

Energy Efficient Cloud Service Provisioning: Keeping Data Center Granularity in Perspective

Leila Sharifi^{*,†}, Llorenç Cerdà-Alabern[†], Felix Freitag[†], Luís Veiga^{*}

^{*}Técnico Lisboa/INESC-ID Lisboa, Portugal

[†]Computer Architecture Dept., Universitat Politècnica de Catalunya, Barcelona, Spain

Email: leila.sharifi @ tecnico.ulisboa.pt, luis.veiga @ inesc-id.pt

{lsharifi, llorenc, felix}@ac.upc.edu

Abstract. The cost of power and its associated delivery are becoming significant factors in the total expenditure of large-scale data centers. Numerous techniques have been proposed to address the energy efficiency issue in cloud systems. Recently, some efforts have been made to decentralize the cloud via distributing data centers in diverse geographical positions, at different scales. In this paper, we elaborate on the energy effectiveness of service provisioning on different cloud architectures, from a mega-data center to a nano data center, which provides the extreme decentralization in terms of cloud architecture, as well as P2P-clouds or community network clouds. We study the energy consumption through an analytical and simulation framework for video streaming and MapReduce applications.

Keywords: Energy Effectiveness; cloud architecture; data center granularity; virtual data center; P2P-cloud

1 Introduction

Power saving in Information Technology (IT) industry is becoming an important issue. Data centers consume vast sums of energy not only to keep the devices running, but also for cooling the data center itself. The higher the power consumption is, the vaster the carbon emissions are. Thus, the problem is both economical and environmental. Indeed, focusing on energy management is required. Energy management is a cross-cutting concern across various disciplines including hardware design, virtualization, operating systems, and software.

Different cloud architectures, topologies, and models have been proposed, to improve either the performance or the energy consumption. However, energy and performance are contradictory goals in many cases, while they may contribute to each others' goals in some cases. For instance, the concept of distributing the data centers around diverse locations improves the service quality by providing local service resources, but induces extra complexity in terms of data center management and coordination. Loosely paraphrasing, reduced latency and increased data transmission rates are achievable by assigning requests to servers which are closer in terms of link distance. Conversely, distributing the data centers provides an opportunity to exert various renewable energy sources which contributes to a more green cloud system [1].

Increasingly, distributed data centers containing similar server machines as a mega-data center, and providing the same processing and storage capabilities, are more energy efficient due to lower cooling and internal network power consumption. Nonetheless, the distributed data center model increases the energy consumption for inter-data center communication, when data center coordination is required to obtain the services. Therefore, it is non-trivial to identify the most energy efficient model at a glance. An in-depth study on the energy consumption of each cloud element is required to obtain the most energy efficient cloud model.

Energy efficiency in clouds is studied as a threefold challenge: i) exploiting energy efficient hardware, ii) employing the hardware in an energy efficient way, and iii) applying energy efficient virtualization techniques. Computer architects and electronic engineers work on designing more energy efficient hardware. Hence, relying on the state-of-the-art hardware, we can meet the first element. Some work on energy efficient VM assignment [1–6] and migration [7] have been performed. Efforts have also been made to compare the private cloud and public cloud energy efficiency [8]. Moreover, data center network energy efficiency has been studied extensively [9–11]. However, data center granularity effects on energy efficient hardware employment and virtualization have not been widely studied.

Taking data center granularity in perspective, in this paper we answer the following question:

- Which platform is more energy efficient-traditional mega-data center or equivalent distributed data centers ?

To answer this question, we examine the energy consumption of a diverse range of cloud models, from a mega-data center to a nano data center which fulfils the extreme decentralized architecture goal. Additionally, we study P2P-cloud energy consumption. The P2P-cloud we address here is the vision of a cloud deployment in community networks, which is hosted on community-owned computing and communication resources providing services of local interest.

Comparing the energy consumption of P2P-cloud and data centers, we determine that the energy consumed for lighting, cooling, etc. in the data centers is partly dispensable in P2P-cloud. At first glance, extra energy may be consumed for routing and transferring data via the communication network. However, the study in [12] reveals that by assigning the requests to the local data centers, not only is communication cost is not increased, but also it is reduced in most cases, due to shorter paths between consumer and server. Thus, currently, cloud providers are moving towards a more distributed cloud architecture for the purpose of energy efficiency [1, 12, 13]. Furthermore, leveraging P2P-cloud improves the energy proportionality of the system because of the nature of the heterogeneous platform nature.

Nonetheless, while providing services within P2P-cloud, we may encounter performance degradation, which is a common pitfall in energy aware techniques. To tackle this issue, we formulate the *energy effectiveness* metric, in which the performance is considered while evaluating energy efficiency.

To summarize, the specific contributions of this work, aiming at forming a cloud system that meets energy efficiency goals, are listed as the following.

- First, we investigate the most eminent techniques designed to address energy efficiency support in processing and networking of clouds, following a bottom-up, distributed approach (Section 2).
- In Section 3, we model the power consumption of each cloud element through analysis.
- We devise the energy effectiveness as a combined, adaptive metric to evaluate the energy efficiency of different systems while adaptively tuning the performance level. This metric is distinguished from the state-of-the-art metrics [14], thanks to its adaptive performance guaranteeing capability, Section 3.8.
- The introduced model is evaluated via an analytical as well as an extensive simulation framework in Section 4 and 5. The distinctive theme of our analysis is in modeling the energy consumed per service rather than the state-of-the-art system centric approach.

In the next section, we explain the system models studied.

2 Background

Since in this work we analyse the energy consumption of applications under different cloud platforms, we outline in this section the elaborated cloud models and data center network topologies as well as terms and conditions used for our models.

2.1 Classic data center

In the classic data center model, as the cornerstone of the cloud computing concept, a gigantic data center embraces a number of host clusters, constituting a powerful computing or storage capacity. The internal organization and hierarchy of the data center can follow a number of variants, as we will discuss in Section 2.7, typically aiming to reduce latency and energy consumption in internal traffic.

2.2 Distributed data center

Distributed cloud architectures consist of a large number of small-sized data centers distributed across diverse geographic locations. This architecture is appealing for network service providers who already have the necessary distributed facilities such as distributed offices that are geographically dispersed to be closer to users, since they can develop a large number of distributed data centers interconnected by high-speed networks.

Distributed cloud architectures [15] can provide several benefits over the traditional centralized cloud architectures. With distributed data centers, requests can be served locally, this helps reduce

network capacity needs, for high-bandwidth applications, which constitute a significant cost when accessing centralized data centers. Distributing the data centers also reduces the latency of access compared to traditional data centers; distribution of data centers has become common with content delivery networks. e.g. Akamai [16]. In fact, the access latency of the traditional data center may have large variations due to the longer path lengths and the need for frequently switching among multiple service providers.

Increasingly, the challenges of supporting business continuity in a cloud environment are not limited to a physical area or data center. The elasticity and flexibility of the network architecture must be addressed as well. Therefore, the computing, storage, and network components used for cloud computing should not all reside in the same physical location. These resources could be spread over multiple locations and interconnected using a transparent transport mechanism that maintains security and end-to-end segmentation. A Distributed cloud data center, in addition to bringing high availability and disaster recovery, provides the opportunity to use different, local energy sources.

Federated cloud features the same architecture as the distributed data center. The only difference is in providing the resources through the aggregation of several providers in the federation, while all the infrastructure remains under the control of a single provider in the distributed data center model. Thus, regarding topology, federated clouds are distributed data centers, only subject to different administrative and organizational domains.

2.3 Nano data center

To take further advantage of distribution, the concept of the nano data center (NaDa) [17, 18] takes to the extreme the distribution already present in the concept of distributed data center architecture. In a nano data center, each host stands for a data center that connects to another nano data centers via the Internet. The advantage of nano data center model is that it is exempted of any effort for cooling or maintaining the data center. Moreover, if location-aware resource allocation mechanisms are applied, the nano data center model performs better than the classic data center model, thanks to the distributed architecture.

NaDa's goal is to combine the power of data centres with the scalability of peer-to-peer (P2P) methods, while not threatening the robustness and the stability of the Internet. While most research worldwide is focused on increasing the efficiency of network architectures deployed around the network backbone and data centres, NaDa is taking a radically different approach, in which network is untouched and the content is located on customer premises.

To combine all unused edge resources, NaDa attempts to use a new, managed P2P communication architecture. The P2P paradigm aids in the deployment of new services such as file sharing or IP-based telephony because it avoids having to scale servers for peak capacity. However, most of the currently deployed P2P systems have been focused on simple file sharing or streaming applications (and often for illegal content). Thus, several fundamental issues must be addressed in order to reinvent the P2P paradigm for the NaDa system.

2.4 Micro-cloud

The micro-cloud concept is introduced by the Supermicro company [19]. Supermicro offers a micro-cloud system as high-density, rack-mounted server units each populated separately with CPUs, memory, and SSD drives. These compact racks can be connected on a rack and can extend the computing power of the cloud with less complexity than that of the classic data center infrastructure. Micro-cloud items can be exploited in server centric data center networks, such as BCube [20] to build a modular data center. Modular data centers have some benefits compared with the data center facilities directly built from server racks: these include shorter deployment time, higher system and power density and lower cooling and manufacturing cost. However, it is difficult or even impossible to service a modular data center once it is deployed.

2.5 Edge(Fog) computing

The term fog computing [21] is introduced by Cisco Systems as a new paradigm to support wireless data transfer via distributed devices in the Internet of Things (IoT). Its hierarchical distributed architecture

extends from the edge of the network to the core to provide a geo-distributed platform that improves location awareness and latency. These features suit the IoT platform best, because the big is in the number of data sources rather than the data volume in IoT.

Edge(Fog) Computing is pushing the frontier of computing applications, data, and services away from centralized nodes to the logical extremes of a network. It enables analysis at the edges (i.e. at the source of the data). Thus, services would be located closer to the end-user to improve on latency and data access compared with those of the data center model. However, Fog computing is not a substitute for the cloud, it is a complement. There is a fruitful interplay between cloud and fog, particularly when it comes to big data management and analytics. By controlling data at various edge points, Fog computing integrates core cloud services with those of a truly distributed data center platform. With Fog services, we are able to enhance the cloud experience by isolating user data that is required to live on the edge. This infrastructure still maintains the concept of the cloud while incorporating the power of Fog computing at the edge.

Fog computing extends direct cloud services by creating an edge network which sits at numerous points. This dense, geographically dispersed infrastructure, helps in numerous ways. Big data and analytics can be performed faster with better results. Administrators are able to support location-based mobility demands and not have to traverse the entire WAN, in this way it provides true support for mobility and the IoT. Not only does Fog computing improve user perceived performance (latency, throughput), it also helps with security and privacy issues. These edge (Fog) systems would be created in such a way that real-time data analytics become a reality on a truly massive scale.

Fog computing is applicable in geo-distributed applications with very low and predictable latency (e.g. pipeline monitoring application, mobile applications and large scale distributed control systems, e.g. smart grids).

2.6 P2P-cloud

The concept of the P2P-cloud is the extended idea of nano data center that replaces the expensive server machines with commodity hosts that are combined with the edge computing platform. A P2P-cloud embraces a set of commodity hosts, including IoT boards, laptops and PCs, connected via a wireless communication platform as depicted in Figure 1. The main goal of the P2P-cloud is to take advantage of distributed data center hosts as well as exploit the commodity hardware of community networks [22]. Hence, they are also often described in the literature as *community network clouds* or *community clouds*, although this last notion should not be confused with NIST [23] definition of community clouds, that addresses closed multi-tenant infrastructures hired or owned by different tenants with shared concerns (e.g., mission, security, compliance) to reduce overall costs due to sharing.

The P2P-cloud topology we address in this paper is the vision of a cloud deployment in community networks: a cloud hosted on community-owned computing and communication resources providing a diverse range of services. Community networks represent the collaborative effort of community members, for building an ICT infrastructure with commodity devices in a bottom-up approach, in order to meet demand for the Internet access and services [24].

The P2P-cloud benefits from the geo-distributed infrastructure such as nano data centers and edge computing platforms, which contributes to location awareness and reduced communication latency for the locally provided services. However, it differs from other edge computing efforts, because it actually shares the available resources among users.

Comparing P2P-cloud with desktop grid [25], we discover that desktop grid is a peer-to-peer volunteer computing platform. However, P2P-cloud services are not confined simply to computing. Moreover, the concept of P2P-cloud may be mixed up with mobile cloud or cloud offloading. Namely, P2P-cloud is a broad concept that embraces all the above mentioned concepts. To exemplify, P2P-cloud hosts may be mobile or static. P2P-cloud reinforces the concept of telco-cloud or telco-backed cloud [26], because communication and IT infrastructures akin to the community network are required to develop a P2P-cloud.

In a P2P-cloud, energy is substantially consumed at hosts, switches, routers and network devices. Compared to the classic clouds, in communities, we encounter much reduced static energy waste, because the machines which do not serve the community may already be on to serve the users' individual applications. Moreover, the Idle to Peak power Ratio (IPR) for the current P2P-cloud hosts is close to the ideal case, and the PC machines consume slightly less energy compared to data center servers.

Increasingly, in P2P-cloud, to alleviate the energy consumption, requests can be assigned to the closest available host in the community. According to this fact, we introduce the P2P-cloud topology as

a set of community hosts scattered within vicinities and communicating via the wireless communication network as depicted in Fig. 1. Each vicinity can access the others via the Internet.

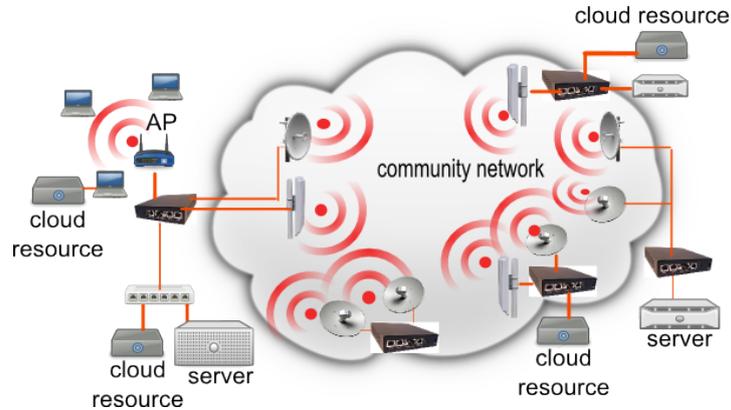


Fig. 1: P2P-cloud intra-vicinity model

This P2P-cloud model suits the locality of services more than classic clouds. Loosely paraphrasing, in this model, each host is adaptable to a specific architecture, configuration and service according to the most prevalent requests it receives. This idea is inspired from the peer-to-peer content and location aware overlay construction [27–29].

Studies reveal that virtually all the requests a user issues for the service, in a specific location, are similar to the others due to the locality of requests [30]. The P2P-cloud can leverage this fact by adjusting the service and computing capabilities of each individual community nodes accordingly, by responding to high resource demanding requests via the federation on more powerful machines such as core i7 PCs, or forwarding them to the classic cloud.

2.7 Intra-data center topology

With increasing computing power in data centers, I/O (a significant part of it in networking) becomes a bottleneck against performance. Conversely, as servers become more energy proportional, the data center network dissipates a remarkable fraction of cluster power. The energy consumption of different structures of data centers with emphasis on energy efficiency of data center architecture has been studied in [9, 10].

Data center topologies, typically, are classified into two categories of switch centric and server centric architectures. The efforts in the design of electronic data centers have succeeded to mitigate many dilemmas and obstacles in providing switch centric architectures which significantly rely on the expensive high-performance switches that can support fault tolerance, load balancing, and agility, and are able to handle high oversubscription ratios. Switch-centric category includes conventional tree topology, Fat-tree [31], VL2 [32], QFabric [33], etc.

Alternatively, server centric data centers, on the other hand, use servers as relay nodes, and provide an infrastructure with low diameter and high capacity, while employing commodity switches in order to support different traffic types for applications with intensive computing requirements. Nevertheless, additional wiring cost and complexity are a result of having servers equipped with more than one port. Server-centric topology embraces Bcube [20], Pcube [34], Camcube [35], etc., as will be explained in the rest of this section.

Three-tier architecture as depicted in Figure 2 is the classic network architecture that embraces three layers of core, distribution and access. Core layer infrastructure and topology is designed to fulfill the high-performance forwarding, while distribution and access layer offer lesser performance and more complicated topologies but more commodity hardware.

Fat-tree topology [31], consists of servers, edge, aggregate and core switches. A set of edge and aggregate switches are connected as a Clos topology [36] and form a complete bipartite in each pod.

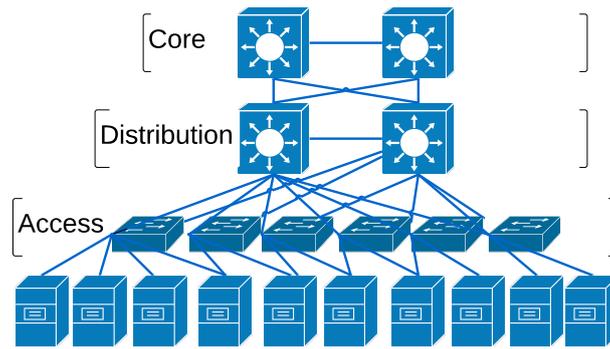


Fig. 2: Intra-data center hierarchical communication model

Each pod is connected to all core switches forming another bipartite graph. Fat-tree IP addresses are in the form of 10:pod:subnet:hosted. With Fat-tree topology issues with oversubscription, costly aggregation, fault tolerance, and scalability are resolved. Most of the switch centric architectures are based on Fat-tree.

VL2 is designed based on the Fat-tree topology [32]. However, core and aggregate switches in VL2 implement a Clos topology. VL2 relies on commodity switches since it employs Equal Cost Multi Path (ECMP) for packet forwarding and topology updates. It implements valiant load balancing by randomly selecting an intermediate switch before forwarding a packet and applies flow hashing in ECMP.

Qfabric provides one-hop paths with no oversubscription and congestion as a result. It is introduced by Juniper [33] to simplify the data center management via reducing the number of switches. Flattening three-tier tree structure to one tier fabric is a solution proposed for modern data center architecture to tackle the oversubscription, costly aggregation, fault tolerance and energy efficiency in the data center network by reducing the number of switches and path length among servers. These kind of architectures are called energy efficient data center topologies due to the energy saving on both static power, by reducing the number of switches; and dynamic power, by reducing the path length. Flattened-butterfly [37] is following the same goal by flattening the butterfly data center network topology. FlatNet [38] is another double layer flattened topology, that attempts to fulfill the flattened data center topology.

BCube and PCube BCube [20] is a server centric data center network architecture which relies on a recursively defined structure consist of two types of devices: multiport servers, typically no more than four ports, and multiple layer of commodity switches. In BCube, servers are not only hosts but also relay nodes for each other and take part in traffic forwarding through multiple parallel short paths. The design is driven by demands for intensive computing and higher bandwidth requirements to support applications for different traffic patterns. BCube supports and accelerates all types of traffic patterns and provides high network capacity due to its low diameter. The benefits of BCube design includes providing fault tolerance and load balancing, while requiring lower cooling and manufacturing cost.

PCube [34] is an elastic data center structure designed to optimize power efficiency through scaling energy consumption according to traffic demand patterns. PCube powers on/off some switches dynamically according to the network bandwidth demand. Adjustment is accomplished in a way that there are always multiple paths between any two pair of servers to obtain fault tolerance. It can be directly applied to existing hypercube structured data center networks, e.g. BCube, without any hardware modification or rewiring.

DCell is a scalable server centric data center network topology architecture providing rich connectivity among servers via commodity switches [39], but additional wiring is required to connect the servers and switches, specially in long paths. DCell is scalable due to its recursive structure, which allows extending network gradually without extra rewiring or address changing.

CamCube is a pure server-centric data center network architecture designed for container data centers [35] based on torus topology to connect a single server with six other servers using network interfaces. Network traffic routing is performed by servers while network switches are not used at all in the CamCube data center network architecture. The CamCube uses different routing schemes for different services and provides a load balanced flow assignment. It eliminates the use of similar routing logic for all services, and facilitates the use of customized routing logic for individual services.

Optical data center network Since optical network devices consume less power with orders of magnitude compare to electric networks, some efforts has been made to move the data center networks toward optical technology, which provides high speed, bandwidth and less complexity. Some topologies such as C-through [40] and Helios [41] are designed to upgrade the current data center network topology based on commercially ready optical network devices, combined with the electric switch centric architecture.

In a nutshell, optical data centers can be more energy efficient compare to the other topologies. Moreover, flattening the data center topology can reduce the energy consumption thanks to fewer number of switches as well as reduced network diameter and average path length [37, 38].

3 Energy Model

To analyze the energy consumption in clouds, first we should identify the main sources of energy consumption. The key sources include hosts and communication devices for P2P-cloud. For classic data centers this list extends to the cooling, lighting and maintenance energy as defined in the data center Power Usage Efficiency (PUE) [42]. To obviate the power efficiency of a data center, PUE parameter is defined as the ratio of total amount of power used by a computer data center facility to the power delivered to the computing equipment. The higher the PUE, the less power efficient the data center is. In the rest of this section, we analyse the energy consumption of each element.

3.1 Power consumption in hosts

Power consumption in a host machine is divided into two parts: static and dynamic power consumption. Static power is consumed even if the machine is idle, while the dynamic power is proportional to the resource utilization within the host. Overall, the power drawn in a host P_{host} is a combination of the static power P_s and dynamic power $P_d = (P_{Max} - P_s) \times U$. P_{Max} indicates nominal power as the maximum power device can dissipate at utilization level U . In case of virtualization and application collocation, there are some techniques to estimate the utilization portion induced by a specific VM [43, 44].

$$P_{host} = P_s + (P_{Max} - P_s) \times U \quad (1)$$

In (1) we assume a linear correlation among the utilization level and the power drain in the host, which is known as ideal power model with Linear Deviation Ratio (LDR) of one. However, in real systems, the LDR is not equal to one. LDR is discussed in Section 3.2. In case of community networks, hosts are commodity machines. The energy consumption is trivial comparing with the data center hosts. By sharing part of their computing resources, the users are contribute to the community. Indeed, idle energy consumed in community networks is much lesser than the data center hosts because: 1) community network hosts are not exclusively on to serve the community, thus leading to higher effective utilization of energy, 2) and depending on the type of community network host, static energy P_s consumed is lesser than 40W which is twice lesser than the server machines in data center. P_s for common servers in a data center is above 100 watts.

3.2 Energy Proportionality

The vision of energy proportional system implies the power model of an ideal system in which no power is used by idle systems ($P_s = 0$), and dynamic power dissipation is linearly proportional to the system load.

LDR indicates the maximum difference of the actual power consumption, $P(U)$, and linear power model over the linear power model as in (2).

$$LDR = \max \frac{P(U) - (P_s + P_d)}{P_s + P_d} \quad (2)$$

IPR is the indicator of idle to peak power consumption as illustrated in (3).

$$IPR = \frac{P_{idle}}{P_{Max}} \quad (3)$$

To measure how far a system power model is from the ideal (energy proportional) one, Proportionality Gap(PG) [45] is defined as the normalized difference of the real power value and the ideal power value, which is indicated as $P_{Max} \times U$, under a certain utilization level as shown in (4). Therefore, having proportionality gap values for a given device, we can reconstruct the power model of the device.

$$PG(U) = \frac{P(U) - (P_{Max} \times U)}{P_{Max}} \quad (4)$$

Given the state of the art hardware, designing individual servers which are fully energy proportional remains an open challenge, power model of an Intel Xeon server machine with 8 physical cores is illustrated in Figure 3- this figure has been obtained using values reported in SPECpower_ssj2008 results [46]. However, even in the absence of redesigned hardware, we can approximate the behavior of energy proportional systems by leveraging combined power saving mechanisms [47] and engaging heterogeneous commodity devices combined with powerful server machines in lieu of homogeneous server hardware platform [45].

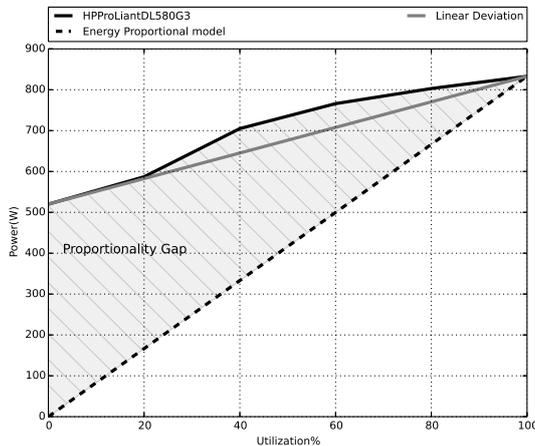


Fig. 3: Energy Proportionality Gap of a Xeon Server

On the other hand, introducing energy proportional servers, the data center network can dissipate a significant fraction (up to 50%) of data center power [37]. Therefore, introducing more energy proportional networks for data center intra connections is necessary. Flatten network topologies such as FlatNet [38] and Flattened butterfly [37] are inherently more power efficient due to lesser number of switches and required hardware. In [37], Abts et al. exploited the characteristics of modern plesiochronous links to adjust their power and performance envelopes dynamically.

3.3 Data center power consumption

In classic data centers power P_{DC} is consumed not only in hosts, but also for intra-data center communication, P_{DC_comm} , cooling, $P_{cooling}$, lighting, etc. which are affecting the PUE parameter within a data center, as in (5).

$$P_{DC} = \sum_{number\ of\ hosts} P_{host} + P_{DC_comm} + P_{cooling} \quad (5)$$

The energy overhead of all the non-IT devices such as cooling, lighting, etc. is typically modeled as $\frac{(PUE-1)}{U}$ coefficient of the overall power of the resource [8, 48], where U represents the utilization of the resource.

For cooling, embedding simple fans or chiller systems in data center aisles are common techniques. Fans use fixed power all the time, while the power consumption in chillers is tightly coupled to the aisle temperature, since chiller systems adjust temperature to predefined values. Most data centers use chillers or air conditioning units to cool things down, requiring 30-70% overhead in energy usage [49]. Therefore, water based cooling is becoming more prevalent, since water can hold more heat than air.

Doyle et al. [12] analyzed the power drawn for different cooling techniques. We refer to them for the water cooling model power consumption as illustrated in (6). Here we assumed that all the hosts are online all the time, and no host is off. Therefore, we have a constant number of hosts available to serve all the time. In (6), $T_{current}$ refers to the current temperature of the computer room, T_{safe} and T_{max} respectively denote the maximum tolerable temperature for the hosts and the maximum temperature of the server inlets in the data center, T_{safe} is defined as $65^{\circ}C$; however, recently Google managed to save in cooling cost by increasing the safe temperature to $80^{\circ}C$. C is the ratio of the heat removed to the work acquired to remove this heat, which can be interpreted as the efficiency of the cooling system. P_{fan} stands for the power dissipated in the rack fan.

$$P_{cooling} = \frac{\sum_{number\ of\ hosts} P_{host}}{C \times (T_{current} + (T_{safe} - T_{max}))} + P_{fan} \quad (6)$$

Besides efficient cooling techniques, some methods are introduced to control the rack temperature by exerting thermodynamic models which combine air flow control and thermal aware VM scheduling. Thermal aware scheduling takes the rack outlets temperature into account for scheduling the VMs [50]. Generally, it is easier to cope with the cooling issue in the smaller data center, since the heat generated by hosts and switches aggravates the temperature rising due to the more hosts and switches they have. Therefore, the smaller the data center is, the lesser the PUE value is easier to achieve.

3.4 Data center topology

In the switch centric communication within a data center, switches that connect the hosts are the major power consumption sources. In the pure server-centric data center networks, servers are in charge of forwarding the data; thus, communication energy is added to server energy profile further to the processing energy. For the hybrid network topologies, communication energy is partly dissipated in the switch and partly in the servers. Moreover, the network topology impacts the power usage profile.

Here we study the power consumption of a three-tier, hierarchical topology. The motivation behind formulating a hierarchical model is that it can be easily generalized to numerous intra-data center topologies (e.g. Fat-Tree [31], VL2 [32], BCube [20], PCube [34], etc.). The tree depth is defined based on the path messages that should traverse within the data center in each layer. For the topologies which deviate from this property, such as CamCube [35], we analyse the energy model separately. We assume an l level tree in which hosts are in the leaves and are connected to an edge switch as their predecessor via Gigabit Ethernet links. The edge switches are connected via an aggregate switch; this process proceeds in two or more levels to create the root of the tree as shown in Figure 2.

To assign a task to a host, the root aggregate switch transmits the task data to the selected host through the tree. Assuming the homogeneous switches in each level of the tree, the power consumed for this purpose is calculated as in (7). P_{switch} stands for power drawn by the switch. Additionally, we added P_{host} to each level consumption to generalize our model.

$$P_{DC_comm}^{intra} = \sum_{i=1}^{l-1} (P_{switch}(i) + P_{host}(i)) \quad (7)$$

Therefore, in a switch centric model, $P_{host} = 0$, while in a pure server centric model $P_{switch} = 0$ and in a hybrid model, power is drawn both in switches and servers.

Referring to (7), the depth of the tree, l , directly influences the power efficiency of the data center. The tree depth is determined by the number of hosts and network topology. The larger the data center is, the greater the number of switches and links required to connect the hosts and the deeper the tree is. Furthermore, flatter data center topologies, such as flattened butterfly [37] and FlatNet [38], obtain shorter paths via less switches. Topologies providing smaller network diameter are also more energy efficient due to shorter average paths to be traversed among the servers.

Therefore, smaller distributed data centers, serving the users independently, are more power efficient than a single mega-data center model, following a tree intra-data center topology. Loosely paraphrasing,

in small data centers, the network diameter is smaller, because the number of switches and links required to connect the hosts within a data center is directly related to the number of hosts. Hence, the path that should be traversed to reach a host contains fewer hops compared to a data center with more hosts.

CamCube [35], as a pure server centric system that conforms to a three-dimensional torus topology, does not follow the tree architecture. Nonetheless, we can model the network energy consumption of this model as the forwarding power drawn in the servers due to forwarding the data through the path between source and destination servers $P_{DC-comm} = \sum_{i=0}^{num-hops} (P_{host}(i) \times U)$. U indicates the utilization induced by the switching process.

3.5 P2P-cloud Power consumption in communication infrastructure

As described in Section 2.6, we assume a P2P-cloud deployed in a community network. Inexpensive WiFi devices have fostered the deployment of such communities in recent years. Some significant examples are Guifi.net [51], with more than 20,000 active nodes, Athens Wireless Metropolitan Network [52], FunkFeuer [53], Freifunk [54], etc.

In these networks, hosts within a vicinity are usually connected via wireless links that form a wireless network. Thus, the power consumed for communication within a vicinity predominantly embraces the wireless network power consumed to transmit data [55].

Community networks are rather diverse in terms of size, topology and organization. This is a consequence of their unplanned deployment, based on the cooperation of their own customers. The characterization of the power consumption of these networks is therefore challenging, and, as far as we know, has not been done before. In this paper, we tackle this problem by evaluating experimentally the power consumption of a production community network in Section 4.4.

3.6 Internet Power consumption

P2P-clouds for inter-vicinity communication and classic data centers for communication with users rely on the Internet. Thus, to analyze the energy consumption of these systems, we should be aware of Internet energy consumption as well. Power drawn in the Internet is subject to the hardware and distances exploited. Internet infrastructures are classified as core, distribution and access. The core layer includes Internet backbone infrastructures such as fiber-optic channels, high speed switch/routers, etc. Distribution infrastructure plays role as an intermediary to connect the ISPs to the core network. The access layer constitutes the user-to-ISP communication infrastructure.

Since there is a diverse range of hardware in each layer, it is not trivial to form a comprehensive analysis on energy consumption of the Internet. However, Baliga, et al. [56] conducted a study on the prevalent Internet hardware energy consumption. We rely on this study for the Internet power consumption component of our analysis by driving the model in (8). In this model, $P_{Internet}$ stands for Internet power consumption which is a combination of power drawn in core $P_{core} = \sum_{P_{router, core}}$, distribution $P_{dist} = \sum_{P_{router, dist}}$ and access $P_{access} = \sum_{P_{router, access}}$ levels. $P_{router, core}$, $P_{router, dist}$ and $P_{router, access}$ denote router power consumption in core, distribution and access level respectively.

$$P_{Internet} = P_{core} \times n_{core}^{hops} + P_{dist} \times n_{dist}^{hops} + P_{access} \times n_{access}^{hops} \quad (8)$$

3.7 Oversubscription

Oversubscription in data center network is the ratio between the servers bandwidth to the total uplink bandwidth at the access layer [56]. Hence, when approaching the aggregation and core layer, the number of servers sharing the uplinks increases, and thus the oversubscription ratio also increases and results in bottlenecks. Oversubscription limits the server-to-server capacity. Increasingly, congestion resulting from oversubscription could also lead to overloading switch buffers which will start dropping packets. The concept of oversubscription exists in Internet communication, where Internet service providers (ISPs) exert it as a strategy to utilize the resources by overbooking the shared infrastructure among users. The more the resources are shared temporally, the less the energy consumption is due to the shared static power dissipated. Since the communication devices are not energy proportional, the amount of static power is still remarkable (i.e. minimum of 10%- 30% is the leak power, drawn regardless of the utilization of the devices). Therefore, oversubscription divides this power among the

consolidated traffic. Moreover, since LDR is decreasing and the dynamic power dissipation in the devices is approaching the linear model, we can say that the overall power consumption decreases due to static power division. Oversubscription, ϕ , in general, can contribute to the power saving if it meets the following condition: $\frac{P_s}{\phi} + P_{U_i} - P_{U_{i-1}} < P_s + P_{U_i}$.

Oversubscription for the home users is 40:1 and for the business connection is approximately 20:1 in the current Internet. Hence, we consider these values for Internet oversubscription in our evaluation framework.

3.8 Energy Effectiveness

In the process of decreasing the energy consumption for cloud services we may end up with the pitfall that energy consumption decreases remarkably, but the performance is violated simultaneously. Users may leave a system if they do not receive the desired quality of service (QoS). Indeed, we need to develop a comprehensive framework to provision QoS for a diverse range of services and applications using collaborative environments.

If the service is not delivered as expected, it may tarnish the provider's reputation. Thus, obtaining a service with desirable response time as well as acceptable throughput, availability and consistency level is required. Attaining high QoS may impose additional energy consumption. Therefore, we should strive to alleviate the burden of high service energy. To this end, the energy efficiency parameter is introduced [14] as $\frac{Performance}{Energy}$. However, in this metric, no mechanism will guarantee the performance, and all sensitive and non-sensitive services are treated equally.

The concept of **energy effectiveness**, as a middleware metric, seems to suit better to achieve this end of conciliating two goals. Thus, Energy Effectiveness, as we propose in (9), quantifies the degree to which the ecosystem is successful in decreasing energy dissipated while ensuring that the QoS is not significantly violated (i.e. fulfilment above a reasonable margin, such as 95%).

$$EE = \alpha \times \frac{E_{ep}}{E_{actual}} + (1 - \alpha) \frac{Perf_{actual}}{Perf_{standard}} \quad (9)$$

In (9), EE introduces the energy effectiveness, E_{actual} , and $Perf_{actual}$ represent the estimated or measured energy consumption and performance of the considered service on the running platform. E_{ep} factorizes to the energy consumed to provide the service in an energy proportional system. $Perf_{standard}$ is quantified based on the Service Level Objectives(SLO) and Service Level Agreement(SLA) parameters. Quantifying the SLA metrics is extensively studied in [57].

Moreover, it is necessary to handle the trade-off among these tightly coupled parameters to achieve an efficient mechanism. Intuitively, an adaptive model that covers the system and user requirements, is appropriate for this purpose, because the parameters are tunable in such a model. The model supports a more diverse range of cases due to its flexibility. Therefore, we introduce α as the adaptiveness parameter. Based on the performance sensitivity of the applications, we can tune the α in the range of 0 to 1 to weight the energy and performance accordingly.

Further to describing energy effectiveness, we define **Responsiveness Factor**, RF , which embodies the range of variability in the energy effectiveness as $RF = \frac{\partial EE}{\partial \alpha}$. Namely, RF represents the slope of the EE equation when α varies in the range of 0-1. The higher the RF, the more influence the adaptiveness factor has on the EE value, and more important it is to establish it properly. RF can be determined in the SLAs according to the user incentives (previously addressed for cycle-sharing [58]) and service requirements (previously addressed for virtual machines [59] and Java applications [60]).

The energy effectiveness metric we define here has a bounded value in the range of 0 to 1 for the sequential processing and interactive applications such as live streaming, while this value can exceed one for parallel processing applications (e.g. MapReduce). This value tightly couples with the level of parallelism and the energy proportionality of the host platform. Quantifying the correlation of the parallelism and energy effectiveness is beyond the scope of this work, but interested readers may refer to [61] for more information.

4 Evaluation Framework

4.1 Experiment Setup and Scenarios

We classified the studied data center architecture into four categories, as specified in Table 1. In each scenario of Table 1, we consider the same number of hosts in total. The purpose for having 32 hosts per

Table 1: Studied data centers

Type	hosts/cluster	clusters	data centers	hosts/DC	PUE
Nano	1	1	2^{20}	1	1
Medium	32	32	1024	1024	1.032
Large	32	1024	32	2^{15}	1.11
Mega	32	32768	1	2^{20}	1.33

Table 2: VM Specifications

Type	Cores	Memory (GB)	Storage (GB)
Small	1	1	1
Medium	1	3.75	4
Large	2	7.5	32

cluster in all scenarios is to tune the cluster size so that each rack with a top-of-rack switch stands for a cluster. Furthermore, the cluster size is independent of data center size to facilitate the comparison of scenarios. We define the PUE according to the data center granularity. The bigger the data center, the higher the PUE value.

Additionally, we elaborate the energy consumption in the P2P-cloud under the same workload condition. We aim to analyse the energy consumption for different cloud models under the video streaming and MapReduce workload with the following configuration. Running MapReduce on P2P-cloud matches the idea of a desktop Grid [25].

Three VM types, as shown in Table 2, are exerted.

For most scenarios, we assume a typical workload of input data size of 3 GB; overall intermediate output size is 30% and the final output size is 20% of the original input. For the sake of comparison through out this evaluation, we take small VMs to execute the tasks, unless the size is explicitly mentioned.

4.2 Host Power Model

The nano data centers and data center hosts are set as HP servers. For the HP machines, the power model is derived through loading the machine with different loads, collecting performance counters, and applying regression; the corresponding power-to-utilization values are depicted in Figure 4.

For the P2P-cloud nodes, we employ the Jetway JBC362F36W with Intel Atom N2600 CPU with a maximum power of 20 W. Additionally, Dell Optiplex 7010 machines are employed with Core i7 CPU, and four cores supporting up to eight parallel threads. We derive a power model for these devices, as shown in Figure 4)b,c by utilizing the machines in different levels, using Stress tool [62]. The portion of Dell machines in the testbed is 20%, while Jetway devices make up 80%.

4.3 Data center Switch Power Model

Different topologies covering switch centric and server centric systems have been studied and simulated using power consumption values of switches that are available in the market. For the core switch, we opt for Cisco Nexus 5596T, which has 32 ports of 10 Gbit Ethernet, and supports optical networking due to SPF+ ports. It typically dissipates 900 W, while the maximum power is 1100 W.

For the distribution and access layer switches, we rely on the Cisco Nexus 2232TM switch, which has 32 ports of 1 Gbit and 10 Gbit Ethernet, with the over subscription of 4:1. Its maximum power consumption is 386 W; nonetheless, it draws 280-350 W, typically. For commodity switches, we employ Cisco Catalyst 37590-48TS, which consumes the maximum of 75 W and provides 48 ports. For all the switches, because they include the recent technology of green switches, the power drawn for each port, in idle case, is almost zero. **N.B:**All the above values are derived from the devices datasheet.

The power consumption of a server's port is set to be 3 W [11]. Internet energy consumption values are derived from [56], which characterizes the metro power of around 10.25 W and core power of less than 0.15 W per connection through a fast Ethernet link, with the over subscription of 40:1.

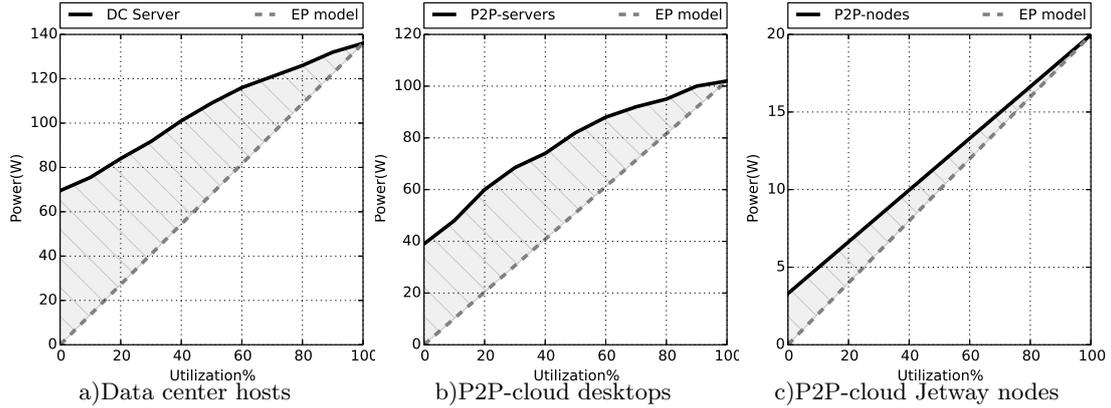


Fig. 4: Power Model of different processing elements

4.4 Wireless Communication Power Model

We characterize the power consumption in the P2P-cloud by means of experimental measurements in a production wireless community network. The network consists of approximately 50 802.11n-based nodes. It was first deployed in a city quarter in Barcelona, Spain, called Sants, as part of the *Quick Mesh Project* (QMP) [63]. Then, nodes at *Universitat Politècnica de Catalunya* (UPC) joined the network supported by the EU CONFINE project [64]. We shall refer to this network as *QMPSU* (from Quick Mesh Project at Sants UPC). QMPSU is part of a larger community network having more than 20.000 operative nodes called Guifi.net [51]. An experimental evaluation of QMPSU can be found in [65], and a monitoring page is available online [66].

Typically, QMPSU users have an outdoor router with a Wi-Fi interface on the roof, which establishes wireless links with other users in the neighborhood. Additionally, the outdoor router has an Ethernet interface connected to an indoor AP as a premises network (see Figure 5).



Fig. 5: QMPSU connectivity

From the QMPSU graph formed by the outdoor routers we have obtained an average path length of 3.78 hops; thus, 4.78 outdoor routers are crossed. Therefore, the average power consumption of a transmission between a pair of nodes in the network is:

$$P_{WN} = 2 P_{AP} + 4.78 P_{router}, \quad (10)$$

Where P_{AP} and P_{router} are the power consumption of the AP and outdoor routers, respectively. To asses these powers, we have proceed as described next.

The most common outdoor router used in QMPSU is the Ubiquiti NanoStation M5 (NS)¹. As an indoor AP, we have considered the TP-LINK WDR4300². We installed the Linux distribution openWRT [67] in these devices, which is used in QMPSU, and experimentally measured the devices' power consumption. To this end, we used a voltage and current sensors, and took power samples using an Arduino type board³.

¹ http://www.ubnt.com/downloads/datasheets/nanostationm/nsm_ds_web.pdf

² <http://www.tp-link.com/lk/products/details/?model=TL-WDR4300>

³ http://dsg.ac.upc.edu/poe_power_meter

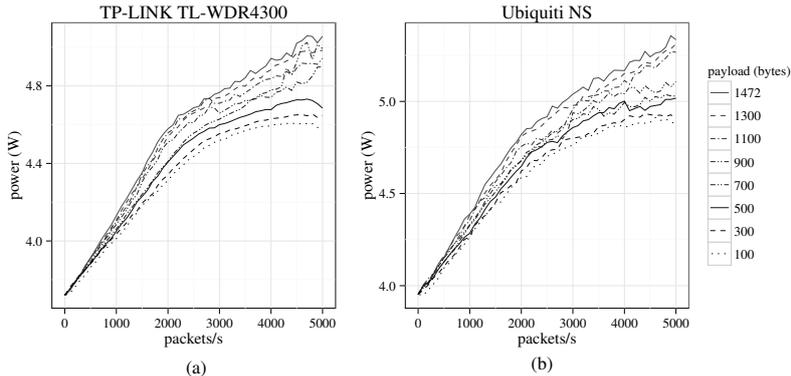


Fig. 6: Power consumption of a typical indoor AP and outdoor router with UDP traffic.

Figure 6 shows the power consumption measured at the (a) TP-LINK and (b) NS devices using UDP traffic vs. packets/s, varying the UDP packet payload. Each point in the figure was obtained averaging around 5×10^4 power samples captured during 1 minute, as described above. Figure 6 shows a static power of 3.7 and 3.9 W, and maximum power of 5 and 5.4 W, for the TP-LINK and NS, respectively. A TCP experiment was carried performed, as well. We obtained a throughput of 65 Mbps and power consumption of 5.2 and 6.1 W at the TP-LINK and NS, respectively.

Additionally, experimental measurements show an average throughput between nodes and their gateway of 10.9 Mbps (see [65]). This can also be estimated as the average throughput between any pair of nodes. Of course, this result is lower than the throughput obtained in the laboratory (65 Mbps) due to obstacle interferences and other transmission impairments which reduce the throughput in the production network.

In summation, we have observed a similar power consumption in the TP-LINK and NS, which falls in the range of approximately 4 W (static power) to 6 W (maximum throughput). However, all devices are far from transmitting at the maximum throughput during a typical transmission. Thus, we conclude that 5 W is a good rule of thumb as power consumption for all networking devices in the QMPSU. Substituting in (10), we have that the average consumption in QMPSU is given by $P_{WN} = 33.9$ W.

Regarding the round trip time (RTT), as results of experimental measurements in QMPSU, an average RTT of each node to the gateway is 18.3 ms, with standard deviation of $\sigma = 50.6$ ms. We shall use these values to estimate end-to-end RTT delays in the network.

4.5 MapReduce workload

To scrutinize the energy consumed in the clouds, we analyze the energy consumed per MapReduce job, both in the data center and P2P-clouds. When a MapReduce request is sent to a data center, the scheduler decides which hosts should perform the job. Being assigned to hosts, the input is split into n_t inputs of $Size_t$ in the map phase. Each individual task with specified input is allocated to a host in the data center; note that more than one task may be assigned to a single host. To complete a task, a host acquires not only the task input data, but also the appropriate VM containing the execution code. Therefore, the data transmitted within the data center communication infrastructure includes VM and input data. For input data size, $Size_{input}$, we assume several cases from the range of 1GB to 50GB. In the second phase of a MapReduce job (i.e. the reduce phase), output is aggregated in the output file of $Size_{output}$ and delivered as the job result. Moreover, for the output of the first phase, the named intermediate output may be exchanged among hosts due to the shuffle-exchange phase. Overall, the size of the transmitted data in this phase is $Size_{intermediate_output}$. Therefore, the size of data to be transmitted is determined by (11).

$$Size_{data}^{MR} = Size_{input} + n_{hosts} \times Size_{VM} + \sum_{i=1}^{n_t} Size_t(i) + Size_{intermediate_output} + Size_{output} \quad (11)$$

$Size_{VM}$ and n_{host} denote the VM size and the number of hosts assigned to the job respectively. The output data size and intermediate output size may vary according to the MapReduce application type and the input file. A diverse range of output and intermediate output sizes is elaborated across our

evaluation scenarios; however, in typical cases, we consider the intermediate output and output size to be 30% and 10% of input data size.

The energy consumed to transmit the required data for a job, as shown in (12) is the multiplication of power drawn for the communication by the amount of data that should be transmitted, as depicted in (11), over the network throughput, τ_{DC} .

$$E_{intra_DC_comm}^{MR} = P_{DC_comm}^{intra_DC} \times \frac{Size_{data}^{MR}}{\tau_{DC}} \quad (12)$$

Exploiting Gigabit Ethernet, the data center network performance is more than 90%; therefore, τ_{DC} is above 967 Mbps.

The energy drained within each host is $\sum_{n_t} P_{host} \times t_{task}$ for each phase. Here, we characterize the Hadoop implementation of MapReduce with five phases of Map, collect, split, merge, shuffle and Reduce. Map, collect, split, and Reduce are generally accomplished in hosts, while merge and shuffle are network and storage hungry phases. t_{task} is the time to process the assigned task in the host, which is directly proportional to the CPU clock frequency. Considering lognormal distribution for the task time [68], the host energy is approximated as $E[t_{task}] \times \sum_{n_t} P_{host}$, where $E[t_{task}]$ represents the expected value of lognormal distribution. The last element of the energy consumed per job is the transmission over the Internet, as illustrated in (8). The only data to be exchanged over Internet in this case is the input and output data. The overall energy consumption for the MapReduce over a data center is (13). PUE is defined for each data center in Table 1.

$$E_{intra_DC}^{MR} = PUE \times [E_{intra_DC_comm}^{MR} + \sum_{i=1}^{numberofphases} (E[t_{task}] \times \sum_{n_t} P_{host})] \quad (13)$$

To analyze the energy consumed in the P2P-cloud per MapReduce job, we should consider two different scenarios. In the first case, jobs are assigned to the hosts within a vicinity (i.e. intra-vicinity); in the second case, inter-vicinity responses are managed. In the case of inter-vicinity responses, a job may be assigned to hosts in another vicinity. The input data, intermediate output, and VM should be sent to the distant host through Internet. Alternatively, in the case of intra-vicinity responses, VM, input, and intermediate output data need only to be sent to a host via a wireless network. To exemplify, considering IEEE 802.11n wireless infrastructure and IPv4 packets, the transmission rate, τ_{intra_P2P} , is 10.9 Mbps as explained before. In this case, the amount of data to transmit over the community network follows (11). Overall, the energy required to accomplish a MapReduce job on community for the intra-vicinity mode is given in (14). t_{P2P} implies the response time of the hosts in P2P-cloud.

$$E_{intra_P2P}^{MR} = P_{WN}^{comm} \frac{Size_{data}^{MR}}{\tau_{intra_P2P}} + \sum_{i=1}^{numberofphases} (E[t_{task}] \times \sum_{n_t} P_{host}) \quad (14)$$

Note that in P_{host} , static power is divided by the number of VMs collocated in the host, while the dynamic power is the amount that is dissipated due to the utilization of resources induced by the task. For the performance of MapReduce jobs, we consider the deadline of the jobs as the ideal performance, and the actual time that the jobs are finished as the measured performance.

4.6 Video Streaming Workload

Video streaming service can provide either online streaming (i.e. content being encoded on the fly) or offline streaming by serving the content previously tailored. Thus, if we consider offline video streaming, no video rendering and encoding in the cloud side is required. Video frames are stored in the cloud storage and retrieved on demand. In this case, communication is the key element in distinguishing energy consumption of different data center models, because the data should be retrieved from network accessible storage. Video decoding, in contrast, is always accomplished on the end-user side. Hence, applying power aware video decoding mechanisms [69] contributes to a more energy efficient service provisioning at the end-user level.

Nonetheless, in cloud-assisted live video streaming (e.g. Amazon CloudFront live video streaming [70]), video rendering is conducted in the data center servers. For instance, CloudFront uses the Adobe Flash Media live encoder.

Table 3: HDD Specifications

Scenario	Type	Throughput random	Throughput sequential	P_{idle}	$P_{sequentialread}$	$P_{randomread}$
Data center	Western Digital RE NAS 4TB	6 Gbps	182 Mbps	8.9 w	10.2 w	10.9 w
P2P-cloud	WD Scorpi Blue (WD5000BPVT) 500MB	3 Gbps	136 Mbps	0.65 w	1.6 w	-

For offline video streaming, as in (15), the energy is dissipated in three parts: retrieving the frames from the NAS, $E_{storage} + E_{transmit}^{intra-DC}$, and transmitting them over the Internet, $E_{transmit}^{Internet}$. On the user side, these frames should be buffered and decoded to play on the screen. Because in all scenarios we target the same end user, we presume that the end-user energy consumption is a constant amount for a given scenario, E_u . In our experiment we assume the chunk of one second; therefore, E_u per chunk is equivalent to the power drawn in the user machine. For instance, playing a YouTube video on our Dell machines induced 25% utilization on one core, which dissipates around 43 W.

$$E_{streaming}^{OD} = E_{storage} + E_{transmit}^{intra-DC} + E_{transmit}^{Internet} + E_u \quad (15)$$

In (15), $E_{storage}$ depends on the drive technology. For the SSD drives this value is only proportional to the data size and is trivial compared to the other parts energy components, while for the HDD drives depends on rapidly rotating technology, this energy not only depends on the data size but also data and disk head location. In HDD, power drawn for retrieving the data is not negligible. Some effort has been made to model disk power usage on HDD technology [8] as $E_{storage} = \frac{P_{HDD}}{\tau_{HDD}}$, where P_{HDD} is the indicator of power drawn to read and write to the disk, and τ_{HDD} specifies the throughput of this process. For the storage in our evaluation, we follow the hardware specified in Table 3.

Apart from the streaming mode (i.e. live or on demand) transmission power over the data center intranet, as introduced on the above section, depends on the data size and the transmission protocol throughput. In this case, we assume that the transmission protocol for intra data center communication, Internet communication and P2P transmission is the same as for the MapReduce case, as discussed in the previous section (i.e. we assume streaming over HTTP, as provided in Amazon Cloud Front).

For live streaming, however, we must to model the video encoding energy consumption $E_{encoding}$ and replace it with the $E_{storage}$ in (15). For the rest of the processes, we can follow the offline streaming model. Video coding tightly couples with the video format. Here, we consider the MPEG4 video format. MPEG4 video is formed as a set of consecutive frames of three different types: I,P, and B frames. I frames are independent images, while P frames are generated based on their previous I frames, and B frames are coded based on the frames before and after them. Typically, the I frame coding follows the JPEG coding. B frame coding draws more power compared to the I and P frames on the same machine, because B frame relies on bi-directional differential coding of the values through the JPEG coding process, $E_{encoding}^I \leq E_{encoding}^P \leq E_{encoding}^B$.

For our evaluation purpose, we model the video streaming in both live and on-demand cases, for MPEG4 video format. Here we assume a chunk of one second, following the typical IBBPBBPBBPB-BIBBPBBPBBPBBI pattern of 24 frames per second. Although the frame sizes vary due to variable-length coding, we assume the average case of 150Kbit, 50 Kbit, and 20Kbit for I, P and B frames, respectively [71]. We also ignore the interlacing and hierarchical coding and simply consider the basic features of MPEG4. For the audio we assumed the average of 30% of the video chunk size.

To be able to compare the energy effectiveness of different platforms for the video streaming process, we compute the measured performance through Algorithm 1. This algorithm, basically counts the number of frames that reached the destination before their expiry time and is used to code the video in its standard format [72].

5 Results and Discussion

In this section, we study different cloud scenarios, throughout the evaluation framework introduced in the previous section, to highlight the pros and cons of different systems. We conclude with a more energy effective architecture.

Algorithm 1 Measure streaming performance

```
1: function MEASURE PERFORMANCE(NumberOfFrames)
2:   perf  $\leftarrow$  0  $\triangleright$  perf is the performance value
3:   while (There is a frame) do
4:     if  $t_{received}^{frame} + t_{process} \leq t_{play}^{PreviousFrame} + t_{playback}$  then
5:       perf ++  $\triangleright$   $t_{process}$  denotes the average time required to process a frame
6:     end if
7:   end while
8:   return perf
9: end function
```

5.1 Idle case energy consumption

As aforementioned, idle power consumption is one of the obstacles that impedes the attainment of energy proportional systems. Here, we study the idle case power of different data center models with the same processing capability, and the corresponding network infrastructure considering hierarchical network topology. In this situation, to be able to compare different cases, we should follow hierarchical network topology, because the switches that we introduce for our evaluation, can only support 2^{13} hosts in a data center of Fat-tree topology, which is not scalable enough for the mega data center case.

Nonetheless, conforming to the server-centric topology of CamCube, we observe that the network idle power is negligible due to the green server ports that consume almost zero watts in the idle case. Moreover, Camcube constitutes a network of servers via peer-to-peer connection, which is exempted of any network hardware switches.

As illustrated in Figure 7, we see that power is significantly drawn in the hosts of the cloud, except for the P2P-cloud which consumes more power in the communication aspect. As explained in the previous section, the P2P-cloud hosts must to connect to an AP which is connected to a NS, and each of these devices draws 4 W in the idle case. Moreover, this figure highlights the effect of data center granularity on idle power dissipation. The larger the data center is, the more power that is drawn in the idle case. Therefore, moving toward a distributed data center model with smaller sizes contributes to reduce idle power and promotes the overall energy proportionality of the cloud.

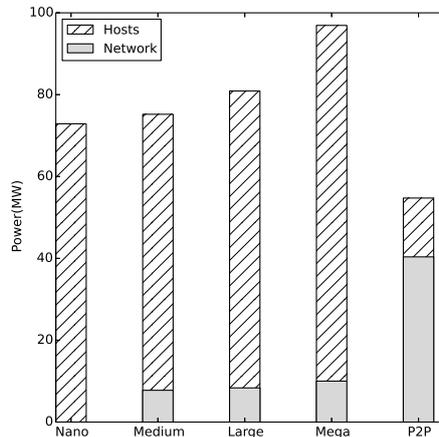


Fig. 7: Idle power consumption across scenarios

5.2 Energy proportionality

As explained earlier, state-of-the-art machines draw power in a non energy proportional pattern, primarily due to the high idle power consumption that dissipates the most power in cloud systems, as shown in Figure 7. Some investigations are focusing on energy-proportional hardware design. Nevertheless, in the architecture level, by combining the heterogeneous processing elements, we can improve the

energy proportionality of the system and even cross the ideal case boundaries in specific situations [73]. Typically in data centers, servers are homogeneous machines to alleviate the burden of management issues.

Here, we elaborate on the scenarios in which the processing capacity of each combination of hosts is equal to a data center server machine. In Figure 8, we see that the combination of P2P-cloud server and host nodes consumes even less power than the energy proportional server. Moreover, this figure implies that combining the data center servers with P2P nodes contributes to improving overall energy proportionality of the hosts, and at low utilization (i.e. less than 20%), we can cross the proportional server in terms of elevating the hosts energy proportionality. In the range of 40% - 60%, we see the improved proportionality of P2P-cloud nodes compared to heterogeneous data center hosts.

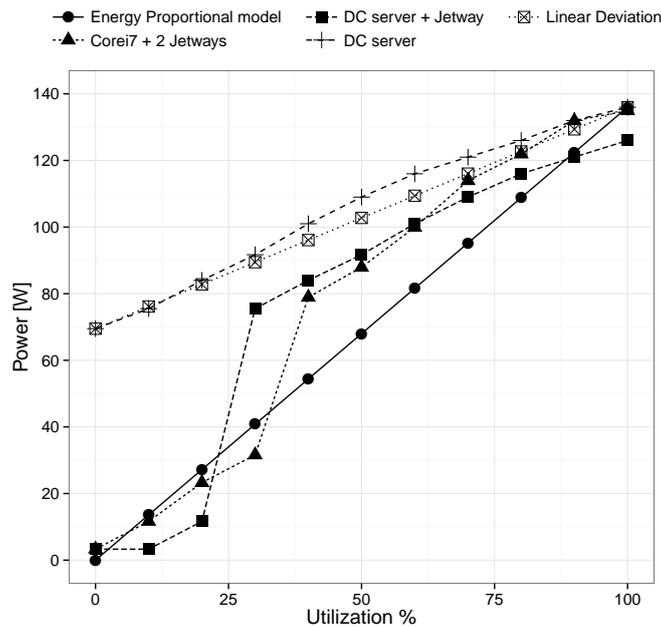


Fig. 8: Heterogeneity effect on energy proportionality

Although currently the scheduling mechanism is first-fit, it can be smarter, if it is equipped with the load level of each node and can decide based on energy proportionality [74].

5.3 Service energy

Figure 9 and 10 show the energy consumption from the service perspective in streaming and MapReduce applications. Results in Figure 9 and 10 indicate that processing that the energy required for processing in cloud systems is much higher than the energy required for transmission energy, except for the offline video streaming in P2P-clouds. Note that in Figure 9, for the sake of clarity, power < is not depicted. As shown in Figure 9, in offline streaming, the P2P-cloud consumes the most energy in the transmission phase, while the energy dissipated for transmission in different data center models is almost the same for both online and offline streaming. However, processing energy is increasing gradually in line with data center granularity for online streaming. The processing draws the same energy in data centers regardless of the data center size, because it includes retrieving and processing the same amount of data from the storage. Note that throughout this section we have assumed that each user and all data center hosts have the same video streaming VM already installed on their machines. This is not an unrealistic assumption as its cost would be quickly amortized after a small number of videos served and played back. Nonetheless, the VM size can affect the energy consumption in the processing.

Figure 10 illustrates that for MapReduce applications, the transmission energy consumed in P2P-cloud is noticeably higher than consumed in the data center model, because nodes are not connected through an energy-efficient network as in the data center. Moreover, we see that the nano data center

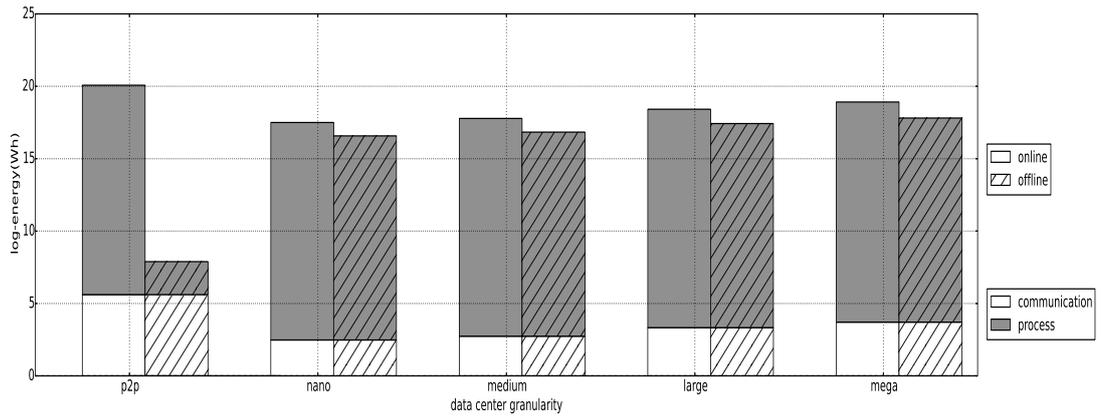


Fig. 9: Streaming of video

model dissipates slightly more energy for transmission due to additional inter data center communication required to interact with different elements of the system in each MapReduce phase. Note that we assume all the MapReduce tasks are accomplished within a single data center in the distributed data center. Nonetheless, transmission energy can be remarkably higher if we have to share the tasks among the data centers. Conversely, the processing energy consumed in the P2P-cloud is slightly lower than in the data center model due to more energy proportional hosts and no cooling mechanism. Data center processing energy is increasing in line with the granularity as a result of greater cooling energy required.

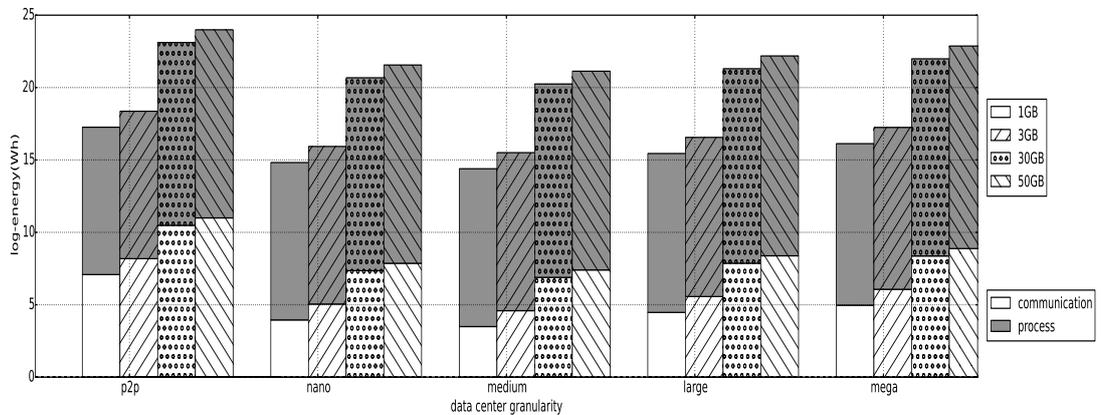


Fig. 10: MapReduce applications

5.4 Replication

Replication of offline video streaming literally increases the number of tasks and data size times to the replication factor. Therefore, replication increases the probability of locally available content and increases the reliability of P2P-cloud service provisioning. Figure 11 and Figure 12 illustrate the energy consumption in the presence of replication for streaming and MapReduce applications, respectively.

In video streaming scenarios, we consider the replication factor of 3 for the content distribution. Comparing Figure 9 and Figure 11, we find out that offline video streaming over the P2P-cloud is more energy efficient, even with replication in comparison to the data center model, provided that we have the content stored locally in the vicinity.

In the case of MapReduce, common Distributed File Systems (DFS), such as Hadoop HDFS which is prevalently exerted, has a replication factor of 3, to replicate the input and output data. However, in the P2P-cloud we also have to replicate the tasks. Therefore, replication affects the energy consumption in P2P-cloud more than the data center model. In data centers, replication substantially affects the communication directly proportional to the replication factor.

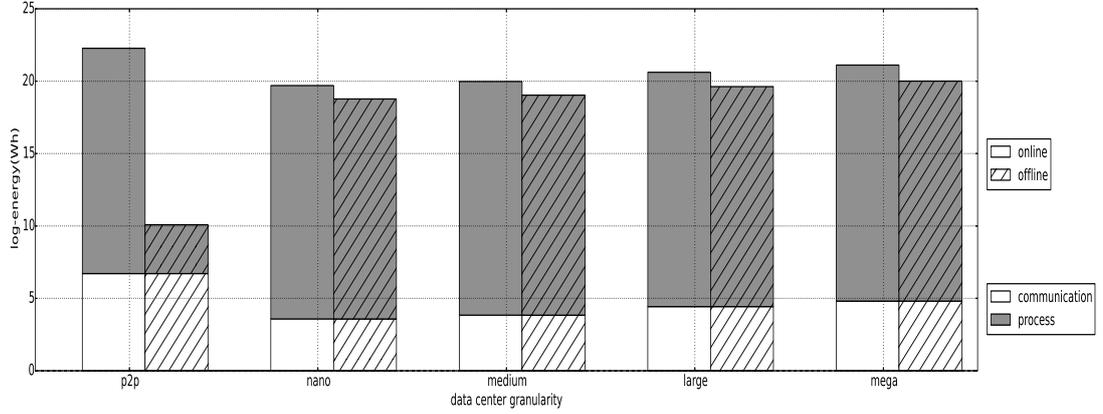


Fig. 11: Video streaming with replication

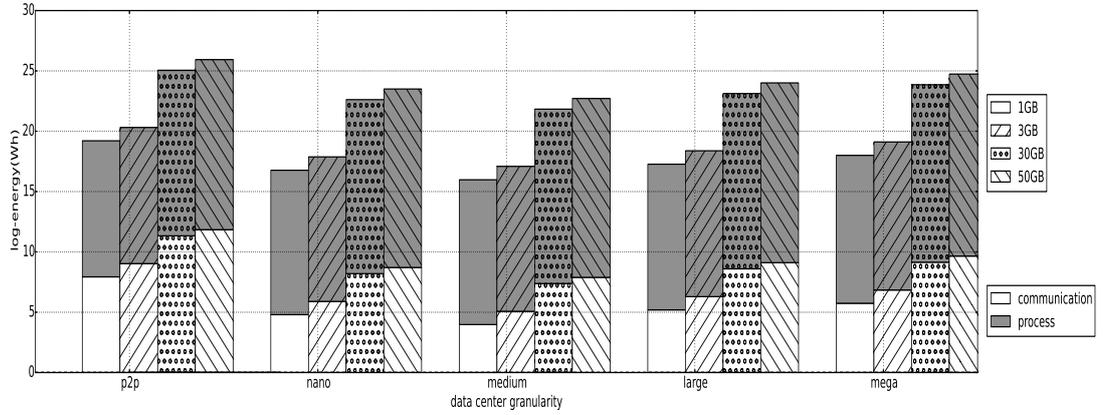


Fig. 12: MapReduce applications with replication

5.5 Energy effectiveness

To compare the energy effectiveness of different data centers, as introduced in Section 3.8, through out the studied services, we assume the energy dissipated to serve the application in an energy-proportional server of the data center as the energy benchmark. For the performance we assume that running the application on the data center can provide the maximum performance. Therefore $\frac{Perf_{measured}}{Perf_{ideal}}$ for data center service provisioning is equal to one, while this value is in the range of zero to one for the P2P-cloud. However, P2P hosts consume less energy to process the applications, which leads to values higher than one for the term $\frac{E_{ideal}}{E_{measured}}$ in (9). As a result, in Figure 13, we can find the energy effectiveness of P2P-cloud to be higher than of the data center by orders of magnitude when α is large enough.

However, as the energy weight decreases, energy effectiveness decreases in the P2P-cloud. This partially indicates the correlation of energy proportionality and energy effectiveness. As discussed in

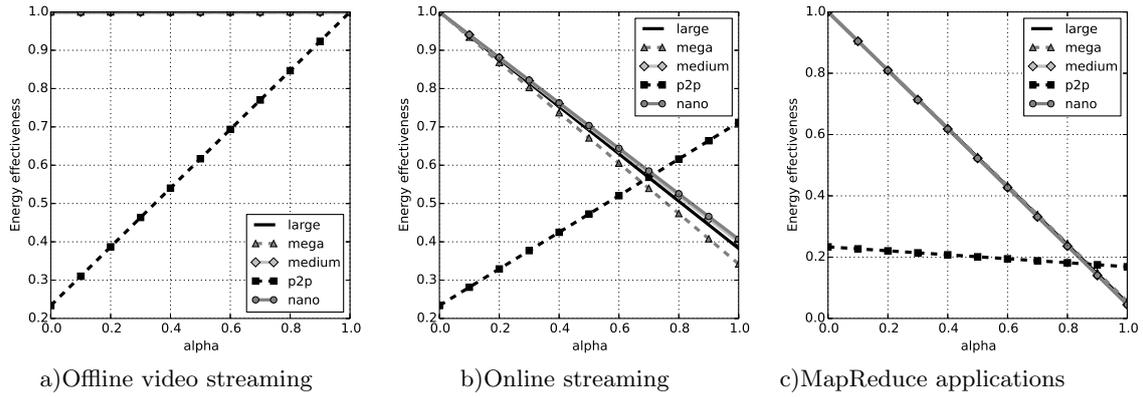


Fig. 13: Energy Effectiveness across scenarios

Section 5.2, the heterogeneous environment of the P2P-cloud delivers a more energy proportional processing ambiance compared to the data center with homogeneous servers. Therefore, the higher the processing power, the higher the chance of promoting the energy effectiveness of the P2P-cloud via increasing the α . Hence, the energy effectiveness of the P2P-cloud offline streaming is always lower than that of the data center model, Figure 13-a, because little processing is conducted in the cloud side to retrieve the video from storage, and the transmission energy is higher in the P2P-cloud model. For MapReduce applications, Figure 13-c, the data center model is more energy effective unless the system is very energy conservative via setting $\alpha \geq 0.9$

Considering data center granularity, Figure 13-b,c highlight that the performance of the online streaming and MapReduce application regardless of the data center granularity, does not change, exerting the same infrastructure hardware, $\alpha = 0.0$, provided that we have sufficient resources in smaller data centers. However, increasing the α value causes the mega data center's energy effectiveness to decrease more, due to higher PUE.

Figure 14, indicates the responsiveness of each platform, in the context of three different application workloads, to the α parameter. From the figure we can see that the P2P-cloud's energy effectiveness varies lesser when compared to other platforms, whereas the medium data centers vary the most based on the α value changes. Medium data center sensitivity always remains the same, because we normalize the other platforms performance and energy values according to medium data center energy and performance values.

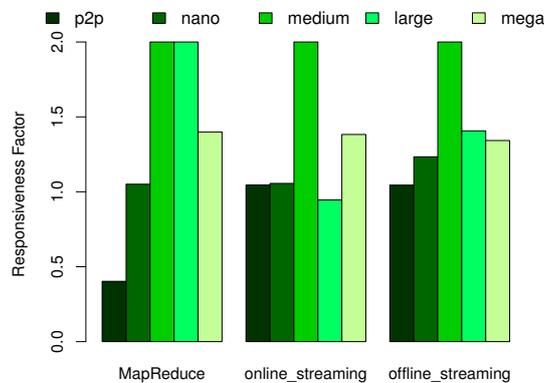


Fig. 14: Sensitivity to α selection across different services and platforms

5.6 Communication pattern

Here, we elaborate the energy consumption to transmit 1GB of data on different topologies in three scenarios: i) intra-cluster scenario which targets the communication among the servers of the same cluster. ii) inter-cluster communication that focuses on the transmission of data among the servers of two different clusters. iii) inter-data centers that looks at the communication among different data centers of a distributed cloud.

Figure 15 reveals that the CamCube server-centric topology consumes the most and same amount of energy in intra-cluster and inter-cluster scenarios. Among switch-centric scenarios, Fat-tree dissipates the least energy in the intra-cluster scenario. Flattened butterfly is the most energy efficient for inter-cluster communication, but it is not easily scalable for the case of the mega data center. For the inter-dc scenario, the difference among the topologies decreases, because the inter-dc connection energy consumption dominates the other parts energy. As a result, a hybrid topology that combines the best of all components of the data center network can be a solution for a more energy efficient topology.

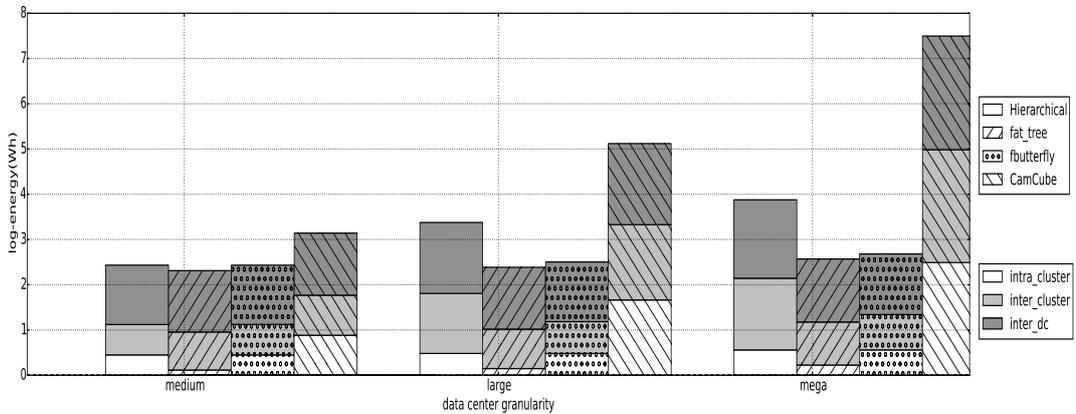


Fig. 15: Topology effect

5.7 VM migration

Because processing is more a function of VM activity log, and can be considered the same for the identical application with the same input, we only consider the communication energy for VM migration. Across Figure 16, Figure 17 and Figure 18, the energy consumption due to VM migration is elaborated for different VMs mentioned in Table 2. As we see in the Figure 16, Figure 17, and Figure 18, migration energy dissipation conforms to the communication pattern energy consumption directly proportional to the data center granularity and networking, as studied in Section 5.6. All in all, the VM migration within a cluster is the most efficient approach in terms of energy efficiency.

5.8 Put It All Together

Overall, referring to the results provided in this section, we can see that for the services requiring distributed data processing and, hence, communication among the processing elements, data center model energy effectiveness outperforms P2P-cloud, while for the content delivery based services (e.g. offline video streaming) P2P-cloud is the predominant model in terms of energy saving. However, considering the performance by defining energy effectiveness, the table may turn in favor of the data center model due to limited resources and processing capabilities of the P2P-cloud. Therefore, a data center assisted P2P-cloud model can fulfill the energy effectiveness ambitions. There are efforts on providing streaming service based on such an architecture, namely CLive [75] which adaptively leverages cloud based resources for P2P streaming of video. Nonetheless, P2P-cloud architecture can affect the energy efficiency. Thus, forming an energy efficient architecture can improve the service energy effectiveness as well. P2P-cloud energy efficiency is studied in [76].

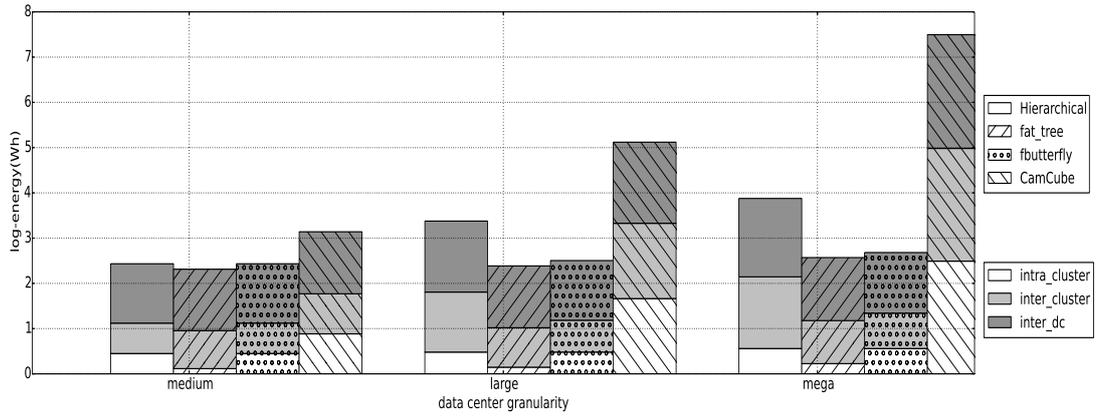


Fig. 16: VM migration across scenarios for small VM

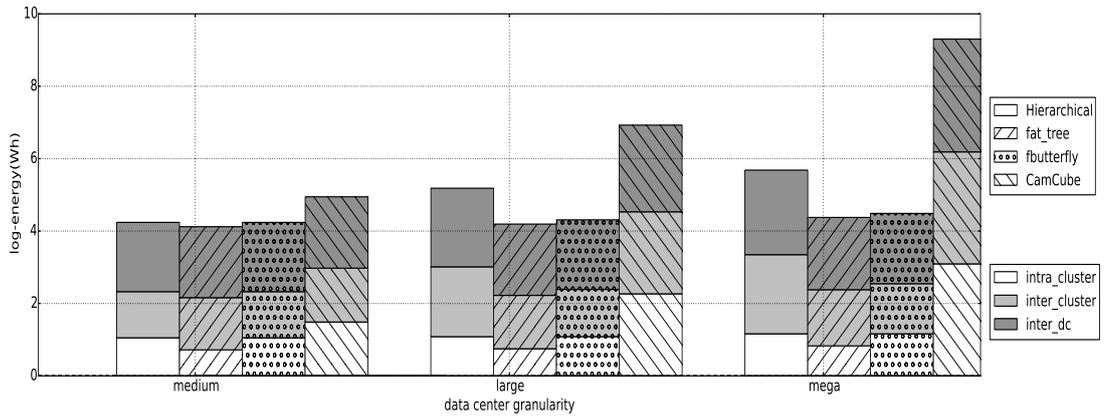


Fig. 17: VM migration across scenarios for medium VM

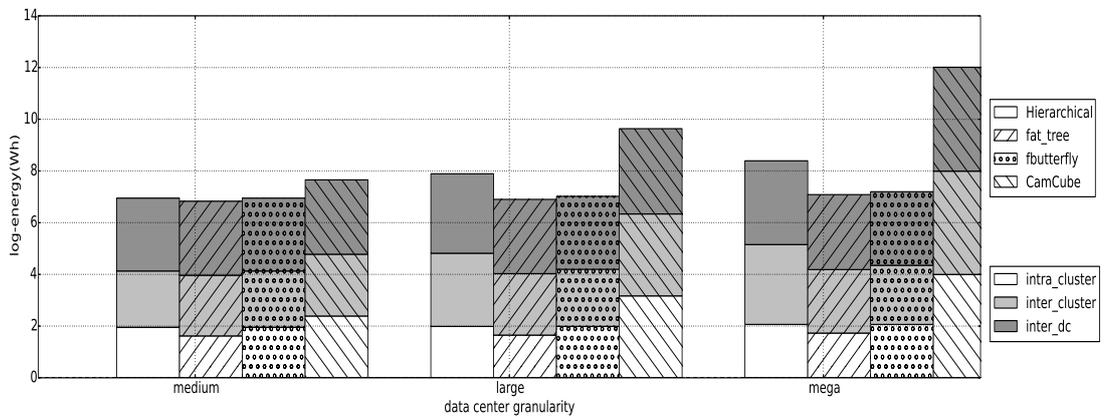


Fig. 18: VM migration across scenarios for large VM

Loosely paraphrasing, a bi-level distributed cloud architecture which combines the concept of P2P-cloud at the edge of the system assisted by local data centers at higher level is an architecture-level solution to improve the energy effectiveness of cloud services. Bi-level architecture is the most energy effective method. It warrants the complexity of interoperability, mobile agents, security, resource discovery, and management.

Increasingly, at the data center level, granularity affects the energy consumption. Smaller data centers are more flexible and induce less PUE, considering the same technology. Moreover, they are exempted from dealing with scalability issues in network topology as in mega data centers. They can easily adapt any network topology to improve the performance and energy consumption. Furthermore, distributed data center architecture brings high availability which contributes to local service provisioning and provides the further opportunity to access local renewable energy sources.

6 Conclusion

In this paper we studied the energy consumption in different data center architectures and compared them with P2P-clouds. Our analysis revealed that the P2P-cloud consumes more energy in networking, while the data center dissipates more energy for processing. Therefore, the P2P-cloud is more suitable for local content delivery services such as offline video streaming, while the data center model is appropriate for the applications acquiring intermediate interaction among distributed resources during the processing (e.g. MapReduce).

Moreover, we realized that centering on energy consumption, the performance may be affected dramatically. To tackle this issue, we introduced the energy effectiveness metric as a combination of energy and performance parameters. Examining the energy effectiveness of different scenarios, we came up with the two-level distributed cloud architecture as the most energy effective solution. It combines the P2P-cloud at the edge of the system, inspired by edge computing, with distributed local data centers in the backbone.

Furthermore, our study on the data center architecture and topology effects on energy consumption contributed to introducing a hybrid intra-data center topology that combines the most energy efficient state-of-the-art topologies in two levels of intra-cluster and inter-cluster topology formation.

Acknowledgment. This work was supported by the European Framework Programme 7 FIRE Initiative projects CONFINE (FP7-288535) and CLOMMUNITY (FP7-317879), by the Universitat Politècnica de Catalunya BarcelonaTech and by the Spanish government under contract TIN2013-47245-C2-1-R. This work was supported by national funds through Fundação para a Ciência e a Tecnologia with reference UID/CEC/50021/2013.

References

1. A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Euro-Par 2013 Parallel Processing*. Springer, 2013, pp. 317–328.
2. A. Kertesz, J. Dombi, and A. Benyi, "A pliant-based virtual machine scheduling solution to improve the energy efficiency of iaas clouds," *Journal of Grid Computing*, pp. 1–13, 2015.
3. O. A. de Carvalho Junior, S. M. Bruschi, R. H. C. Santana, and M. J. Santana, "Green cloud meta-scheduling," *Journal of Grid Computing*, pp. 1–18.
4. Z. Tang, L. Qi, Z. Cheng, K. Li, S. U. Khan, and K. Li, "An energy-efficient task scheduling algorithm in dvfs-enabled cloud environment," *Journal of Grid Computing*, pp. 1–20, 2015.
5. J. A. Pascual, T. Llorido-Bostrán, J. Miguel-Alonso, and J. A. Lozano, "Towards a greener cloud infrastructure management using optimized placement policies," *Journal of Grid Computing*, pp. 1–15, 2014.
6. V. Ebrahimirad, M. Goudarzi, and A. Rajabi, "Energy-aware scheduling for precedence-constrained parallel virtual machines in virtualized data centers," *Journal of Grid Computing*, vol. 13, no. 2, pp. 233–253, 2015.
7. A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.
8. J. Baliga, R. W. Ayre, K. Hinton, and R. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2011.
9. K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali *et al.*, "A taxonomy and survey on green data center networks," *Future Generation Computer Systems*, 2013.
10. A. Hammadi and L. Mhamdi, "A survey on architectures and energy efficiency in data center networks," *Computer Communications*, vol. 40, pp. 1–21, 2014.
11. L. Gyarmati and T. A. Trinh, "How can architecture help to reduce energy consumption in data center networking?" in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*. ACM, 2010, pp. 183–186.
12. J. Doyle, R. Shorten, and D. O'Mahony, "Stratus: Load balancing the cloud for carbon emissions control," *IEEE Transaction on Cloud Computing*, 2013.
13. F. F. Moghaddam, M. Cheriet, and K. K. Nguyen, "Low carbon virtual private clouds," in *Cloud Computing (CLOUD)*, 2011, pp. 259–266.

14. J. Subirats and J. Guitart, "Assessing and forecasting energy efficiency on cloud computing platforms," *Future Generation Computer Systems*, 2014.
15. M. Alicherry and T. Lakshman, "Network aware resource allocation in distributed clouds," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 963–971.
16. Akamai cloud computing services and content delivery network, <http://akamai.com>.
17. Nanodatacenters, <http://www.nanodatacenters.eu/>.
18. V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 37–48.
19. Supermicro microcloud solution, <http://www.supermicro.com/>.
20. C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009.
21. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.
22. B. Braem, C. Blondia, C. Barz, H. Rogge, F. Freitag, L. Navarro, J. Bonicioli, S. Papathanasiou, P. Escrich, R. Baig Viñas *et al.*, "A case for research with and on community networks," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 68–73, 2013.
23. T. G. Peter Mell, "The nist definition of cloud computing," *Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930*, 2011.
24. A. M. Khan, L. Sharifi, L. Veiga, and L. Navarro, "Clouds of Small Things: Provisioning Infrastructure-as-a-Service from within Community Networks," in *2nd International Workshop on Community Networks and Bottom-up-Broadband (CN-BuB 2013)*, within *IEEE WiMob*, Lyon, France, Oct. 2013.
25. S. Choi, H. Kim, E. Byun, M. Baik, S. Kim, C. Park, and C. Hwang, "Characterizing and classifying desktop grid," in *Cluster Computing and the Grid, CCGRID 2007*. IEEE, 2007, pp. 743–748.
26. X. Zhiquan, C. Duan, H. Zhiyuan, and S. Quynying, "Emerging of telco cloud," *Communications, China*, vol. 10, no. 6, pp. 79–85, 2013.
27. Y. Liu, L. Xiao, X. Liu, L. M. Ni, and X. Zhang, "Location awareness in unstructured peer-to-peer systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 2, pp. 163–174, 2005.
28. E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1-4, pp. 72–93, 2005.
29. V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002, pp. 300–307.
30. M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1447–1460, 2008.
31. M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 63–74.
32. A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "V12: a scalable and flexible data center network," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 51–62.
33. Revolutionizing network design flattening the data center network with the qfabric architecture, <http://www.itbiz.com.ua/media/docs/juniper/qfx/the%20qfabric%20architecture.pdf>.
34. L. Huang, Q. Jia, X. Wang, S. Yang, and B. Li, "Pcube: Improving power efficiency in data center networks," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 65–72.
35. P. Costa, A. Donnelly, G. O'Shea, and A. Rowstron, "Camcuboes: a key-based network stack for 3d torus cluster topologies," in *Proceedings of the 22nd international symposium on High-performance parallel and distributed computing*. ACM, 2013, pp. 73–84.
36. C. Clos, "A study of non-blocking switching networks," *Bell System Technical Journal*, vol. 32, no. 2, pp. 406–424, 1953.
37. D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3. ACM, 2010, pp. 338–347.
38. D. Lin, Y. Liu, M. Hamdi, and J. Muppala, "Flatnet: Towards a flatter data center network," in *Global Communications Conference (GLOBECOM), 2012 IEEE*. IEEE, 2012, pp. 2499–2504.
39. C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: a scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 75–86, 2008.
40. G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, "c-through: Part-time optics in data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 327–338.
41. N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: a hybrid electrical/optical switch architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 339–350, 2011.
42. C. Belady, A. Rawson, J. Pflueger, and T. Cader, "Green grid data center power efficiency metrics: Pue and dcie," Technical report, Green Grid, Tech. Rep., 2008.
43. R. Bertran, Y. Becerra, D. Carrera, V. Beltran, M. Gonzalez Tallada, X. Martorell, J. Torres, and E. Ayguade, "Accurate energy accounting for shared virtualized environments using pmc-based power modeling techniques," in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*. IEEE, 2010, pp. 1–8.
44. S. Schubert, D. Kostic, W. Zwaenepoel, and K. G. Shin, "Profiling software for energy consumption," in *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*. IEEE, 2012, pp. 515–522.
45. D. Wong and M. Annaram, "Knightshift: Scaling the energy proportionality wall through server-level heterogeneity," in *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*. IEEE, 2012, pp. 119–130.
46. Spec power benchmark, <https://www.spec.org/benchmarks.html>.
47. N. Tolia, Z. Wang, M. Marwah, C. Bash, P. Ranganathan, and X. Zhu, "Delivering energy proportionality with non energy-proportional systems-optimizing the ensemble." *HotPower*, vol. 8, pp. 2–2, 2008.
48. M. X. Makkes, A. Taal, A. Osseyran, and P. Grosso, "A decision framework for placement of applications in clouds that minimizes their carbon footprint," *Journal of Cloud Computing*, vol. 2, no. 1, pp. 1–13, 2013.
49. Google data center cooling, <http://www.google.com/about/datacenters/efficiency/internal/water-and-cooling>.
50. N. Vasic, T. Scherer, and W. Schott, "Thermal-aware workload scheduling for energy efficient data centers," in *Proceedings of the 7th international conference on Autonomic computing*. ACM, 2010, pp. 169–174.
51. "Open, Free and Neutral Network Internet for everybody," <http://guifi.net/en>.
52. "Athens Wireless Metropolitan Network," <http://www.awmn.net>.
53. "FunkFeuer free net," <http://www.funkfeuer.at>.
54. "Freifunk," <http://freifunk.net>.
55. A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi, "Energy consumption anatomy of 802.11 devices and its implication on modeling and design," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 169–180.
56. J. Baliga, K. Hinton, and R. S. Tucker, "Energy consumption of the internet," in *Joint International Conference on Optical Internet, and the 32nd Australian Conference on Optical Fibre Technology. COIN-ACOFT 2007*. IEEE, 2007, pp. 1–3.
57. S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

58. P. da Silva Dias Rodrigues, C. N. da Cruz Ribeiro, and L. Veiga, "Incentive mechanisms in peer-to-Peer networks," in *15th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems (DPDNS), 24th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2010)*. IEEE Press, Apr. 2010.
59. J. Simão and L. Veiga, "Flexible SLAs in the cloud with partial utility-driven scheduling," in *IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom 2013)*. IEEE, Dec. 2013.
60. —, "VM economics for java cloud computing - an adaptive and resource-aware java runtime with quality-of-Execution," in *The 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012) Doctoral Symposium: Cloud Scheduling, Clusters and Data Centers*. IEEE, May 2012.
61. S. Roy, A. Rudra, and A. Verma, "An energy complexity model for algorithms," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. ACM, 2013, pp. 283–304.
62. Stress tool, <http://linux.die.net/man/1/stress>.
63. "Quick Mesh Project," <http://qmp.cat>.
64. "Community Networks Testbed for the Future Internet, CONFINE," <http://confine-project.eu/>, FP7 European Project 288535.
65. L. Cerd-Alabern, A. Neumann, and P. Eschrich, "Experimental evaluation of a wireless community mesh network," in *The 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'13*. Barcelona, Spain: ACM, Nov. 3–8, 2013.
66. "qMp Sants-UPC monitoring web page," <http://dsg.ac.upc.edu/qmpsu>.
67. "OpenWrt Linux distribution for embedded devices," <https://openwrt.org>.
68. H. Yang, Z. Luan, W. Li, and D. Qian, "Mapreduce workload modeling with statistical approach," *Journal of grid computing*, vol. 10, no. 2, pp. 279–310, 2012.
69. J. Pouwelse, K. Langendoen, R. Lagendijk, and H. Sips, "Power-aware video decoding," in *22nd Picture Coding Symposium, Seoul, Korea, 2001*, pp. 303–306.
70. Amazon cloud front live streaming service, <http://docs.aws.amazon.com/amazoncloudfront/>.
71. Mpeg2, <http://www.h264encoder.com/>.
72. R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of http video streaming," in *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. IEEE, 2011, pp. 485–492.
73. B.-G. Chun, G. Iannaccone, G. Iannaccone, R. Katz, G. Lee, and L. Niccolini, "An energy case for hybrid datacenters," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 1, pp. 76–80, 2010.
74. L. A. Barroso and U. Hözlze, "The case for energy-proportional computing," *IEEE computer*, vol. 40, no. 12, pp. 33–37, 2007.
75. A. H. Payberah, H. Kavalionak, V. Kumaresan, A. Montresor, and S. Haridi, "Clive: Cloud-assisted p2p live streaming," in *Peer-to-Peer Computing (P2P), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 79–90.
76. L. Sharifi, N. Rameshan, F. Freitag, and L. Veiga, "Energy efficiency dilemma: P2p-cloud vs. datacenter," 2014.

7 Appendix: Abbreviation List

Abbreviation	Term
AP	Access Point
DC	Data Center
EE	Energy Effectiveness
EP	Energy Proportionality
IPR	Idle to Peak Power Ratio
IoT	Internet of Things
LDR	Linear Deviation Ratio
NS	Nano Station
RF	Responsiveness Factor
PUE	Power Usage Efficiency
QMPSU	Quick Mesh Project Sants UPC