Estimating Energy Consumption of Cloud, Fog and Edge Computing Infrastructures

Ehsan Ahvar, Anne-Cécile Orgerie, Adrien Lebre

▶ **To cite this version:**

HAL Id: hal-02083080
https://hal.science/hal-02083080v1

Submitted on 28 Mar 2019

Estimating Energy Consumption of Cloud, Fog and Edge Computing Infrastructures

Ehsan Ahvar, Anne-Cécile Orgerie and Adrien Lebre

Abstract—In order to improve locality aspects, new Cloud-related architectures such as Edge Computing have been proposed. Despite the growing popularity of these new architectures, their energy consumption has not been well investigated yet. To move forward on such a critical question, we first introduce a taxonomy of different Cloud-related architectures. From this taxonomy, we then present an energy model to evaluate their consumption. Unlike previous proposals, our model comprises the full energy consumption of the computing facilities, including cooling systems, and the energy consumption of network devices linking end users to Cloud resources. Finally, we instantiate our model on different Cloud-related architectures, ranging from fully centralized to completely distributed ones, and compare their energy consumption. The results show that a completely distributed architecture, because of not using intra-data center network and large-size cooling systems, consumes between 14% and 25% less energy than fully centralized and partly distributed architectures respectively. To the best of our knowledge, our work is the first one to propose a model that enables researchers to analyze and compare energy consumption of different Cloud-related architectures.

Index Terms—Cloud computing, energy consumption, distributed Clouds, Fog computing, Edge computing, Peer-to-peer.

I. INTRODUCTION

Energy consumption of Cloud Computing solutions has become one of the critical challenge of our modern computing facilities. Data centers (DCs) constitute major elements consuming energy in Cloud computing. According to Lawrence Berkeley National Laboratory (LBNL)'s report, published in 2016 [1], DCs consumed an estimated 70 billion kWh (about 1.8% of total U.S. electricity consumption) and they are projected to consume approximately 73 billion kWh by 2020 in the U.S.

At the same time, the proliferation of new usages related to Internet of Things (IoT) calls for more distributed cloud-related architectures, relying on resources deployed across and at the edge of the network. Referred to as Fog, Edge and sometimes P2P computing infrastructures [2], [3], these emerging virtualized architectures aim at satisfying low latency and high bandwidth requirements expected by IoT-based applications. While there is no more debate on whether such infrastructures will be deployed, the question of their energy footprint has not been studied yet.

Ehsan Ahvar and Anne-Cécile Orgerie are with Univ. Rennes, Inria, CNRS, IRISA, Rennes - France

Adrien Lebre is with IMT-Atlantique, Inria, LS2N, Nantes - France

Corresponding author: A.-C. Orgerie (email: anne-cecile.orgerie@irisa.fr).

The main objective of this paper is to propose a generic energy model to accurately evaluate and compare the energy consumed by these new Cloud-related architectures.

Frontiers of cloud-related architectures are currently moving, and we first present the existing definitions and the key features of each architecture. We then propose to identify each of them as one well defined architecture, ranging from centralized to fully distributed ones. To this end, we propose a taxonomy that categorizes available Cloud-related architectures based on their characteristics and classifies hardware elements of each infrastructure. Based on the proposed taxonomy, an energy model is proposed. As the model accurately considers the differences among architectures, it can be used to estimate energy consumption of one particular architecture as well as comparing them.

Our key contributions are to:

(i) Develop a taxonomy of Cloud-related architectures and components that provide an extensive coverage of this field in terms of technical features, services and user satisfaction. The main aim of our proposed taxonomy is to explore the unique features of different architectures and components from similar paradigms and to provide a basis for categorizing present and future architectures.

(ii) Present a generic and accurate model to estimate energy consumption of the existing Cloud-related infrastructures. The model provides a basis for an in-depth analysis and clear understanding of the current Cloud landscape. Besides, this work gives an insight into the underlying technologies that are currently deployed in the domain.

The rest of the paper is organized as follows: Section II presents the related work. Section III categorizes Cloud-related architectures using our proposed taxonomy. Section IV introduces the proposed energy model. Energy consumption of different architectures are analyzed in Section V. Finally, Section VI concludes our work.

II. RELATED WORK

In this section, we first introduce existing Cloud-related infrastructures. Then, we provide a comparison among the energy models of the most commonly used Cloud and DC simulators. Finally, the efforts to compare energy consumption of different Cloud-related architectures are presented.

A. Virtualized architectures

Cloud computing has recently moved to more distributed architectures as an answer to users' needs for mobility, high data volume processing and low latency. In 2014, the European

Telecommunications Standards Institute (ETSI) has proposed a new Cloud architecture named Mobile Edge Computing [4]. According to the initial definition, "*Mobile-edge Computing provides IT and cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile subscribers*" [4]. The Mobile Edge Computing (MEC) group of ETSI was renamed to Multi-access Edge Computing in 2016 to enlarge its access to heterogeneous networks [5]. MEC's infrastructures are managed through hierarchical or distributed control depending on the considered application [5], [6]. They can include mobile devices when they are used to provide computation to smaller devices [7].

Driven by the expansion of Internet of Things (IoT), Cisco proposed in 2012 the concept of Fog Computing [8]. According to the Open Fog Consortium, "*Fog computing is a system-level horizontal architecture that distributes resources and services of computing, storage, control and networking anywhere along the continuum from Cloud to Things*" [9]. In particular, this architecture aims at dealing with the time-sensitive and large volumes of data generated at the network edge by IoT devices [10]. To this end, Fog computing can extend the cloud to any device nearby the Things and with computing, storage and network connectivity [8]. The Fog Computing architecture can be seen as hierarchical with three layers: the IoT devices, the Fog nodes and the Cloud data center [10]. Depending on the applications, the fog nodes are managed either in a decentralized or distributed way [10].

From the industrial point of view, the differences between both models are unclear and some actions such as the Open Glossary of Edge Computing initiated by the Linux Foundation aims at providing accurate definitions on this topic [11].

B. Energy models and simulation tools

Recent studies [12], [13] show that more than 20 Cloud and DC simulators have been proposed in the past decade, $CloudSim$ [14] being one of the most popular. Despite its advantages (*e.g.*, modeling large-size DCs, federation model and supporting large number of virtual machines), $CloudSim$ has several limitations, including limited communication and network energy models. Because of the extensible nature of $CloudSim$, a lot of works [15]–[17] have been done for enriching it. Among them, a few works focused on energy model of CloudSim. For example, Li et al. [17] proposed $DartCSIM$ to enable the simulation of energy and network simultaneously. Other Cloud simulators have been proposed, like $iCanCloud$ [18], but they are not designed to support distributed DC architectures [19]. Malik et al. [19] proposed $CloudNetSim$++ which is a GUI-Based framework for modeling and simulation of DCs in OMNeT++ including simulation models for network devices. Yet, this simulator does not consider effect of packet length on energy consumption in its models. Finally, the SimGrid toolbox proposes accurate modeling of virtual machines [20], and energy consumption of servers [21]. But, the energy consumption of network components is not provided.

Despite existence of above mentioned tools, we could not find a simple model or simulator that easily supports different Cloud-related architectures, ranging from fully centralized to completely distributed, without going into the details of programming and extending the aforementioned simulators. For example, in order to compare different Cloud-related architectures, we need to consider the number and location of their control nodes, as discussed later in this article. Finally, a majority of the available simulators also does not produce accurate results of architectures components or do not consider their idle energy consumption part. For all these reasons, we design a generic energy model to compare different Cloud-related architectures.

To the best of our knowledge, our work is the first one to propose a model that enables researchers to analyze (and so compare) energy consumption of different Cloud-related architectures. Although, Jalali et al. [22] compare energy consumption of applications provided by a centralized DC and Fog Computing, several parts of their proposed energy models are not accurate enough to answer the questions of energy footprint of Cloud-related architectures. Li et al. [23] compare the energy consumption of IoT applications provided by centralized Cloud and Edge computing. C. Fiandrino et al. [24] also propose metrics to assess and compare energy efficiency of communication systems of Cloud Computing DCs. In all cases [22]–[24], the proposed models are incomplete for accurately analyzing the various Cloud-related architectures. For instance, as the size and intra-network topology of centralized DC are not considered, the part of intra-DC network cannot be analyzed. Moreover, they often focus on the impact of a given application or kind of applications [25].

III. PROPOSED TAXONOMY

In recent years, many Cloud-related architectures have been proposed to satisfy both users and providers requirements. Putting architectures with several similar characteristics into the same group can noticeably simplify presentation and design of our energy model. To this end, we propose a taxonomy to appropriately cover and classify these architectures.

A. Cloud-related Architectures

A Cloud-related architecture includes one or more DCs and a number of end users, all connected together through telecommunication networks. An end user is a sender of requests and a receiver of services. A DC offers different Cloud services to the end users and consists of a number of Compute Nodes (CNs) that host the Virtual Machines (VMs), and Service Nodes (SNs) that are either infrastructure management nodes or storage nodes. The main distinctions among Cloud-related architectures are location, number, size and role (*e.g.*, controller, storage, compute together or part of them) of their DCs.

Figure 1 shows our proposed taxonomy. Based on DC characteristics, we divide architectures into four main groups: Fully Centralized (FC), Partly Distributed (PD), Fully Distributed-Centralized Controller (FD-CC), and Fully Distributed (FD). **Fully Centralized (FC) –** The FC architecture includes a large-size DC. All end users connect to the central DC and receive services from it.
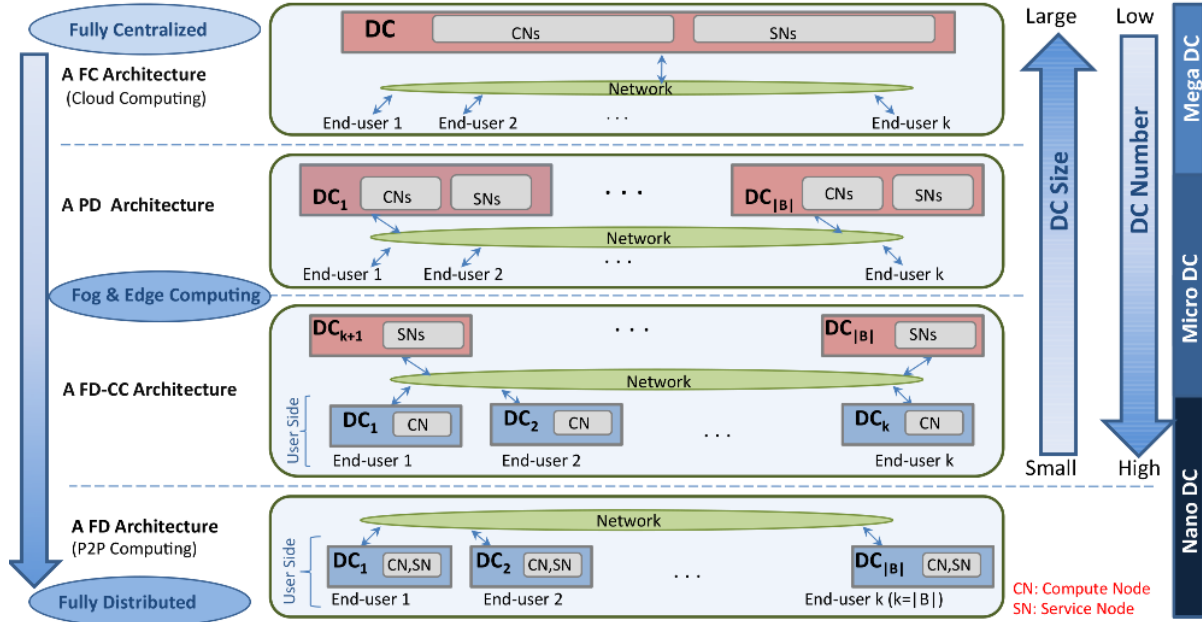
Fig. 1. Considered architectures; ranging from fully centralized (FC) to fully distributed (FD).

Partly Distributed (PD) – In a PD architecture, several DCs are distributed through different locations and working together to serve the end users. A telecommunication network connects the Cloud DCs together. It also connects end users to DCs. In a PD architecture, the DCs are usually located closer to end users in comparison to the FC architecture, because of the higher number of DCs. However, the DCs are not located in end users premises.

Fully Distributed with Centralized Controller (FD-CC) – FD-CC architecture includes several DCs that are located on end users premises (*i.e.*, one DC for one or a few end users). However, controller(s) (*i.e.*, as SNs) are located in one or more DCs in the core of network. The DCs that are located in the end users premises are usually in size of a Physical Machine (PM). The remaining DCs, in core of network, are only used for management aspect and they are not used for computing and storage services.

Fully Distributed (FD) – In an FD architecture, unlike for FD-CC, the management system is also distributed on end users locations. It means there is no DC in the core of network in charge of controlling the resources (*i.e.*, no independent SN).

The employed abbreviations are summarized in Table I. The FC architecture represents centralized Cloud infrastructures. Examples of PD architectures can be found among decentralized Fog infrastructures [10]. The FD-CC architecture gathers the hierarchical Fog and Edge infrastructures with distributed nodes and decentralized management nodes, hosting for instance Virtual Network Functions' (VNFs) orchestration [5], [10]. Finally, the FD architecture represents the fully distributed edge infrastructures [6]. Note that in the context of this work, we do not consider the wide variety of wireless network technologies [26]. It is considered as future work to include an architecture with mobile fog nodes.

TABLE I
LIST OF ABBREVIATIONS

General terms	
DC	Data Center
PM, VM	respectively Physical Machine and Virtual Machine
CN, SN	respectively Compute Node and Service Node
PUE	Power Usage Effectiveness
C, B, M, F	respectively Core, Backbone, Metro and Feeder router of the telecommunication network
Cloud architectures	
FC	Fully Centralized architecture
PD	Partly Distributed architecture
FD-CC	Fully Distributed with Centralized Controller architecture
FD	Fully Distributed architecture

B. Architecture's components

Figure 2 presents a comprehensive overview of ICT equipment of all Cloud-related architectures. Any architecture includes two main parts: DC(s) and telecommunication network, that are described hereafter.
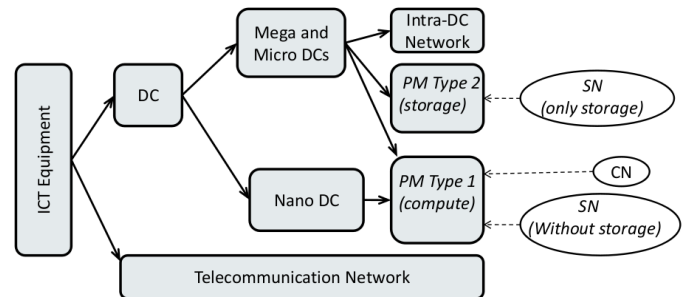

Fig. 2. ICT equipment composing the different architectures.

Besides, DCs can be classified by their size: *mega* when the number of hosted PMs is higher than 10,000, *micro* when this number is less than 500, and finally *nano* when there is

only a few PMs. In our case, we consider an extreme case where a *nano* DC consists of one single PM and does not have any extra additional ICT equipment such as the intra-DC network elements (*i.e.*, switches and links). Moreover, it is noteworthy that all DCs but *nano* ones also have non-ICT equipment such as an independent cooling system. To reflect the energy consumption of non-ICT equipment available in DCs (*i.e.*, from the micro to the mega ones), the Power Usage Effectiveness (PUE) is a well-known DC energy-efficiency indicator. It represents the ratio between the total facility and the IT equipment energy consumption [27]. In other words, the overall energy consumption of a DC can be estimated by multiplying the energy consumption of its ICT equipment and its PUE value.

As Figure 2 shows, PMs can be classified into two main types: *Type 1 PMs* are specialized for computing purposes (*e.g.*, Computing and Cloud management nodes). *Type 2 PMs* are used to store large data-sets thanks to their specific storage backends (*i.e.*, storage node). Because of the additional storage devices, each type of PM presents a different energy profile. In the DC terminology, Compute Nodes (CNs) correspond to type 1, while Service Nodes (SN) correspond to type 1 if they manage the infrastructure, or to type 2 if they are used for storage purposes. In the extreme case of FD architecture, with a single CN-based PM, SN functions are simply served by a VM directly hosted on the CN and using local disks as the default storage backend.

In addition to PMs, ICT equipment of a mega or micro DC includes network elements, like routers, switches and links, to connect PMs together. The number and type of network elements to connect PMs inside a DC is determined by its network topology. More than thirty network topologies have been proposed for DCs [28] since 2008. In this work, we consider a fat tree switch-centric architecture (*i.e.*, n-ary fat tree topology) which corresponds to the network topology of many currently available DCs [29].

Utilizing small-size and identical switches is considered as one noticeable benefit of the fat tree topology. This enables to utilize cheap commodity parts for all of the switches in the communication architecture. Besides, it simplifies the system model and analysis compared to architectures with non identical switches. This topology is flexible and scalable, thus adaptable to different sizes of DCs. We consider a n-ary fat tree topology with three layers (*i.e.*, core, aggregation and edge), $(n/2)^2$ core switches, and n pods (*i.e.*, each pod has two layers of $n/2$ aggregation and $n/2$ edge switches) [30]. This topology uses similar n-port switches, and can support up to $n^3/4$ PMs. An example with $n = 4$ is provided in Figure 3.
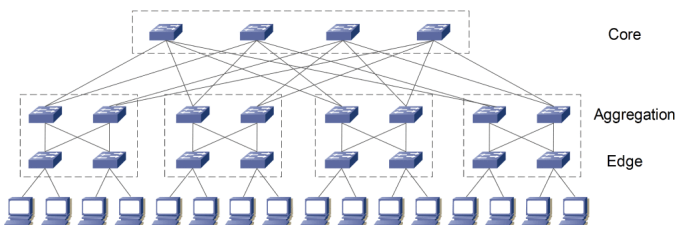
Finally, the telecommunication network connects all DCs and end users together. It usually includes four main logical layers [31]: core, backbone, metro and feeders (or end users). The metro layer provides access to the network for the end users. The backbone layer provides policy-based connectivity. The core layer offers high-speed transport to satisfy the requirements of the backbone layer devices. For the FC architecture, we assume the single DC to be connected directly to the core layer of the telecommunication network. For the PD architecture, the DCs are connected either to backbone or metro routers. For the FD-CC architecture, the DCs with CNs are connected either to backbone or metro routers and DCs with SNs are connected to the core layer. Finally, for the FD architecture, DCs are connected to the feeder layer. We assume that the routing policy always uses the shortest path.

IV. THE ENERGY MODEL

In this section, we introduce our energy model defined for the different Cloud-related architectures. We consider a scenario with V active VMs requested by a set of U end users. Our goal is to provide a generic model in order to estimate the energy consumption of each aforementioned cloud-related infrastructures for a given time period T when the allocated VMs are running. We underline that we do not take into consideration the differences among these architectures in terms of Quality-of-Service (*i.e.*, latency). Besides, only the energy consumption of the infrastructure itself is estimated: it includes the telecommunication network between DCs and users but not the end users' devices. In order to estimate energy consumption of an architecture, the model receives a set of inputs listed in Table II, along with their definitions.

A. General overview

As Figure 4 shows, our model divides energy consumption of an ICT equipment into static and dynamic parts [32]. The static energy consumption is the energy consumption without considering any workload (*i.e.*, resources are idle). The dynamic cost is calculated based on the current usage of Cloud resources by the active VMs.

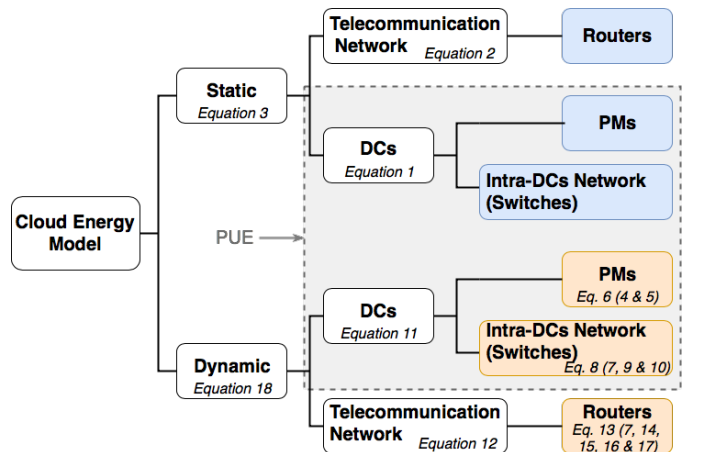Fig. 3. 4-ary fat tree topology from [30]

Fig. 4. General view of the proposed energy model and the equations used in the following to express the different parts.

TABLE II
LIST OF INPUTS FOR THE ENERGY MODEL

General Inputs	
U	number of end users
V	number of VMs
T	considered time period (in s)
Compute Inputs	
DC	set of the N DCs $DC = \{DC^1, \ldots, DC^N\}$ with N depending on the considered architecture
pm^i	number of PMs in DC^i with $i \in \{1, \ldots, N\}$
PUE^i	PUE of DC^i
CN^i	set of CNs located on DC_i
SN^i	set of SNs located on DC_i (for FD architecture, $SN^i = \emptyset$)
P_x^{idle}	power consumption of PM x belonging to DC^i in idle mode (in Watts) where $x \in \{CN^i \bigcup SN^i\}$
$P_{x,j}^{cores}$	dynamic power consumption of PM x when j of its k cores are fully loaded (in Watts) with $j \in \{0, \ldots, k\}$ and $P_{x,0}^{cores} = 0$
u_{SN}	average CPU utilization ratio per SN for service and storage functions
u_{VM}	average CPU utilization ratio per VM on CNs
u_H	average CPU utilization ratio per PM for the hypervisor and monitoring processes on CNs
Network Inputs	
L	average length of a packet on the intra-DC networks and the telecommunication network (in Bytes)
λ	average traffic rate generated by a PM belonging to a DC (in bps)
q	average ratio of packets sent by a PM to end users
μ	average traffic rate generated by a user to DCs (in bps)
λ_{SN}	average traffic rate from a CN to SNs in FD-CC architecture (in bps)
NE^i	set of switches belonging to DC_i with n ports each
P_c^{idle}	power consumption of switch c belonging to DC^i in idle mode (in Watts) where $c \in NE^i$
E_{switch}^{pkt}	dynamic energy consumption of a switch to process a packet (in Joules)
$E_{switch}^{S\&F}$	dynamic energy consumption of a switch to store and forward a Byte (in Joules)
RT	set of routers of the telecommunication network between DCs, and between DCs and end users
n_X	number of routers of category X in the telecommunication network with $X \in \{C, B, M, F\}$ where C, B, M and F stands respectively for core, backbone, metro and feeder routers
P_r^{idle}	power consumption in idle mode of router r with $r \in RT$
$E_{r_X}^{pkt}$	dynamic energy consumption of a router to process a packet (in Joules) with $X \in \{C, B, M, F\}$
$E_{r_X}^{SF}$	dynamic energy consumption of a router to store and forward a Byte (in Joules) with $X \in \{C, B, M, F\}$

Before presenting the static and dynamic parts of the energy model, we introduce technical assumptions, that simplify the model without loss of generality:

1) VMs are homogeneous in size (CPU and RAM). Modeling heterogeneous VMs can be done by assuming that a large size VM includes a set of small-size VMs (*i.e.*, a set of the basic unit of resources).
2) All V VMs are running during the entire time period T.
3) For each architecture, VMs are uniformly distributed on all active CN-based PMs. This assumption is considered to simplify design and utilization of the model. It can be relaxed by providing a scheduling algorithm specifying the number of VMs on each CN and the traffic exchanged with other VMs.
4) The intra-DC network has a fat tree topology, and the telecommunication network follows a hierarchical topol-

ogy as previously discussed.
5) Interfaces of switches are bi-directional, and capacities of uplink and downlink are similar. If there is no traffic in one direction, the entire interface is still active.

B. Static energy part of the model

As Figure 4 shows, static energy part of a Cloud-related architecture includes static energy of PMs and switches. For a PM, the idle power consumption corresponds to the power consumed by the server when powered on but not running any task [32]. Similarly, the energy consumption of load-independent part of a switch is considered as its idle power consumption. It includes the power consumption of elements such as chassis power supply and fans [33].

For each device, PM or switch, its static energy consumption corresponds to idle power consumption over the considered time period T as defined in Table II. As the idle power consumption is constant over time, multiplying it by T gives the energy consumption over the time period T. Consequently, the static energy consumption of DC^i corresponds to:

$$E_{DC^i}^{static} = \sum_{a \in CN^i} (P_a^{idle} \cdot T) + \sum_{b \in SN^i} (P_b^{idle} \cdot T) + \sum_{c \in NE^i} (P_c^{idle} \cdot T)$$
(1)

In Equation 1, CN^i, SN^i, and NE^i correspond to the number of CN-based PMs, SN-based PMs, and intra-DC switches for DC^i. P_x^{idle} corresponds to the idle power consumption of device x that can be: a PM type 1 if it belongs to CN^i or SN^i for some architectures; a PM type 2 if it belongs to SN^i for some architectures; or an intra-DC switch if it belongs to NE^i. For fully distributed architectures with *nano* DCs, consisting of one PM only, $NE^i = \emptyset \; \forall i$ since *nano* DCs do not need switches.

For the telecommunication network part, the static energy consumption corresponds to the idle power consumption of routers that are employed to link the DCs and the end users. So the static energy consumption of the architecture's network can be expressed as:

$$E_{Net}^{static} = \sum_{r \in RT} (P_r^{idle} \cdot T)$$
(2)

If some devices are in sleep mode instead of being idle for energy-aware management purpose, then the idle power consumption of these devices should be replaced by their power consumption in sleep mode in Equations 1 and 2. This parameter is not included in the equations for clarity's sake.

Putting all pieces together, the total static energy consumption of an architecture is computed as follows:

$$E_{total}^{static} = \sum_{i=1}^{N} (E_{DC^i}^{static} \cdot PUE^i) + E_{Net}^{static}$$
(3)

This equation gives the total static energy consumption as the sum of the static energy consumption of the telecommunication network devices and the static energy consumption of each DC including its cooling-related energy consumption (expressed by the PUE). In the following, we describe the dynamic energy consumption part attributed to PMs, intra-DC switches and telecommunication routers.

C. PMs dynamic energy consumption

Dynamic energy consumption of PMs depends mainly on CPU utilization [21], [32]. Based on previous work on multi-core servers [21], [34], we use a piecewise linear function to express the dependence of PM's dynamic energy consumption to its CPU utilization. This mapper function is instantiated with values taken from real energy measurements when fully loading the cores of a PM one by one. In [21], the authors show that this model is accurate to a few percent for the considered applications. So, for a PM x with k cores, average power consumption values are measured for 1, 2, and up to k fully loaded cores. Then, to these values are subtracted the idle power consumption of the PM P_x^{idle} to keep only the dynamic part. These dynamic power consumption values are denoted $P_{x,j}^{cores}$ with $j \in \{0, \ldots, k\}$ and $P_{x,0}^{cores} = 0$. The energy consumption mapper function is linear between each of these values. So, for a PM using a ratio u of its full CPU capacities, if $j/k \le u \le (j+1)/k$ with $j \in \{0, \ldots, k-1\}$, then the dynamic energy consumption of a given PM x during time period T can be expressed as:

$$E_x^{dyn} = T \cdot [(P_{x,j+1}^{cores} - P_{x,j}^{cores}) \cdot k \cdot u + [(j+1) \cdot P_{x,j}^{cores} - j \cdot P_{x,j+1}^{cores}]] \tag{4}$$

From the generic formula for a PM provided by Equation 4, one can derive the formulas for our two categories of PMs in DC^i by defining how to compute u_{CN} and u_{SN} respectively for CNs and SNs-based PMs. For SNs, we assume an average CPU utilization over time period T to determine their energy consumption, so u_{SN} is a fixed constant value. Recall that, SN-based PMs may have more disks than the other PMs. This additional energy consumption is reflected in the static energy consumption part by a higher idle power consumption for these nodes (P^{idle} in Equation 1). Similarly, for distributed architectures, as CN nodes play both roles (computing and storage), they have a higher idle power consumption.

For CNs belonging to CN^i for DC^i, their CPU utilization depends on the services and VMs it runs. For the VMs, their size in number of virtual CPUs is not a good estimator of their actual CPU utilization. Indeed, typical Cloud users tend to over-provision their VMs. Indeed, they target peak usage and do not fully utilize resource elasticity, thus inducing a low CPU usage inside VMs [34]. Moreover, Cloud providers resort to CPU over-commitment techniques for increasing their revenue without additional investment [34]. Hence, estimating energy consumption of a PM based on reserved resources of its VMs can lead to overestimation (*i.e.*, estimating even more than maximum power consumption of the PM). An alternative approach would consist in obtaining the actual CPU usage of PMs. Yet, it would be tedious to determine the exact CPU usage for each VM on each PM, and it might be difficult to acquire it on a production infrastructure. Furthermore, our model does not target a fine-grained computation of the energy consumption, but rather an overall estimation to compare Cloud-related architectures under the same initial workload conditions. Consequently, we consider here u_{VM} the average ratio for the CPU utilization per VM. Similarly, since our goal does not consist in evaluating VM placement strategies, we consider here a uniform allocation of VMs over

PMs. For instance, such an even distribution is the default scheduling behavior of the well-known cloud operating system OpenStack [35].

The values for the average CPU utilization per VM u_{CN} and the number of VMs per CN provide an estimate of the CPU usage per CN. To this utilization, we add a fixed amount u_H in order to account for the hypervisor energy consumption and all the Cloud-related processes for monitoring the PMs. The overall model for computing the dynamic energy consumption of a CN-based PM is depicted in Figure 5. So we have:

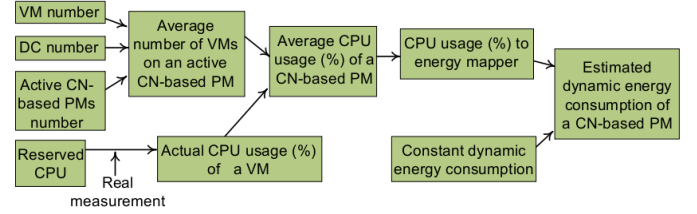$$u_{CN} = u_H + u_{VM} \cdot V / \sum_{i=1}^{N} |CN^i| \tag{5}$$

Fig. 5. Model for dynamic energy consumption of a CN-based PM

Finally, using Equation 4, the dynamic energy consumption related to PMs in DC^i is:

$$E_{DC^i\_PM}^{dyn} = \sum_{a \in CN^i} E_a^{dyn} + \sum_{b \in SN^i} E_b^{dyn} \tag{6}$$

D. Intra-DC network dynamic energy consumption

For estimating dynamic energy consumption of an intra-DC network, we need to estimate dynamic load-dependent energy consumption of each switch inside each DC. This step is done for the Cloud-related architectures whose DCs possess switches (*i.e.*, not fully distributed ones). When a switch carries traffic, it consumes additional load-dependent power for packet processing and also for storing and forwarding the payload across the switch fabric [33], [36], [37]. This additional load-dependent power consumption corresponds to the dynamic energy consumption of the switch.

As we consider a fat tree topology for the intra-DC network with identical switches, their energy profiles are identical. For a given switch belonging to a DC, we denote E_{switch}^{pkt} the energy to process a packet and E_{switch}^{SF} the energy for storing and forwarding a byte (from a packet). In [33], the authors show that this model based on two parameters (*i.e.*, consumption per packet and per byte) presents less than 1% error compared to real measurements when correctly instantiated.

Similarly to the VM's CPU utilization, it would be tedious to keep trace of the size of each packet and this fine-grained accuracy is not required for our model. So, we assume packets to have an average length of L bytes for the entire infrastructure. Consequently, if the switch receives np_T packets during a time period T, it has to process $np_T \times L$ bytes during T.

So, for a given switch, during a time period T, its dynamic energy consumption is computed as follows:

$$E_{switch}^{dyn} = E_{switch}^{pkt} \cdot np_T + E_{switch}^{SF} \cdot np_T \cdot L \tag{7}$$

Hence, we can express the dynamic energy consumption due to the switches in DC^i during a time period T as follows:

$$E_{DC^i\_NE}^{dyn} = E_{switch}^{dyn} \cdot |NE^i| = E_{switch}^{dyn} \cdot 5n^2/4 \quad (8)$$
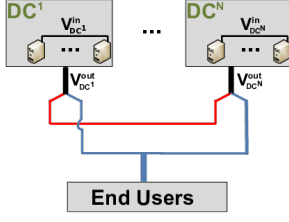

Fig. 6. Traffic model overview from the DCs point of view

In order to estimate np_T, we use the properties of the n-ary fat tree topology of the intra-DC network. As illustrated by Figure 6, there are two types of traffic inside a data center DC^i: internal traffic (*i.e.*, traffic exchanged between PMs of the DC) denoted $V_{DC^i}^{in}$, and external traffic (*i.e.*, traffic between PMs of the DC and outside) $V_{DC^i}^{out}$. For the external traffic, it is sent to or received from end-users or other data centers. First, we identify the paths taken by each type of traffic, then we quantify the data volumes for each of them.

A DC with a 3-level fat tree topology, using similar switches with n ports can embed up to $n^3/4$ PMs and $5n^2/4$ switches [30]. Due to the topology properties, each server can transmit at line speed if packets are uniformly distributed along the redundant paths. Furthermore, for such a topology, the longest path between PMs comprises 6 hops, and asymptotically, the number of hops also tends to 6 for a 3-level fat tree topology [38]. Thus, we estimate that the traffic staying inside DC uses 5 switches on average. The traffic going out of the DC uses 3 switches (*i.e.*, the shortest path, which includes 1 access, 1 aggregation and 1 core switches). Similarly, the traffic coming from outside the DC to PMs uses 3 switches.

As VMs are assumed to be evenly distributed among PMs and generate the same amount of traffic, we assume that the traffic generated per VM is also uniform on average. We denote λ the average traffic rate (in bps) of a PM. It depends on the average number of VMs hosted on the PM. So, during a time period T, each PM produces an average number of packets of $(\lambda \cdot T/8L)$ (with L being the average length of a packet in bytes). We denote q the average ratio of packets sent to end users per PM. The external traffic comprises the packets sent to end users and to PMs belonging to other DCs. A given DC^i comprises pm^i PMs with $pm^i = |CN^i| + |SN^i|$. So, for DC^i, $(pm^i \cdot q \cdot \lambda \cdot T/8L)$ packets are sent to end users during T.

As a uniform distribution of VMs is assumed among PMs, the traffic from a given PM to other PMs is evenly distributed among PMs. This means that we can compute p the probability that the traffic generated by a PM for other PMs stays inside its host DC:

$$p = (pm^i - 1)/\left(\sum_{i=1}^{N} pm^i - 1\right) \quad (9)$$

These packets have a probability p of staying inside the data center (internal traffic), and a probability $(1 - p)$ of

going outside the data center (external traffic). For *nano* DCs (containing only one node), $p = 0$. So, during T each PM of DC^i sends $(p \cdot \lambda(1-q)T/8L)$ packets that stay inside DC^i, and $((1-p)\lambda(1-q)T/8L)$ packets that are on average uniformly distributed among the other DCs. Similarly, DC^i receives packets from other DCs uniformly distributed.

For the traffic coming from end users, we denote μ the average traffic rate generated by each user to DCs (in bps). So, DC^i receives $(U \cdot \mu \cdot T)/(8L \cdot N)$ from the end users during T with U the total number of end users and N the number of DCs. We can then express the overall traffic processed in a DC as the sum of: the traffic sent to end users, the internal traffic exchanged between its PMs, the traffic sent to other DCs, the traffic coming from other DCs and the traffic coming from end users. Then np_T, the average number of packets processed by a switch during T, is equal to the overall traffic processed in DC^i by all the switches during T divided by the number of switches of DC^i and the average size of a packet ($8L$ in bits):

$$np_T = \Big[3 \cdot q \cdot \lambda \cdot pm^i + 5 \cdot p \cdot \lambda \cdot (1-q) \cdot pm^i$$
$$+ 3 \cdot (1-p) \cdot \lambda \cdot (1-q) \cdot pm^i$$
$$+ \frac{3 \cdot \sum_{j \in \{1...N\}\setminus\{i\}}(1-p) \cdot \lambda \cdot (1-q) \cdot pm^j}{(N-1)}$$
$$+ \frac{3 \cdot U \cdot \mu}{N} \Big] \Big/ \Big[8 \cdot L \cdot \frac{5 \cdot n^2}{4} \Big] \quad (10)$$

Finally, combining Equations 6 and 8, we compute the overall dynamic energy consumption of DC^i:

$$E_{DC^i}^{dyn} = (E_{DC^i\_PM}^{dyn} + E_{DC^i\_NE}^{dyn}) \cdot PUE^i \quad (11)$$

E. Telecommunication network dynamic energy consumption

Now we detail how to compute the dynamic energy consumption of the telecommunication network that is linking DCs among them and with the users. We consider a full telecommunication network with four different types of routers: core, backbone, metro and feeder routers all included in RT. Similarly to the dynamic energy consumption of switches in Equation 7, the dynamic energy consumption of the telecommunication network during T is equal to:

$$E_{Net}^{dyn} = \sum_{r \in RT} (E_r^{pkt} \cdot np_{r,T} + E_r^{SF} \cdot np_{r,T} \cdot L) \quad (12)$$

with $np_{r,T}$ the average number of packets processed by a given router r during T, E_r^{pkt} the energy to process a packet and E_r^{SF} the energy for storing and forwarding a byte. These two last parameters depend on the router type since routers from different types have different architectures and energy profiles. We denote n_C, n_B, n_M and n_F respectively the number of Core, Backbone, Metro and Feeders routers ($|RT| = n_C + n_B + n_M + n_F$). So, if we split Equation 12 to express it with each category of router, we obtain:

$$E_{Net}^{dyn} = n_C \cdot (E_{r_C}^{pkt} \cdot np_{r_C,T} + E_{r_C}^{SF} \cdot np_{r_C,T} \cdot L)$$
$$+ n_B \cdot (E_{r_B}^{pkt} \cdot np_{r_B,T} + E_{r_B}^{SF} \cdot np_{r_B,T} \cdot L)$$
$$+ n_M \cdot (E_{r_M}^{pkt} \cdot np_{r_M,T} + E_{r_M}^{SF} \cdot np_{r_M,T} \cdot L)$$
$$+ n_F \cdot (E_{r_F}^{pkt} \cdot np_{r_F,T} + E_{r_F}^{SF} \cdot np_{r_F,T} \cdot L) \quad (13)$$

In order to estimate the average number of packets of each type of router, we need to detail the possible paths taken

by traffic and estimate probability of each path. There are three types of traffic using the telecommunication network: we denote $np_T^{DC\_DC}$, $np_T^{EU\_DC}$ and $np_T^{DC\_EU}$ respectively the overall number of packets during T using the telecommunication network between DCs, from end users to DCs and from DCs to end users. From the previous subsection, we have:

$$np_T^{DC\_DC} = \big(\sum_{i=1}^{N}(1-p)\cdot\lambda\cdot(1-q)\cdot pm^i\big)\cdot T/(8\cdot L)$$
$$np_T^{EU\_DC} = U\cdot\mu\cdot T/(8\cdot L)$$
$$np_T^{DC\_EU} = \big(\sum_{i=1}^{N} q\cdot\lambda\cdot pm^i\big)\cdot T/(8\cdot L) \tag{14}$$

Now, for each traffic type, we have to estimate the different paths that it can take and their probability in order to determine $np_{r,T}$ for all the routers ($r \in RT$). Note that while the volume is different, traffic from end users to DCs and from DCs to end users employs the same paths (in the other direction). Each path depends on the considered architecture, so we will detail them one by one for a given topology in order to determine the probabilities for each path. For the telecommunication network topology, we consider a recursive topology presented in [31] as a realistic simplified version of a national Internet Service Provider. Figure 7 illustrates this topology. To simplify the model, we assume all core routers are directly connected together.
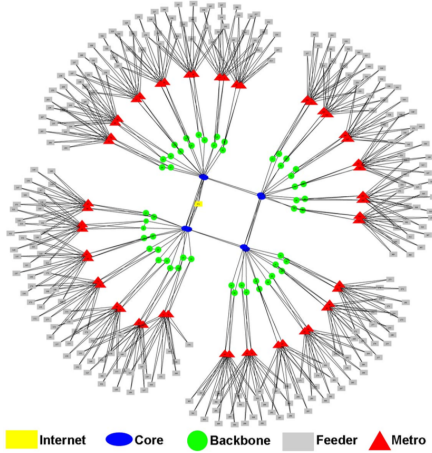

Fig. 7. Telecommunication network structure (Figure from [31])

For the FC architecture (fully centralized), the only DC is connected to one of the core routers ($p = 0$). So, there are only two possible paths for the traffic between DC and end users: CBMF or CCBMF (in case the user is not connected to the same core router), with C, B, M and F respectively standing for Core, Backbone, Metro and Feeder routers. As the core layer comprises n_C routers and the users are assumed to be evenly distributed among the feeders, the first path is taken in one case over n_C, and the second path in $n_C - 1$ cases over n_C. It means that for the FC architecture, we have:

$$np_{r_C,T}^{FC} = (1/n_C + 2\cdot {n_C-1}/n_C)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_C$$
$$np_{r_B,T}^{FC} = (np_T^{EU\_DC} + np_T^{DC\_EU})/n_B$$
$$np_{r_M,T}^{FC} = (np_T^{EU\_DC} + np_T^{DC\_EU})/n_M$$
$$np_{r_F,T}^{FC} = (np_T^{EU\_DC} + np_T^{DC\_EU})/n_F \tag{15}$$

For the PD architecture (partially distributed), DCs can be connected to Metro or Backbone routers. Table III indicates the possible paths and their probability (considering a uniform distribution of the allocated VMs).

TABLE III
USER-TO-DC AND DC-TO-DC PATHS IN A PD CLOUD; F: FEEDER, M: METRO, B: BACKBONE AND C: CORE ROUTERS

User-DC paths	Probability	DC-DC paths	Probability
FM	$1/n_M$	MBM	$1/2n_M$
FMBCBM	$1/n_C - 1/n_M$	MBCBM	$1/2n_C - 1/2n_M$
FMBCCBM	$3/n_C$	MBCCBM	$3/2n_C$
FMB	$1/n_M$	MB	$1/n_M$
FMBCB	$1/n_C - 1/n_M$	MBCB	$1/n_C - 1/n_M$
FMBCCB	$3/n_C$	MBCCB	$3/n_C$
		BB	$1/2n_M$
		BCB	$1/2n_C - 1/2n_M$
		BCCB	$3/2n_C$

Using Table III, we can compute, for the PD architecture, the number of packets processed by a router during T for each type of router:

$$np_{r_C,T}^{PD} = (14/n_C - 2/n_M)\cdot np_T^{DC\_DC}/n_C$$
$$+ (14/n_C - 2/n_M)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_C$$
$$np_{r_B,T}^{PD} = (16/n_C - 3/2n_M)\cdot np_T^{DC\_DC}/n_B$$
$$+ (16/n_C - 3/n_M)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_B$$
$$np_{r_M,T}^{PD} = (8/n_C)\cdot np_T^{DC\_DC}/n_M$$
$$+ (1 + 4/n_C - 1/n_M)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_M$$
$$np_{r_F,T}^{PD} = (np_T^{EU\_DC} + np_T^{DC\_EU})/n_F \tag{16}$$

For the FD architecture (fully distributed), DCs are connected to Feeder routers. It means that User-DC paths are identical to DC-DC paths. Table IV indicates the possible paths and their probability.

TABLE IV
USER-DC AND DC-DC PATHS IN A FD CLOUD; F: FEEDER, M: METRO, B: BACKBONE AND C: CORE ROUTERS

User-DC paths	Probability	DC-DC paths	Probability
F	$1/n_F$	F	$1/n_F$
FMF	$2n_M/n_F - 1/n_F$	FMF	$2n_M/n_F - 1/n_F$
FMBCBMF	$2/n_C - 2n_M/n_F$	FMBCBMF	$2/n_C - 2n_M/n_F$
FMBCCBMF	$6/n_C$	FMBCCBMF	$6/n_C$

Using Table IV, we compute for FD the number of packets processed by a router during T for each type of router:

$$np_{r_C,T}^{FD} = (14/n_C - 2n_M/n_F)\cdot np_T^{DC\_DC}/n_C$$
$$+ (14/n_C - 2n_M/n_F)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_C$$
$$np_{r_B,T}^{FD} = (16/n_C - 4n_M/n_F)\cdot np_T^{DC\_DC}/n_B$$
$$+ (16/n_C - 4n_M/n_F)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_B$$
$$np_{r_M,T}^{FD} = (16/n_C - 2n_M-1/n_F)\cdot np_T^{DC\_DC}/n_M$$
$$+ (16/n_C - 2n_M-1/n_F)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_M$$
$$np_{r_F,T}^{FD} = (16/n_C - 1/n_F)\cdot np_T^{DC\_DC}/n_F$$
$$+ (16/n_C - 1/n_F)(np_T^{EU\_DC} + np_T^{DC\_EU})/n_F \tag{17}$$

Finally, for the FD-CC architecture, DCs containing the CNs are located as for the FD architecture, but SNs are

considered to be located in control DCs connected to the core. While the traffic between CN and SN is internal for all the other architectures (intra-DC), it is external for FD-CC. Thus, it represents an additional traffic compared to the FD architecture. We denote λ_{SN} the average traffic sent by a CN to an SN (in bps). It means $\lambda_{SN} \leq \lambda$. The path for the DC-SN traffic is FMBC. Consequently, during T, each type of router has an additional traffic of $\sum_{i=1}^{N}(\lambda_{SN}pm^i) \cdot T/(8 \cdot L)$ packets compared to the FD architecture. For User-DC traffic, it is highly dominated by traffic between users and CNs.

Combining Equations 11 and 12 gives the total dynamic energy consumption of a Cloud-related architecture:

$$E_{total}^{dyn} = \sum_{i=1}^{N} E_{DC^i}^{dyn} + E_{Net}^{dyn} \tag{18}$$

Finally, for each architecture, its static energy consumption during T is given by Equation 3 and its dynamic energy consumption by Equation 18.

V. MODEL EXPLOITATION

This section analyzes and compares the four different categories of architectures together, using the proposed energy model.

A. Model instantiation

In order to have a fair comparison, we consider the same number of CNs for all Cloud architectures: $1,000$ CNs, for a duration of one hour. Recall that, in FD and FD-CC architectures, CNs also support SN and/or storage services as well, so CNs of these architectures are more powerful. We consider a FC architecture with one large-size DC embedding the $1,000$ CNs. The PD infrastructure includes 8 medium-size homogeneous DCs with 125 CNs each. The FD-CC architecture comprises $1,000$ *nano* DCs with 1 CN each, and 8 central small-size controller DCs with the SNs. Finally, the FD architecture, similar to FD-CC, consists of $1,000$ *nano* DCs with 1 CN each. Table V summarizes the configuration for each architecture.

TABLE V
ARCHITECTURES CHARACTERISTICS

Architecture	FC	PD	FD-CC	FD
Number of DCs (N)	1	8	1008	1000
Number of PMs ($\sum_{i=1}^{N} pm^i$)	1016	1024	1024	1000
Number of CNs ($\sum_{i=1}^{N} |CN^i|$)		1000		
Number of SNs ($\sum_{i=1}^{N} |SN^i|$)	16	24	24	0
Number of switches per DC ($|NE^i|$)	320	80	1 or 0	0
Number of core routers (n_C)		8		
Number of backbone routers (n_B)		52		
Number of metro routers (n_M)		52		
Number of feeder routers (n_F)		260		

The number of SN-based PMs is based on the number of CNs and the architecture characteristics. For a mega/micro DC in FC or PD architectures, we consider 3 global SNs (to ensure high availability) and 1 additional SN for each 100 CNs. Regarding FD-CC, we consider 3 SNs inside each of 8 small-size DCs in the core of network. For FD architecture, there is only one multi-purpose PM per *nano* DC.

To obtain realistic energy profiles for PMs, we did experiments on the French experimental testbed Grid'5000 [39] on Taurus servers (12 cores, 32 GB memory and 598 GB/SCSI storage), used for PMs of FC and PD clouds, and Parasilo servers (with 16 cores, 128 GB memory and 5*600 GB HDD + 200 GB SSD/SAS storage) for PMs of FD and FD-CC clouds. We used stress benchmark to get the consumption values per loaded core ($P_{x,j}^{cores}$). The Grid'5000 testbed provides power measurements based on external wattmeters on each of the Taurus servers with one value per second and a 0.125 Watts accuracy. Table VI shows the results of these measurements.

For FC architecture with a large size DC (*i.e.*, 1000 CNs and 13 SNs) a 16-ary Fat Tree switch topology (*i.e.*, 320 switches) is considered. As a DC of PD architecture includes 125 CNs and 3 SNs, we consider a 8-ary Fat Tree topology. In FD-CC category, a central control DC includes 3 SN-based PMs. Thus, one switch is required per DC. Regarding FD category, it does not need any intra-DC network. Concerning telecommunication network topology, as described on Figure 7, motivated by [31], we consider a real telecommunication network (*i.e.*, a simplified version of a ISP). It comprises 8 core routers, 52 backbone routers, 52 metro routers for aggregation layer and 260 feeders in access layer (*i.e.*, totally 372 routers connected with 718 links).

For VMs, we assume each VM reserves one CPU core (*i.e.*, small-size homogeneous VMs) and a VM, on average, utilizes 15% of its reserved CPU (u_{VM}). This value was measured on a real Cloud provider [34]. In addition, as there is not a visible difference between utilized CPU resource of one core of Taurus and one CPU core of Parasilo, we assume all CPU cores are similar (*i.e.*, no difference between resources of a reserved VM on Taurus and a reserved VM on Parasilo). On the other hand, finding average CPU utilization of SNs (u_{SN}) is a challenge due to the variety of workloads and number of active PMs. For this reason, we simply assume an average CPU utilization of 40% for SN-based PMs for our simulations.

For the energy profile of routers, we consider the measurements presented in [40]: $11,000$, $4,000$, $2,000$, and $2,000$ Watts for idle power consumption (P_r^{idle}) of Core, Backbone, Metro and Feeder routers respectively. On the other side, for the switches inside DCs, we consider a 16-ports switch (*i.e.*, 16-ary fat tree topology): Cisco WS-C6503 switch equipped with only one 16-port WS-X6516-GE-TX line card. It consumes around 280 Watts according to the Cisco Power Calculator [41]. In addition, inspired by [42], we consider that 70% of total traffic of a mega and micro DC remains inside it. Real measurements in [33], [36] show that there is not a visible difference between E_r^{pkt} for various routers. Therefore, we simply assume the same value of $1,300$ nJ/pkt for all types of routers and switches. Similarly, we consider E_r^{SF}=14 nJ/Byte for all network elements.

We consider a PUE value of 1.2 for all mega and micro DCs in FC and PD architectures. As a comparison, Google currently exhibits an average PUE of 1.11 on its data centers [43]. In order to eliminate effects of different energy management techniques (*i.e.*, controlling sleep and active modes of devices) on evaluating clouds energy consumption, we assume that all PMs and network elements are always active.

TABLE VI
PM ENERGY PROFILES

Number of active cores	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Taurus	97	128	150	158	165	171	177	185	195	200	204	212	220				
Parasilo	98	148	161	176	178	180	184	200	208	212	215	217	218	221	230	237	241

B. Simulation results

We consider a general scenario where $10,000$ VMs are requested by a set of end users. The VMs have been allocated and are running for all the considered time period T (1 hour). It means that 10 VMs on average are hosted per CN. Based on the proposed model, we compute the energy consumption of each architecture. The variables consist in the average traffic generated by one PM (*i.e.*, λ: 250, 500 and 1000 Mbps) and the average packet length (L: 100, 576 and 1000 bytes). Note that the packet length varies for a given amount of traffic, thus influencing the number of packets, not the total data volume.

Figure 8 shows the total energy consumption of the four Cloud-related architectures. In all cases, the FD architecture outperforms the others in terms of energy consumption. In particular, for the highest traffic rate (*i.e.*, $1,000$ Mbps), it consumes between 14% and 25% less energy than FC and PD architectures respectively. FD-CC presents performance close to FD with the considered traffic rates (with maximum $1,000$ Mbps per PM). To have a more detailed view of the results, we consider the static energy part detailed in Table VII (provided by Equation 3) and the dynamic part zoomed in Figure 9 (given by Equation 18).

TABLE VII
STATIC (INFRASTRUCTURE) ENERGY CONSUMPTION OF DIFFERENT
CLOUD CATEGORIES IN kWh

	FC	PD	FD-CC	FD
CNs	116.4 (10%)	116.4 (9%)	98 (10%)	98 (10%)
SNs	1.51 (<1%)	2.8 (<1%)	2.4 (<1%)	0.0 (<1%)
Switches	107.5 (9%)	215 (19%)	2.2 (<1%)	0.0 (<1%)
Core	88 (8%)	88 (7%)	88 (9%)	88 (9%)
Backbone	208 (18%)	208 (17%)	208 (20%)	208 (20%)
Metro	104 (9%)	104 (8%)	104 (10%)	104 (10%)
Feeder	520 (46%)	520 (42%)	520 (51%)	520 (51%)
Total	1145.3	1254.2	1022.5	1017.8

Table VII shows that the FD infrastructure consumes less energy than others. As expected, because of variety of PMs size (and therefore energy consumption), total static energy consumption of CNs in FD and FD-CC architectures is slightly more than the FC and PD architectures. Concerning SNs static energy, because of a higher number of utilized SNs, PD and FD-CC clouds consume more energy than the FC architecture in this part. However, the total static energy consumption of SNs in the FD-CC architecture is less than the FC Cloud despite using a higher number of SNs. This is because of a lower PUE for the FD-CC architecture. For the intra-DC network, the PD architecture consumes more static energy as it uses more switches than the others. Besides, the FD architecture can noticeably save energy in this part as it does not require any switch. Concerning telecommunication network routers, it is the same for all cases. We can observe that, although one Core router consumes the most, overall,

the Feeder routers' consumption greatly dominates this static energy consumption due to their number.

For the static energy consumption part, the number of intra-DCs switches plays an important role on energy consumption of a cloud-related architecture. The other influencing factors on static energy efficiency are the number of SNs, the energy efficiency of PMs, and the PUE.

Figure 9 shows total dynamic energy consumption of the four architectures. While packet length does not play an important role on dynamic energy consumption in a low traffic (*i.e.*, λ=250 Mbps), its effect at higher rates (*i.e.*, 500 and 1000 Mbps) is noticeable. Figure 9 also indicates that the FD architecture consumes less dynamic energy than others mostly because of the absence of intra-DC network.

As FD-CC and FD architectures use more energy-consuming CNs, their dynamic energy consumption due to PMs is higher than the others. Besides, FD-CC architecture uses the highest number of SN nodes, and consequently consumes more energy than others. Concerning dynamic energy consumption of switches, it is null for the FD infrastructure. FD-CC cloud consumes less energy than FC and PD architectures since FD-CC only needs few switches to connect its SNs. Although, in general, there is a negligible difference between results of FC and PD architectures, PD slightly outperforms FC when increasing traffic. It is due to longer paths between PMs in larger size data centers.

Concerning dynamic energy consumption of routers, there is a direct relation between locations of DCs and traffic's impact on energy consumption. Although FD exhibits a slightly lower dynamic energy consumption for routers than FD-CC, a larger difference between them should be expected for applications with heavy traffic exchange between controllers (SNs) and CNs.

C. Discussion

In this part, we summarize the key findings.

Telecommunication network: as we consider the entire consumption of the network between DCs and end users, its energy consumption heavily dominates the total consumption. This is consistent with literature stating that telecommunication networks constitute the predominant part (37% in 2014) in the overall ICT consumption including end-user devices [44]. Future distributed Cloud architectures could reduce the need for network routers in keeping traffic as local as possible.

Packet length: the results show that packet length, in heavy traffic rate, has an important effect on dynamic energy consumption of all architectures.

PUE: PUE greatly impacts the energy consumption of the architectures with medium and large-size DCs. Gains on the energy efficiency of ICT devices can be wiped out by a high PUE.
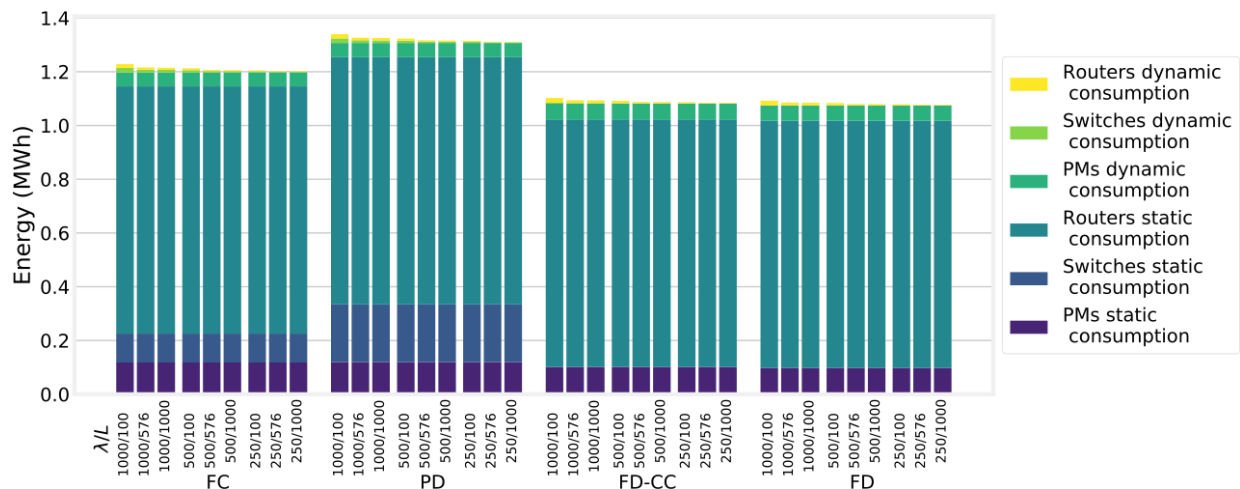
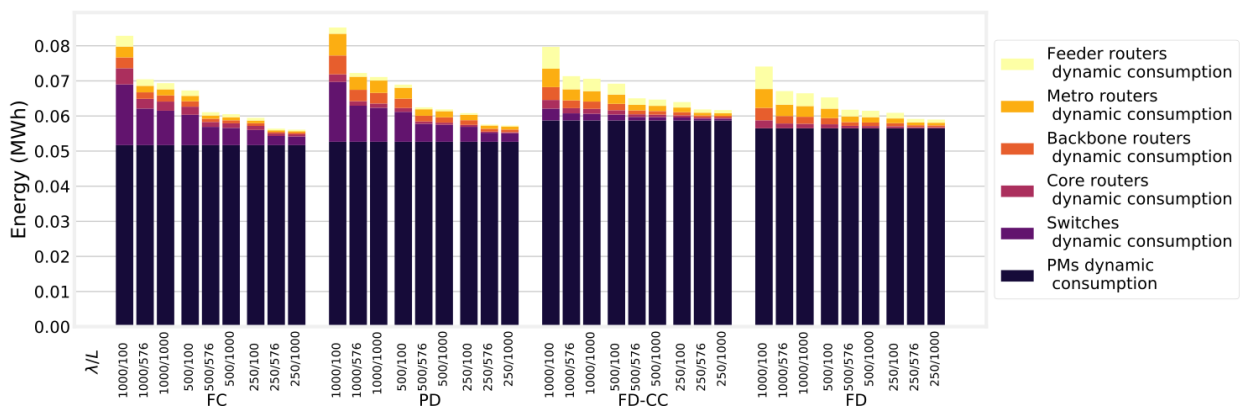Fig. 8. Total energy consumption of different architectures for 1 hour

Fig. 9. Dynamic energy consumption of different architectures for 1 hour

Energy consumption of intra-DC network: in FC and PD architectures, the intra-DC switches can consume more than 50% of total DC energy and even up to 20% of the total infrastructure energy. Selecting appropriate network architecture and switch size could significantly improve the energy efficiency of an infrastructure.

Overall energy consumption: FD architecture does not need intra-DC switches, nor DCs, and consequently, its PUE is equal to 1 (no need for air conditioning equipment). This makes FD and even FD-CC architectures more energy efficient than others. However, we consider here CNs with similar performance for all architectures. It may not be the case in some future Fog and edge computing infrastructures where heterogeneous end users devices are employed.

VI. CONCLUSION

By increasing the number of Cloud-related architectures, ranging from fully centralized to completely distributed, many Cloud researchers and providers are interested to know which infrastructure can be more energy efficient. This paper tackled this issue.

First, a taxonomy has been proposed to categorize Cloud-related architectures and to detect main topological differences between them. Inspired by this taxonomy, a comprehensive,

easy-to-use and scalable energy model has been then proposed. The model accurately highlights architectural differences between the infrastructures in terms of energy consumption.

Four different Cloud-related architectures (*i.e.*, FC, PD, FD-CC and FD) have been analyzed using the energy model. The results showed that a completely distributed architecture (FD) consumes between 14% and 25% less energy than fully centralized (FC) and partly distributed (PD) architectures respectively. Through the utilization of the Power Usage Effectiveness (PUE) metric, the proposed model captures the impact of cooling systems, that are currently non-negligible in terms of energy consumption.

As a result of this analysis, future work on greening emerging cloud architectures should focus on improving their telecommunication network use, the efficiency of their computing infrastructures (*i.e.*, PUE) and exploiting heterogeneous infrastructures to better fit the users' needs.

Although the model considers many important technical points, several improvements can be considered as future work. For example, the model currently assumes an even distribution of end users. But, in real environments, end users can be unevenly distributed. We also plan to extend the model to support other technologies (containers, VNFs) and architectures (heterogeneous telecommunication networks,

mobile networks).

References

[1] A. Shehabi, S.J. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, E. Masanet, D. Sartor, M. Herrlin, W. Lintner, "United States Data Center Energy Usage Report", Lawrence Berkeley National Laboratory, Berkeley, California. LBNL-1005775, 2016.

[2] R. Mahmud, R. Kotagiri, R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions", In B. Di Martino, KC. Li, L. Yang, A. Esposito (eds) Internet of Everything, Internet of Things (Technology, Communications and Computing). Springer, 2018.

[3] C. Mouradian, D. Naboulsi, S. Yangui, R. Glitho, M. J. Morrow, P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-art and Research Challenges", IEEE Communications Surveys and Tutorials Vol. 20(1), 2018.

[4] ETSI, "Mobile-Edge Computing - Introductory Technical White Paper", ETSI White paper, 2014.

[5] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration", in IEEE Communications Surveys & Tutorials, 19(3), pp. 1657-1681, 2017.

[6] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective", in IEEE Communications Surveys & Tutorials, 19(4), pp. 2322-2358, 2017.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, 2016.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things", MCC workshop on Mobile cloud computing, pages 13-16, 2012.

[9] The OpenFog Consortium, https://www.openfogconsortium.org, accessed December 2018.

[10] P. Hu, S. Dhelim, H. Ning and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues", Journal of Network and Computer Applications, Volume 98, pages 27-42, 2017.

[11] The Linux Foundation, "Open Glossary of Edge Computing", https://github.com/State-of-the-Edge/glossary, 2018.

[12] A. Ahmed and A. Saumya Sabyasachi, "Cloud Computing Simulators: A Detailed Survey and Future Direction", IEEE International Advance Computing Conference, 2014.

[13] F. Fakhfakh, H. Hadj Kacem, and A. Hadj Kacem, "Simulation Tools for Cloud Computing: A Survey and Comparative Study", IEEE International Conference on Computer and Information Science, 2017.

[14] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Software: Practice and Experience, vol. 41, pp. 23-50, 2011.

[15] M. Bux and U. Leser, "DynamicCloudSim: Simulating heterogeneity in computational clouds", Future Generation Computer Systems, vol. 46, pp. 85-99, 2015.

[16] A. Zhou, S. Wang, C. Yang, L. Sun, Q. Sun, and F. Yang, "FTCloudSim: support for cloud service reliability enhancement simulation", Journal of Web and Grid Services, vol. 11, no. 4, pp. 347-361, 2015.

[17] X. Li, X. Jiang, K. Ye and P. Huang, "DartCSim+: Enhanced CloudSim with the Power and Network Models Integrated", IEEE International Conference on Cloud Computing (CLOUD), 2013.

[18] A. Nez, J. Vzquez-Poletti, A. Caminero, G. Casta, J. Carretero, and I. Llorente, "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator", Journal of Grid Computing, vol. 10, no. 1, pp. 185-209, 2012.

[19] A. W. Malik, K. Bilal, S. U. Malik, Z. Anwar, K. Aziz, D. Kliazovich, N. Ghani, S. U. Khan, R. Buyya, "CloudNetSim++: A GUI Based Framework for Modeling and Simulation of Data Centers in OMNeT++", IEEE Transactions on Services Computing, vol. 10, pp. 506–519, 2017.

[20] T. Hirofuchi, A. Lebre and L. Pouilloux, "SimGrid VM: Virtual Machine Support for a Simulation Framework of Distributed Systems", IEEE Transactions on Cloud Computing, pp. 1-14, 2018.

[21] C. Heinrich, T. Cornebize, A. Degomme, A. Legrand, A. Carpen-Amarie, S. Hunold, A.-C. Orgerie M. and Quinson, "Predicting the Energy Consumption of MPI Applications at Scale Using a Single Node", IEEE Cluster, pp. 92-102, 2017.

[22] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog Computing May Help to Save Energy in Cloud Computing", IEEE J. on Selected Areas in Communications, 34(5), pp. 1728-1739, 2016.

[23] Y. Li, A.-C. Orgerie, I. Rodero, B. L. Amersho, M. Parashar and J.-M. Menaud, "End-to-end Energy Models for Edge Cloud-based IoT Platforms: Application to Data Stream Analysis in IoT", Future Generation Computer Systems, pp.1-12, 2017.

[24] C. Fiandrino, D. Kliazovich, P. Bouvry, and A. Y. Zomaya, "Performance and Energy Efficiency Metrics for Communication Systems of Cloud Computing Data Centers", IEEE Transactions on Cloud Computing, vol. 5, no. 4, 2017.

[25] T. Baker, M. Asim, H. Tawfik, B. Aldawsari and R. Buyya, "An energy-aware service composition algorithm for multiple cloud-based IoT applications", Journal of Network and Computer Applications, Volume 89, pages, 96-108, 2017.

[26] Y. Li, A.-C. Orgerie, I. Rodero, B. Lemma Amersho, M. Parashar, and J.-M. Menaud, "End-to-end energy models for Edge Cloud-based IoT platforms: Application to data stream analysis in IoT", Future Generation Computer Systems, vol. 87, pages 667-678, 2018.

[27] ISO/IEC 30134-2:2016, "Information technology – Data centres – Key performance indicators – Part 2: Power usage effectiveness (PUE)", 2016.

[28] T. Chen, X. Gao and G. Chen, "The features, hardware, and architectures of data center networks: A survey", Journal of Parallel and Distributed Computing, Elsevier, Volume 96, pp. 45–74, 2016.

[29] Cisco, "Cisco's Massively Scalable Data Center: Network Fabric for Warehouse Scale Computer", Cisco Systems, Inc., 2010.

[30] T. Wang, Z. Su, Y. Xia and M. Hamdi, "Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing", IEEE Access, vol. 2, pp. 1481-1496, 2014.

[31] L. Chiaraviglio, M. Mellia and F. Neri, "Energy-aware Backbone Networks: a Case Study", International Workshop on Green Communications (GreenComm), 2009.

[32] M. Kurpicz, A.-C. Orgerie and A. Sobe, "How much does a VM cost? Energy-proportional Accounting in VM-based Environments" Europicro PDP, pp. 651-658, 2016.

[33] A. Vishwanath, K. Hinton, R.W.A. Ayre and R.S. Tucker, "Modeling Energy Consumption in high-capacity routers and switches", IEEE Journal on selected areas in communication, 32(8), pages 1524–1532, 2014.

[34] Y. Li, A.-C. Orgerie and J.-M. Menaud, "Opportunistic Scheduling in Clouds Partially Powered by Green Energy", GreenCom: IEEE Int. Conf. on Green Computing and Communications, pp. 448-455, 2015.

[35] OpenStack Configuration Reference, [online] https://docs.openstack.org/mitaka/config-reference/compute/scheduler.html, accessed May 2018.

[36] V. Sivaraman, A. Vishwanath, Z. Zhao and C. Russell, "Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router", INFOCOM Workshops, pages 331-336, 2011.

[37] A. C. Orgerie, B. L. Amersho, T. Haudebourg, M. Quinson, M. Rifai, D. L. Pacheco and L. Lefèvre, "Simulation toolbox for studying energy consumption in wired networks", International Conference on Network and Service Management, 2017.

[38] F. Yao, J. Wu, G. Venkataramani and S. Subramaniam, "A comparative analysis of data center network architectures", IEEE International Conference on Communications (ICC), 2014, pp. 3106-3111.

[39] D. Balouek, et al.,"Adding Virtualization Capabilities to the Grid'5000 Testbed", In I. Ivanov, M. Sinderen, F. Leymann, and T. Shan, editors, Cloud Computing and Services Science, (367) of Communications in Computer and Information Science, pp. 3-20. Springer, 2013.

[40] F. Jalali, "Energy Consumption of Cloud Computing and Fog Computing Applications", PhD thesis, University of Melbourne, Australia, 2015.

[41] Cisco's power consumption calculator. [Online]. Available: http://tools.cisco.com/cpc

[42] Cisco, "Cisco Global Cloud Index: Forecast and Methodology, 2015–2020", Cisco publications, 2016.

[43] Google Data Centers Efficiency, accessed in May 2018, [online] https://www.google.com/about/datacenters/efficiency/internal/

[44] W. Van Heddeghem, S. Lambert, B. Lanoo, D. Colle, M. Pickavet and P. Demeester, "Trends in worldwide ICT electricity consumption from 2007 to 2012", Computer Communications 50, 64-76, 2014.