}, \texttt{receive}\}$. Table 5 shows the model parameters along with the error, calculated over all the samples. The error is always below 9.4% which demonstrates a good accuracy.

5. Energy Modelling of VM Migration

In this section we build a model of VM migration based on the network transfer model.

5.1. Power Characteristics of VM Migration

In this section we provide an overview of the power characteristics of VM migration. We already described the VM migration process and depicted the actors involved in this process in Section 3. Now, we investigate the workloads impacting VM migration energy consumption. Finally, we identify the phases that occur during a migration.

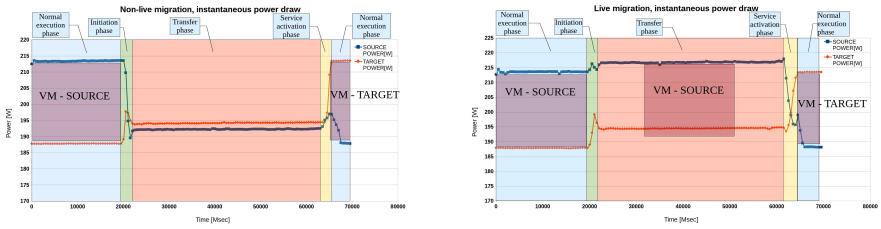
5.1.1. Workloads

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

Although there may be different kind of workloads running in a data center (e.g. CPU-intensive, memory-intensive, network-intensive, or mixed), we focus in the following on the CPU and memory-intensive ones because they mostly impact the VM migration process. Table 6 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 host are fully loaded because of the CPU shared between the workload running on the machines and on the migrating VM. If the migrating VM is running a memory-intensive workload that continuously updates RAM locations, this will highly impact performance of a live migration since several transfers are needed to achieve a consistent state between the source and the target. 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 an high memory dirtying ratio (i.e. a high percentage of memory pages marked dirty over a given amount of time).

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



(a) Non-live migration.

(b) Live migration.

Figure 7: Energy consumption phases of non-live and live migration.

5.1.2. Migration energy phases

As we discussed in the previous sections, both live and non-live migration go through different phases with different energy-wise behaviour for each actor, and highly influenced by their hosted workloads. In this section, we identify the migration phases from an energy point of view by collecting and analysing power traces of a VM migration as shown in Figure 7 and described next.

Normal execution. During this phase, the VM and the other actors are performing their normal operation before a migration decision is taken. We assume that the power consumption over this phase is constant, since it has no influence on VM migration. We describe it here anyway, for clarity reasons.

Initiation. This phase starts when a migration is issued and ends when the target host is ready to receive the VM state. Regarding the source host, we will experience a strong decrease in power consumption because of the VM suspension in case of non-live migration, while for live migration there will be a peak for saving the VM state and sending it to the target. On the target host,

the peaks in power consumption are due to checking of resource availability and sending of acknowledgement to the source that a migration can start.

Transfer. During this phase, all the data needed by the VM is transferred over the network from the source to the target host. For non-live migration, VM suspension has limited influence on the power consumption which is only influenced by the exchanged VM state data. For live migration, we experience an additional consumption in the source that has to keep track of the modifications to the VM state.

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 owned by the VM in the case of non-live migration, while for live migration the VM needs to be first shut down. Finally, each actor returns to the normal execution phase.

5.2. Model

In this section, we model the energy consumption of each migration phase described in the previous section. The energy consumption of the complete VM migration process will be the sum of the energy consumption of each phase.

5.2.1. Migration model

In this section we formally define a VM migration transferring the state of a migrating VM v from a source host `S` to a target host `T`. As we saw in Section 5.1.2, VM migration goes through different energy phases. Therefore, we define for each migration m_s as the instant when the migration starts, t_s and t_e the time instances when the transfer phase of the migration starts, respectively ends, and m_e as the instant when the migration ends. The time interval between m_s and t_s is the initiation and the one between t_e and m_e is the activation phase.

5.2.2. Resource utilisation model

According to our analysis in Section 5.1, the most impacting actors for VM migration are the source and target hosts `S` and `T` and the migrating VM v. In

this section, we present a model for resource utilisation of the selected actors to which 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 6, the most impacting parameters on migration are: (1) CPU utilisation of the source $\texttt{CPU}(\texttt{S}, t)$ and target $\texttt{CPU}(\texttt{T}, t)$ hosts at the instant t and CPU utilisation $\texttt{CPU}(v, t)$ of the migrating VM v at instant t, (2) memory dirtying ratio $\texttt{DR}(v, t)$ of the VM v at instant t, (3) the memory $\texttt{MEM}(v)$ allocated to the migrating VM v, and (4) the 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 $\texttt{DR}(v, t) = 0$. 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 $\texttt{CPU}(v, t)$ for 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:

$$\texttt{CPU}(h, t) = \texttt{CPU}_{\texttt{VMM}}(\mathcal{V}(h, t)) + \sum_{v \in \mathcal{V}(h,t)} \texttt{CPU}(v, t) + \texttt{CPU}_{\texttt{migr}}(h, t), \tag{5}$$

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

5.2.3. Energy model

In this section we describe the model for energy consumption of VM migration. For each physical host $h \in \{\texttt{S}, \texttt{T}\}$, this energy consumption is the integral of the migration power P_{migr} over the migration time $[m_s, m_e]$:

$$E_{migr}(h, v) = \int_{m_s}^{m_e} P_{migr}(h, v, t)\, dt, \tag{6}$$

where $P_{migr}(h, v, t)$ is the sum of the power consumed over the three energy phases ($P_{(\texttt{i})}(h, v, t)$ for initiation, $P_{(\texttt{t})}(h, v, t)$ for transfer and $P_{(\texttt{a})}(h, v, t)$ activation) identified in Section 5.1.2. Integrating each one of this values over the migration time we obtain the energy consumption over each phase, respectively $E_{(i)}(h, v)$, $E_{(t)}(h, v)$ and $E_{(a)}(h, v)$. Summing these values we obtain energy

consumption of VM migration E_{migr}:

$$E_{migr}(h, v) = E_{(i)}(h, v) + E_{(t)}(h, v) + E_{(a)}(h, v). \tag{7}$$

Depending on whether the host is acting as source or target, some parameters can be ignored. For example, we do not need to consider the resource utilisation of the migrating VM on the target during the initiation phase, as the VM is still running on the source host. Finally, the energy consumption during each phase also changes according to the migration approach and the VM workload, as we analyse in the next sections.

5.2.4. Non-live migration

In this section, we model the energy consumption of the three phases of a non-live migration.

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, because the VM will still be running over this phase:

$$P^{(\mathtt{i})}_{\mathtt{nonlive}}(h, v, t) = \quad \alpha_{(\mathtt{i})}(h) \cdot \mathtt{CPU}(h, t) + \beta_{(\mathtt{i})}(h) \cdot \mathtt{CPU}_{\mathtt{vm}}(v, t) + \mathcal{C}_{(\mathtt{i})}(h), \tag{8}$$

where $\alpha_{(\mathtt{i})}(h)$, $\beta_{(\mathtt{i})}(h)$ model the relationship between the CPU usage of the two hosts and of the migrating VM to the power consumption. We approximate the power consumption with a linear function, as done by [39]. $\mathcal{C}_{(\mathtt{i})}(h)$ include the power consumption for establishing a connection between the two hosts. On the source host, it also includes the power consumption for suspending the VM.

Transfer phase. In the transfer phase, the state of the VM is transferred from source to the target host. Therefore, we will use a simplified version of the network transfer model developed in Section 4.4 assuming a linear relationship between the network bandwidth and the energy consumption. Since the transfer is contiguous and uses the maximum packet size, we consider neither the

throttle/burst sizes nor the packet size. We also ignore the number of concurrent connections, since the data transfer is not performed in parallel. In this phase, we also consider the CPU usage on both hosts proportional to the power consumption, but ignore the resource utilisation of the suspended VM:

$$P^{(\mathtt{t})}_{\mathtt{nonlive}}(h,t) = \alpha_{(\mathtt{t})}(h) \cdot \mathtt{CPU}(h,t) + \beta_{(\mathtt{t})}(h) \cdot \mathtt{BW}(\mathtt{S},\mathtt{T},t) + \mathcal{C}_{\mathtt{t}}(h), \tag{9}$$

$\alpha_{(\mathtt{t})}(h)$ models the linear relationship between power and CPU usage, $\beta_{(\mathtt{t})}(h)$ the relationship between bandwidth and power, and $\mathcal{C}_{\mathtt{t}}(h)$ the power consumption for moving the state of the migrating VM to the target host. We expect the latter to be higher in the target host than in the source because it also needs to write the VM state in the RAM.

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

$$P^{(\mathtt{a})}_{\mathtt{nonlive}}(h,v,t) = \alpha_{(\mathtt{a})}(h) \cdot \mathtt{CPU}(h,t) + \beta_{(\mathtt{a})}(h) \cdot \mathtt{CPU}_{\mathtt{vm}}(v,t) + \mathcal{C}_{(\mathtt{a})}(h) \tag{10}$$

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.

Live migration. For the live migration, we do not expect any difference in the initiation and activation phases compared to the non-live case. We therefore focus in the following on the transfer phase.

Transfer phase. The main difference to 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:

$$\begin{aligned} P^{(\mathtt{t})}_{\mathtt{live}}(h,v,t) = \quad & \alpha_{(\mathtt{t})}(h)\cdot \mathtt{CPU}(h,t) + \beta_{(\mathtt{t})}(h)\cdot \mathtt{BW}(\mathtt{S},\mathtt{T},t) + \\ + \quad & \gamma_{(\mathtt{t})}(h)\cdot \mathtt{DR}(v,t) + \delta_{(\mathtt{t})}(h)\cdot \mathtt{CPU}_{\mathtt{vm}}(v,t) + \mathcal{C}_{(\mathtt{t})}(h), \end{aligned} \tag{11}$$

where $h \in \{\mathtt{S},\mathtt{T}\}$, $\alpha_{(\mathtt{t})}(h)$ models the linear relationship between power and CPU usage, $\beta_{(\mathtt{t})}(h)$ the relationship between power and bandwidth, $\mathtt{DR}(v,t)$ the percentage of pages marked as dirty at the instant t, $\gamma_{(\mathtt{t})}(h)$ the linear relationship between the dirtying ratio and power consumption, $\delta_{(\mathtt{t})}(h)$ the linear relationship between the migrating VM's CPU usage and its power consumption, and $\mathcal{C}_{\mathtt{t}}(h)$ the power consumption for moving the state of the migrating VM to the target host. Finally, we define the term $\mathtt{DR}(v,t)$ as:

$$\mathtt{DR}(v,t) = \frac{\mathtt{DIRTYPAGES}(v,t)}{\mathtt{MEM}(v)}, \tag{12}$$

where $\mathtt{DIRTYPAGES}(v,t)$ is the number of pages marked as dirty at the instant t in the memory of VM v and $\mathtt{MEM}(v)$ is the VM memory size in pages. We expect a linear relationship between dirtying ratio and power consumption due to the increased contention on memory.

5.3. Experimental methodology

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

5.3.1. Experimental design

Our experimental settings are summarised in Table 7a, and the VM and hardware configurations in Tables 7b and 7c. We use Xen version 4.2.5, including both `xm` and `xl` toolstacks configured to perform the live and non-live migrations between two hosts and deploy two machines and a networking switch, as specified in Table 7c. We perform the experiments on two sets of machines (`m01-m02` and `o1-o2`) with different CPU and Gigabit NIC architectures. We do not include experiments using Infiniband NICs delivering similar results for brevity reasons. For each experiment, we employ paravirtualized VMs mostly

encountered in modern data centres as they ensure a nearly-native performance. For the migrating VMs, we chose a 4 GB VM size for the RAM memory which gives us a long enough migration time to clearly identify energy consumption phases.

According to our analysis in Table 6, the highest impact on VM migration have the CPU-intensive workloads running on source or target hosts and the memory-intensive workloads running on the migrating VM. Therefore, we design two families of experiments: *CPULOAD* and *MEMLOAD*.

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

1. *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 that there is some multiplexing of them.
2. *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 with small communication and synchronisation overheads. Concerning the VM configuration, among the instances described in Table 7b we select the `load-cpu` and the `migrating-cpu` type. 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 we need to these instances to increase the load by 25% in every step.

MEMLOAD. We study the effect of the dirtying ratio (see Equation 12) of the VM workload on migration, according to our analysis in Table 6. To compare the impact of the memory-intensive workloads with the CPU-intensive ones, we design experiments involving CPU-intensive workloads running on both source and target, as follows:

1. *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.
2. *MEMLOAD-SOURCE* investigates how live migration is impacted by (1) CPU-intensive workloads 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.
3. *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 $\texttt{DR}(v,t) = 0$. 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 migrating VM (see Table 7b).

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,	32GB	Broadcom	Cisco Catalyst	4.2.5
m02	dual threaded)		BCM5704	3750	
o1	40 (20×Xeon E5-2690v2,	128GB	Intel	HP	4.2.5
o2	dual threaded)		82574L	1810-8G	

(c) Hardware configuration.

Table 7: Experimental setup.

5.3.2. Energy measurement methodology

We employ two Voltech PM1000+[6] power measurement devices to the connected to the AC side of the source and target hosts, measuring the power 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.

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

We say that the power consumption of the host stabilizes when we have twenty consecutive power values with a difference lower than 0.3%, that is below our measurement device's accuracy. Moreover, each experiment is repeated 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 5.1.2 and 5.2.3), and the total migration energy as the sum of the three metrics. In addition, also measure the CPU and memory consumption during each migration using the `dstat` tool and average the values of all executions.

5.4. Experimental Results

In this section, we show the results of our experiments described in Section 5.3. 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.

5.4.1. CPULOAD-SOURCE

The results for this experiment displayed in Figures 8a and 8b 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. We clearly see that on the source host (Figure 8a) 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 8b), 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.

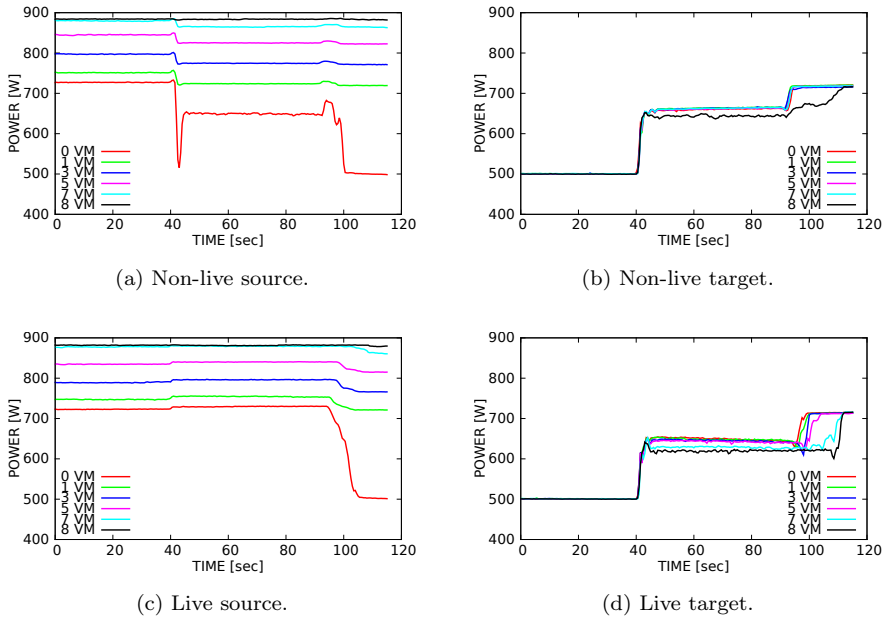
For live migration (Figures 8c and 8d), we observe an increased power consumption over the transfer phase due to the running VM because of: (1) the additional power consumption for network transfers and (2) the increased CPU usage of the virtualization software to handle the live migration. Concerning the source host, we notice a constant power consumption in case of CPU multiplexing, for the same reason as in Figure 8a.

For power consumption on the target host (Figure 8d), 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, 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 8b. We also notice a strong difference in power consumption before and after migration in the 25% load scenario because the power draw of the source host returns back to idle after the migration.

We conclude that CPU-intensive workloads impact 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.

5.4.2. CPULOAD-TARGET

For this experiment, we observe fist in Figure 9a 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 9b, we can notice (1) a small increase in power draw 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 5).



(a) Non-live source.

(b) Non-live target.

(c) Live source.

(d) Live target.

Figure 8: CPULOAD-SOURCE results.

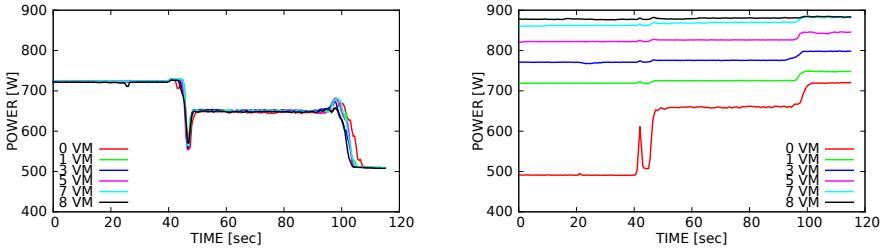
For the live migration (Figure 9c), we notice for the source host a small increase in power consumption over the transfer phase due to: (1) the network transfer of the VM state and (2) the CPU increase for handling the migration. We do not notice any impact of the target load on this host except for 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 9d, we see similar trends to the non-live migration except that: (1) the power draw is slightly lower in the transfer phase and (2) the live migration takes at least 60 seconds longer. Since this tendency is present also in the idle target case, it seems mostly related to hardware configuration than host load.

5.4.3. MEMLOAD-VM

For the MEMLOAD-VM experiment, we observe in Figures 10a and 10b 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

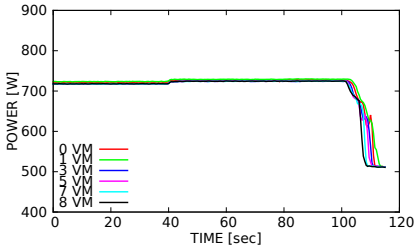
Host	*Initiation*				*Transfer*				*Activation*			
	$\alpha_{(\mathrm{i})}$	$\beta_{(\mathrm{i})}$	$\mathcal{C}^1_{(\mathrm{i})}$	$\mathcal{C}^2_{(\mathrm{i})}$	$\alpha_{(\mathrm{t})}$	$\beta_{(\mathrm{t})}$	$\mathcal{C}^1_{(\mathrm{t})}$	$\mathcal{C}^2_{(\mathrm{t})}$	$\alpha_{(\mathrm{a})}$	$\beta_{(\mathrm{a})}$	$\mathcal{C}^1_{(\mathrm{a})}$	$\mathcal{C}^2_{(\mathrm{a})}$
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 8: Coefficients for non-live migration.

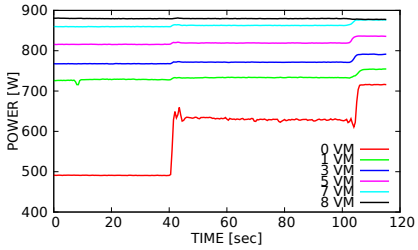


(a) Non-live source.

(b) Non-live target.



(c) Live source.



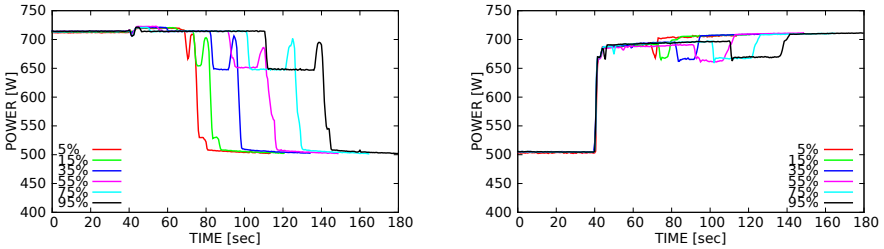
(d) Live target.

Figure 9: CPULOAD-TARGET results.

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.

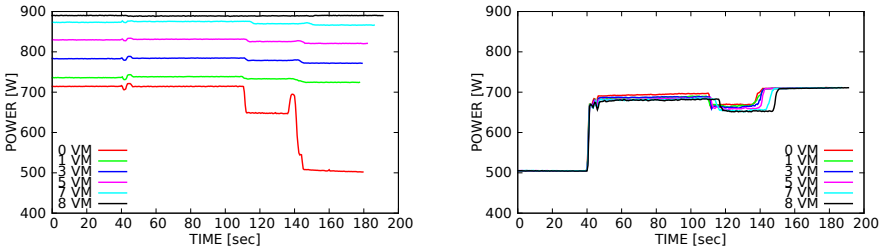
5.4.4. MEMLOAD-SOURCE

For the MEMLOAD-SOURCE experiment, we observe in Figure 11a 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



(a) Source. (b) Target.

Figure 10: MEMLOAD-VM results.



(a) MEMLOAD-SOURCE source. (b) MEMLOAD-SOURCE target.

Figure 11: MEMLOAD-SOURCE results.

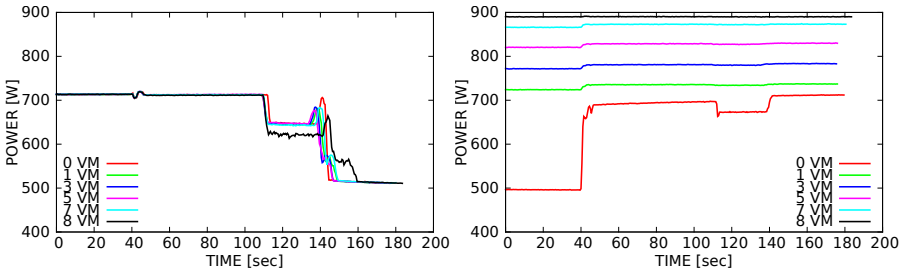
source. This tendency is better seen for high amount of loads for the target host (Figure 11b), when we notice a considerable increase in the transfer phase due to the reduced bandwidth. We also observe that the CPU load on the source host has an impact on the energy consumption of migration even in case of memory-intensive workloads, for which reason we included it in Equation 11. 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 8a and 8b.

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 9: Coefficients for live migration.

Model	*Host*	*NRMSE* (non-live) (m01 − m02)	*NRMSE* (live) (m01 − m02)	*NRMSE* (non-live) (o1 − o2)	*NRMSE* (live) (o1 − o2)	*MAPE* (non-live) (m01 − m02)	*MAPE* (live) (m01 − m02)	*MAPE* (non-live) (o1 − o2)	*MAPE* (live) (o1 − o2)
WAVM[3]	Source	11.8%	11.8%	12.5%	12.7%	4%	11%	4%	10.3%
	Target	12%	5%	16.3%	17.2%	4.1%	6.8%	4.1%	11.4%

Table 10: Normalised root mean square error (NRMSE) and mean absolute percentage error (MAPE) of our model on the two datasets.



(a) MEMLOAD-TARGET source. (b) MEMLOAD-TARGET target.

Figure 12: MEMLOAD-TARGET results.

5.4.5. MEMLOAD-TARGET

For the MEMLOAD-TARGET experiment, we see in Figure 12a 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 12b), 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 9a and 9b.

5.5. Regression analysis

In this section we use the results obtained with our experiments (Section 5.4) to compute the coefficients $\alpha, \beta, \gamma, \delta$ of the model we developed in Section 5.2. The coefficients are determined for each one of the energy phases previously identified. We use regression analysis based on the Non Linear Least Square Marquardt-Levenberg algorithm, implemented in `gnuplot` 5.0. 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 a subset of the readings obtained by running our experiments on the machines `m01` – `m02`. The coefficients for non-live migration are summarised in Table 8, while the coefficients for live migration are summarised in Table 9. 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 (`o1` – `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 (`m01` – `m02`) and C^2 for the prediction on (`o1` – `o2`). The error for our model in both datasets is shown in Table 10.

5.6. Comparison

We compare in this section 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 [5]. We describe each one of these models in detail next.

HUANG [3]. This model is based on the assumption that the instantaneous power consumption P of each host is linear with the CPU utilisation `CPU` [39]:

$$P = \alpha \cdot \texttt{CPU} + \mathcal{C}, \tag{13}$$

where P 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).

LIU [4]. This model is based on the assumption that energy consumption of VM migration $E_{\texttt{migr}}$ depends only on the amount of data `DATA` exchanged by the two hosts during the VM migration:

$$E_{\texttt{migr}} = \alpha \cdot \texttt{DATA} + \mathcal{C}, \tag{14}$$

In their work, the authors calculate the amount of data exchanged during migration as a function of VM memory size, memory transmission rate and memory dirtying ratio. We use the amount of data transferred measured with our network instrumentation as the `DATA` value. In this model, α models the linear relationship between the transferred data and energy consumption and $\mathcal{C}$ is an 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 by [40], this assumption may lead to inaccurate results.

STRUNK [5]. This model considers VM memory size $\texttt{MEM}(v)$ and network bandwidth between source and target $\texttt{BW}(\texttt{S}, \texttt{T})$ as parameters in a linear model:

$$E_{\texttt{migr}} = \alpha \cdot \texttt{MEM}(v) + \beta \cdot \texttt{BW}(\texttt{S}, \texttt{T}) + \mathcal{C}, \tag{15}$$

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 [41], many works show the benefits of consolidating VMs executing

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

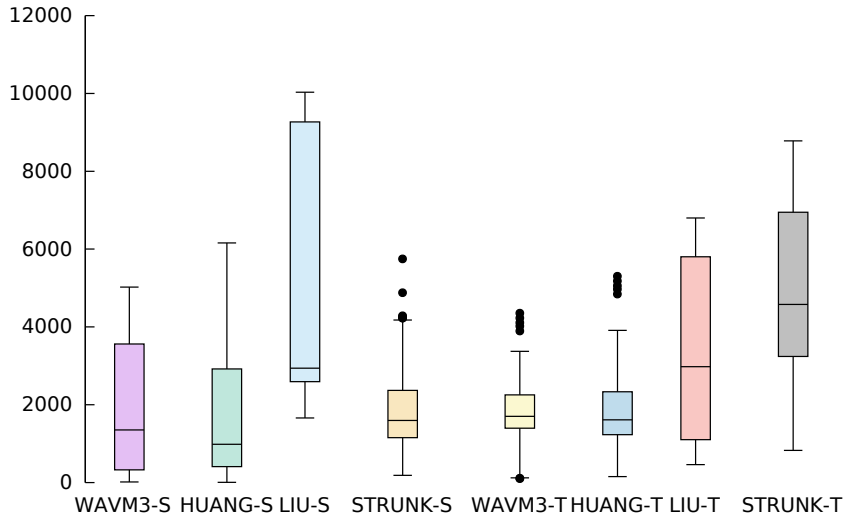
Table 11: Training phase coefficients for LIU, HUANG and STRUNK models.

tasks to/from hosts that are not idle [42]. 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.

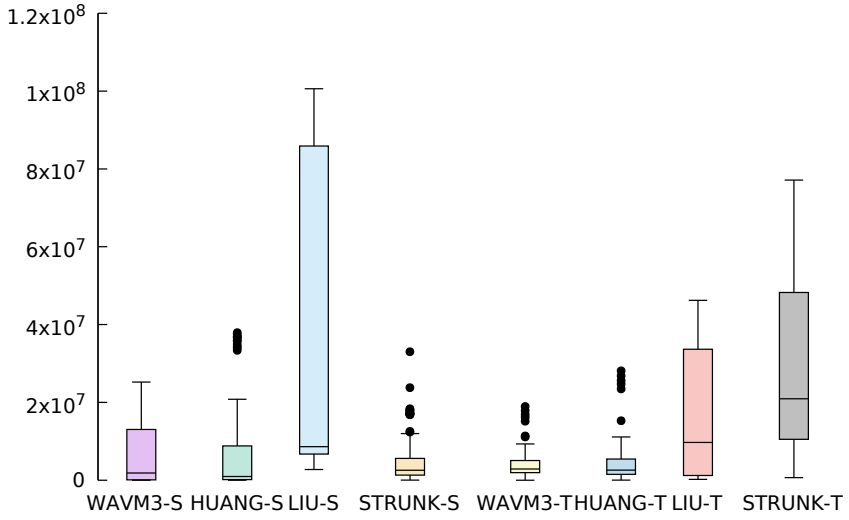
We train these models using the same training set used to train our model and the coefficients obtained for each model are summarised in Table 11. Afterwards, we compute four error metrics on the test set, summarised in Table 12. We also show the error distribution in Figure 13 for the non-live migration and in Figure 14 for live migration. The error metrics we choose are the Mean Absolute Error (MAE), the Mean Absolute Percentage Error (MAPE), the Root Mean Square Error (RMSE) and the Normalised Root Mean Square Error (NRMSE). Unless differently specified, we compare the models by using the NRMSE metric, because it is more sensitive to large deviations in the predictions. We add the other metrics for completeness of information. In the next subsections, we compare the results of our model named Workload-Aware Virtual Machine Migration Model (WAVM^3) with the other three.

5.6.1. Non-live migration

By looking at Table 12, we observe that, among the models we chose for comparison, 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,



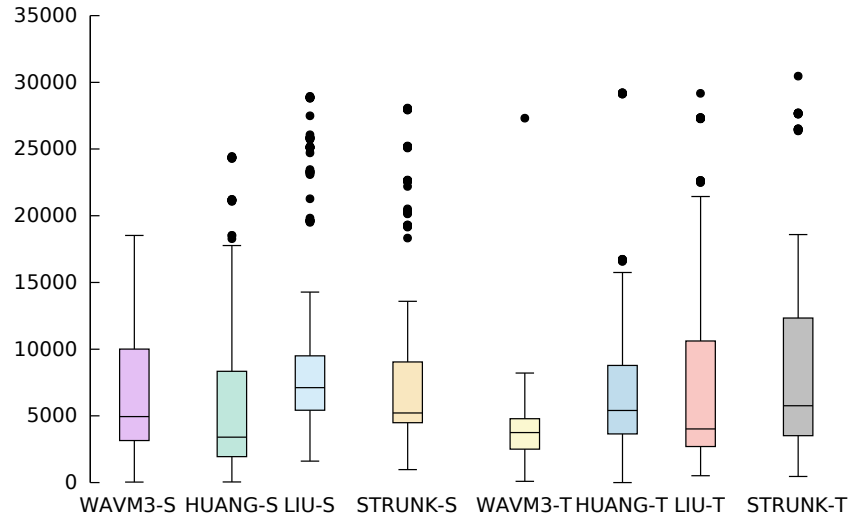
(a) Absolute Error. S: source host, T: target host



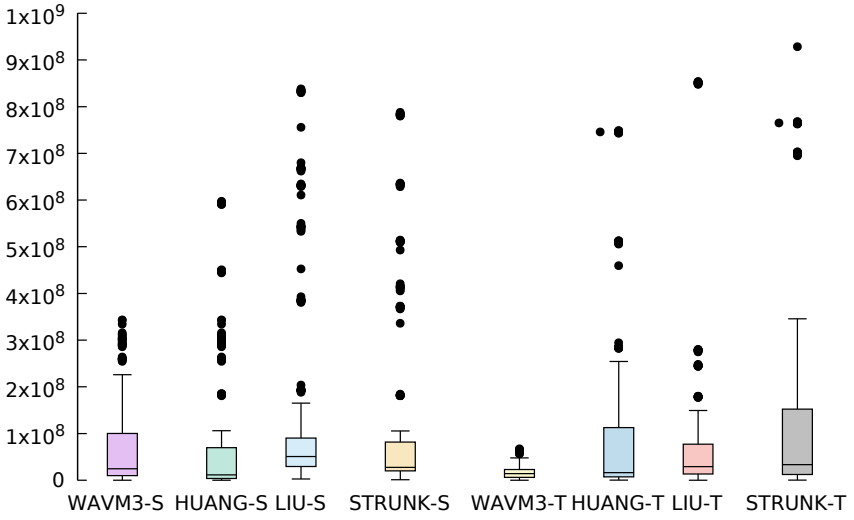
(b) Square Error. S: source host, T: target host

Figure 13: Error distribution for non live migration

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



(a) Absolute Error. S: source host, T: target host



(b) Square Error. S: source host, T: target host

Figure 14: Error distribution for live migration

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

Model	*Host*	*MAE* (non-live) [kJ]	*RMSE* (non-live)	*NRMSE* (non-live)	*MAPE* (non-live)	*MAE* (live) [kJ]	*RMSE* (live)	*NRMSE* (live)	*MAPE* (live)
WAVM^3	Source	1.8	2558	11.8%	4%	6.3	8432	11.8%	11%
	Target	1.7	1789	12%	4.1%	3.6	4056	5%	6.8%
HUANG	Source	1.8	2587	12%	3.2%	5.5	9234	15.7%	10.4%
	Target	1.8	2067	12.8%	4.2%	7.1	9102	12.9%	11.6%
LIU	Source	4.8	5812	26.9%	10.7%	9.8	12117	36.3%	39.7%
	Target	3.4	4121	25.3%	7.2%	7	9622	29.4%	31.9%
STRUNK	Source	0.026	3824	17.7%	24.9%	0.028	4547	35.4%	24.9%
	Target	0.058	5187	30%	27%	0.019	4382	36.2%	27%

Table 12: Comparison of WAVM^3 with other models on dataset m01-m02.

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.

5.6.2. Live migration

The errors for the live migration are summarised in Table 12. 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 6.1% (6.8% vs 12.9% NRMSE) for the target host. Comparing with the other models, the increase we obtain is up to 24.5% (vs Liu et al.) for source and 23.5% (vs Strunk et al.) for target, showing that our modeling approach increaseS the accuracy of the current state of the art.

6. Conclusion

We performed in this paper a comparative analysis of the energy efficiency of today's mostly used NIC families in data centres, Gigabit Ethernet and Infiniband. First, we introduced NNETS, a versatile network benchmarking tool offering eight configuration parameters, some not covered by existing tools (e.g. variable traffic patterns, full duplex connections). Second, we designed a set of benchmarks and evaluated the energy efficiency of the NICs' software stacks in different configurations covering a wide spectrum of possible application behaviours. Third, we introduced energy models capable of providing accurate estimations based on the NIC type of adapter and transfer characteristics including payload size, connection concurrency and traffic patterns with an average error of 6.1%.

Afterwards, we used these results to develop `WAVM`^{3}, an energy model for VM migration. We considered both the network transfer energy and the impact of workloads running on different actors and identified how much their load impacts the energy consumption of VM migration. Then, we evaluated the accuracy of the model on a different set of machines than the one we used for training our model. Finally, we compared the accuracy of our model versus other state-of-art models that do not consider it. We quantify how much each actor's workload influences the VM migration energy-wise. Compared to other state-of-art models, `WAVM`^{3} shows an improvement of up to 24% in prediction accuracy of the VM migration energy consumption, proving that: (1) using our network transfer model does not affect the accuracy of the energy consumption of our VM migration prediction, and (2) the workload impact on VM migration cannot be ignored when predicting its energy consumption.

Our results show that employing our model can be helpful for taking more energy-efficient VM consolidation decisions. For example, one may think not to consolidate VMs with a high dirtying rate to a host that is running CPU intensive workloads since, as shown in Figure 12, 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 workload running on the target host and may not be able to provide the same accuracy in predictions. We intend to integrate our models in the GroudSim Cloud simulators to provide a more accurate estimation of the energy consumption in data centres. We further plan to extend this work by also considering the impact of network-intensive workloads and to integrate it into existing consolidation managers and show the improvements in automatic consolidation decisions.

References

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

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

[3] Q. Huang, F. Gao, R. Wang, Z. Qi, Power consumption of virtual machine live migration in clouds, in: Communications and Mobile Computing (CMC), 2011 Third International Conference on, 2011, pp. 122 –125.

[4] H. Liu, C.-Z. Xu, H. Jin, J. Gong, X. Liao, Performance and energy modeling for live migration of virtual machines, in: Proceedings of the 20th International Symposium on High Performance Distributed Computing, ACM, 2011, pp. 171–182.

[5] 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, 2013, pp. 510–517.

[6] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, H. Liu, Energy proportional datacenter networks, SIGARCH Comput. Archit. News 38 (3) (2010) 338–347.

[7] L. Huang, Q. Jia, X. Wang, S. Yang, B. Li, Pcube: Improving power efficiency in data center networks, in: Cloud Computing (CLOUD), 2011 IEEE International Conference on, IEEE, 2011, pp. 65–72.

[8] D. Kliazovich, P. Bouvry, S. U. Khan, Dens: Data center energy-efficient network-aware scheduling, in: Green Computing and Communications (GreenCom), 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing (CPSCom), IEEE, 2010, pp. 69–75.

[9] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall, Reducing network energy consumption via sleeping and rate-adaptation, in: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, USENIX, 2008, pp. 323–336.

[10] H. sheng Wang, L. shiuan Peh, S. Malik, A power model for routers: Modeling alpha 21364 and infiniband routers, IEEE MICRO 23 (1) (2003) 26–35.

[11] P. Alonso, R. Badia, J. Labarta, M. Barreda, M. Dolz, R. Mayo, E. Quintana-Orti, R. Reyes, Tools for power-energy modelling and analysis of parallel scientific applications, in: Parallel Processing (ICPP), 2012 41st International Conference on, 2012, pp. 420–429.

[12] S. Jana, O. Hernandez, S. Poole, B. Chapman, Power consumption due to data movement in distributed programming models, in: Euro-Par 2014, Vol. 8632, Springer International Publishing, 2014, pp. 366–378.

[13] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, M. Sridharan, Data center tcp (dctcp), SIGCOMM Comput. Commun. Rev. 41 (4) (2010) –.

[14] H. Shirayanagi, H. Yamada, 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, 2012, pp. 000460–000467.

[15] V. De Maio, V. Nae, R. Prodan, Evaluating energy efficiency of gigabit ethernet and infiniband software stacks in data centres, in: Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on, IEEE, 2014, pp. 21–28.

[16] M. Alicherry, T. V. Lakshman, Network aware resource allocation in distributed clouds, in: INFOCOM '12, IEEE, 2012, pp. 963–971.

[17] Y. Shang, D. Li, M. Xu, A comparison study of energy proportionality of data center network architectures, in: Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, 2012, pp. 1–7.

[18] H. Shirayanagi, H. Yamada, K. Kono, Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks, 2014 IEEE Symposium on Computers and Communications (ISCC) 0 (2012) 000460–000467.

[19] Y. Shang, D. Li, M. Xu, Energy-aware routing in data center network, in: SIGCOMM '10 workshop on Green networking, ACM, 2010, pp. 1–8.

[20] A. Bianzino, C. Chaudet, F. Larroca, D. Rossi, J. Rougier, Energy-aware routing: A reality check, in: GLOBECOM Workshops (GC Wkshps), 2010 IEEE, 2010, pp. 1422–1427.

[21] L. Wang, F. Zhang, C. Hou, J. Arjona Aroca, Z. Liu, Incorporating rate adaptation into green networking for future data centers, in: Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on, 2013, pp. 106–109.

[22] Y. Hanay, W. Li, R. Tessier, T. Wolf, Saving energy and improving tcp throughput with rate adaptation in ethernet, in: Communications (ICC), 2012 IEEE International Conference on, 2012, pp. 1249–1254.

[23] M. Gupta, S. Singh, Greening of the internet, in: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, 2003, pp. 19–26.

[24] A.-C. Orgerie, L. Lefevre, I. Guerin-Lassous, D. Pacheco, Ecofen: An end-to-end energy cost model and simulator for evaluating power consumption in large-scale networks, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a, 2011, pp. 1–6.

[25] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Warfield, Live migration of virtual machines, in: Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2, NSDI'05, USENIX, pp. 273–286.

[26] S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, in: HotPower'08, USENIX, 2008, pp. 10–10.

[27] C.-H. Hsu, S.-C. Chen, C.-C. Lee, H.-Y. Chang, K.-C. Lai, K.-C. Li, C. Rong, Energy-aware task consolidation technique for cloud computing, CloudCom '11, IEEE, 2011, pp. 115–121.

[28] R. Sinha, N. Purohit, H. Diwanji, Energy efficient dynamic integration of thresholds for migration at cloud data centers, IJCA Special Issue on Communication and Networks (1) (2011) 44–49.

[29] A. Beloglazov, R. 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) (2012) 1397–1420.

[30] W. Voorsluys, J. Broberg, S. Venugopal, R. Buyya, Cost of virtual machine live migration in clouds: A performance evaluation, in: Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09, Springer-Verlag, 2009, pp. 254–265.

[31] A. Verma, G. Kumar, R. Koller, A. 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, IEEE Computer Society, 2011, pp. 3–11.

[32] T. Hirofuchi, A. Lèbre, L. Pouilloux, Adding a Live Migration Model Into SimGrid, One More Step Toward the Simulation of Infrastructure-as-a-Service Concerns, in: IEEE CloudCom, 2013.

[33] A. Strunk, Costs of virtual machine live migration: A survey, in: Services (SERVICES), 2012 IEEE Eighth World Congress on, 2012, pp. 323–329.

[34] Q. Huang, F. Gao, R. Wang, Z. Qi, Power consumption of virtual machine live migration in clouds, in: Communications and Mobile Computing (CMC), 2011 Third International Conference on, 2011, pp. 122–125.

[35] J. Chu, V. Kashyap, Transmission of IP over InfiniBand (IPoIB), RFC 4391 (2006).

[36] V. Kashyap, IP over InfiniBand: Connected Mode, RFC 4755 (2006).

[37] T. Benson, A. Akella, D. A. Maltz, Network traffic characteristics of data centers in the wild, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, ACM, 2010, pp. 267–280.

[38] W. Li, H. Yang, Z. Luan, D. Qian, Energy prediction for mapreduce workloads, in: Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on, 2011, pp. 443–448.

[39] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, F. Zhao, Energy-aware server provisioning and load dispatching for connection-intensive internet services, in: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, USENIX Association, 2008, pp. 337–350.

[40] A.-C. Orgerie, L. Lefevre, J.-P. Gelas, Demystifying energy consumption in grids and clouds, in: Green Computing Conference, 2010 International, 2010, pp. 335–342.

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

[42] J. Sekhar, G. Jeba, S. Durga, A survey on energy efficient server consolidation through vm live migration, International Journal of Advances in Engineering & Technology, 5 (2012) 515–525.