Energy Efficiency Dilemma: P2P-cloud vs. datacenter

Leila Sharifi*,†, Navaneeth Rameshan†, Felix Freitag†, Luís Veiga*
*Técnico Lisboa/INESC-ID Lisboa, Lisboa, Portugal
Email: leila.sharifi@tecnico.ulisboa.pt, luis.veiga@inesc-id.pt
†Computer Architecture Dept., Universitat Politecnica de Catalunya, Barcelona, Spain
Email: {lsharifi, rameshan, felix}@ac.upc.edu

***Abstract*—Energy consumption is increasing in the IT sector and a remarkable part of this energy is consumed in datacenters. Numerous techniques have been proposed to solve the energy efficiency issue in cloud systems. Recently, there are some efforts to decentralize the cloud via distributing datacenters in diverse geographical positions. In this paper, we elaborate on the energy consumption of different cloud architectures, from a mega-datacenter to a P2P-cloud that provides extreme decentralization in terms of datacenter size. P2P-cloud is defined as a set of commodity host machines, connected to each other to serve a community. Our evaluation results reveal the fact that the more decentralized the system is, the less energy may be consumed in the system. Studying the energy efficiency of P2P-cloud infrastructure shows that the additional system design complexity involved is warranted with improved energy-efficiency and better locality for some services. Our analysis indicates that such P2P-cloud outperforms the classic datacenter model as long as it meets the locality conditions, which are commonplace in communities. Moreover, we illustrate how much energy can be saved for MapReduce applications with a diverse range of specifications by switching to P2P-cloud.**

I. Introduction

Power saving in IT industry is becoming a critical issue. Datacenters consume vast energy not only to keep the devices running, but also for cooling the datacenter. Indeed, it has become vital to design hardware, architecture, application and resource management algorithms, protocols and policies that are energy aware.

Energy efficiency in clouds is a threefold challenge: i) exploiting energy efficient hardware, ii) employing the hardware in an energy efficient way, and iii) applying energy efficient virtualization techniques. The last two elements are related to Peer-to-Peer cloud (P2P-cloud) architecture as well, since virtualization is the key concept in such systems. A diverse range of techniques [1] have been introduced to accomplish energy aware assignment in cloud systems, some of them are directly applicable to P2P-cloud. However, P2P-cloud requires additional consideration to thoroughly deal with the energy efficiency.

In this paper, we speculate the energy consumption of two different cloud architectures, a mega-datacenter and a P2P-cloud which fulfils the ultimate decentralized architecture goal. In the vision of a P2P-cloud deployment, a cloud hosted on peer-to-peer computing and communication resources provisions the services. Reduced latency and increased data transmission rates are achievable by assigning requests to servers which are closer in terms of link latency. Additionally, we study the P2P-cloud energy consumption and propose the location based P2P-cloud model for the sake of energy efficiency. The motivation behind studying P2P-cloud energy efficiency is the move toward decentralized datacenter model for the sake of energy efficiency [2]. There is some work on comparing the energy efficiency of datacenters and private clouds [3]; however, energy efficiency of P2P-cloud as an extremely decentralized cloud architecture has not been studied so far.

Comparing the energy consumption of P2P-cloud and datacenters, we find out that the energy consumed for cooling in the datacenters 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 [4] reveals that by assigning the requests to the local datacenters, not only communication cost is not increased, but also it is reduced in most cases, thanks to more energy efficient paths between consumer and server. Thus, nowadays, cloud providers are switching to a more distributed cloud architecture for the purpose of energy efficiency [2], [4], [5]. Our proposed cloud model, i.e. a P2P-cloud, fulfils the ultimate decentralized architecture, but is different than classic cloud models due to relying on commodity devices as its hardware platform.

To summarize, the specific contributions of this work are:

- An analytical study of the energy consumed in a datacenter and P2P-cloud, Section III.
- Comparison of energy efficiency for different MapReduce applications in a datacenter and P2P-cloud, Section IV.
- Propose a cache model to improve energy efficiency of P2P-cloud, Section V.

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

II. System Models

In this section, we outline the elaborated cloud models as well as terms and conditions used for the models.

A. Classic Datacenter

In the classic datacenter model from which the idea of cloud computing stems from, a gigantic datacenter consists of numerous clusters of hosts that constitute a powerful computing and/or storage capacity. Each host on the cloud may be further divided into multiple virtual machines to provide application isolation and improve resource efficiency. In this section, we draw the differences between a distributed datacenter and P2P-cloud.

1) Distributed Datacenter: The challenges of supporting business continuity in a cloud environment are not limited to a physical area or the datacenter alone, the elasticity and flexibility of the network architecture is a key requirement. Therefore, the compute, storage, and network components used for cloud computing may not 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. Distributed cloud datacenters, alongside with bringing high availability and disaster recovery, provide the opportunity to have different energy sources.

B. P2P-cloud

A P2P-cloud constitutes of a number of vicinities, each comprises a set of commodity hosts, including Internet of Things boards, laptops and PCs, connected via a wireless communication platform as depicted in Figure 1. The main goal of P2P-cloud is to take advantage of distributed datacenter hosts as well as exploiting the commodity hardware of community networks.

Community networks represent collaborative effort of community members, for building ICT infrastructure with commodity devices in a bottom-up approach, in order to meet demand for Internet access and services [6]. The P2P-cloud 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.

Comparing P2P-cloud with desktop grid [7], we find out that desktop grid is a peer to peer volunteer computing platform. However, P2P-cloud services are not confined to the 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 above mentioned concepts. To exemplify, P2P-cloud hosts may be mobile or static. P2P-cloud reinforces the concept of telco-cloud [8], since communication and IT infrastructures akin to the community network is required to develop a P2P-cloud.

In 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, since 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 slighter energy compare to datacenter servers.

Increasingly, in P2P-cloud, to alleviate the energy consumption, requests can be assigned to one of the closest available hosts in the community. The closer the client and the server are, the less energy is consumed in the network. Based on this observation, we define the P2P-cloud topology as a set of community hosts scattered within dynamic vicinities and communicating via wireless communication network (intra-vicinity communication) as depicted in Figure 1. Each vicinity can access the others via Internet; this is known as inter-vicinity communication.

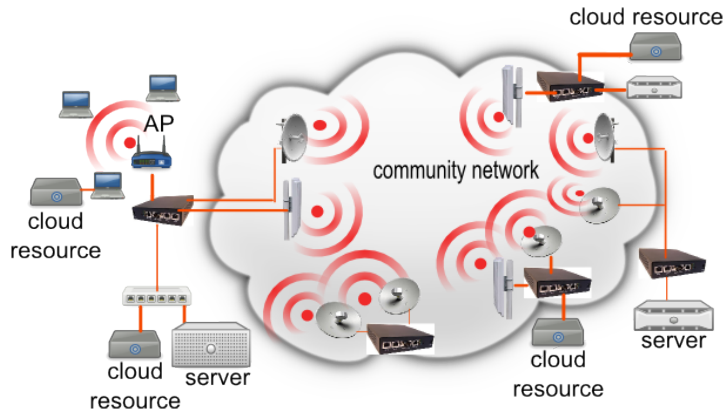


Fig. 1: P2P-cloud intra-vicinity model

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 [9]–[11].

Studies reveal that virtually all the requests a user issue for the service, in a specific location, are akin to the others due to the locality of requests [12]. The P2P-cloud can adapt to and leverage this fact by adjusting the service and computing capabilities of each individual community nodes accordingly; whereas, responding to high resource demanding requests via the federation of more powerful machines like core i7 PCs, or forwarding them to the classic cloud.

III. Energy Consumption Analysis

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 datacenters this list extends to the cooling, lighting and maintenance energy as defined in the datacenter Power Usage Efficiency (PUE) [13]. To obviate the power efficiency of a datacenter, PUE parameter is defined as the ratio of total amount of power used by a computer datacenter facility to the power delivered to the computing equipment. The higher the PUE, the less power efficient the datacenter is. In the rest of this section, we analyse the energy consumption of each element.

A. 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 $P_d = (P_{Max} - P_s) \times U$. U is the utilization level of the host, and P_{Max} indicates nominal power as the maximum power device can dissipate.

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

In (1) we assume a linear correlation among the utilization level and the power drain in the host, which is known as hypothetical linear power model. This model reveals the ideal power model with Linear Deviation Ratio (LDR) of one. However, in real systems, the LDR is not equal to one. LDR [14] is a metric defined as the maximum difference of the actual power consumption and hypothetical linear power model over the hypothetical linear power model as in (2).

$$LDR = \frac{P(U) - [P_s + (P_{Max} - P_s) \times U]}{P_s + (P_{Max} - P_s) \times U} \tag{2}$$

P_s for common servers in a datacenter is above 100 watts. In case of community networks, hosts are commodity machines. The energy consumption is trivial comparing with the datacenter hosts. By sharing part of their computing resources, the users are contributing in the community. Indeed, idle energy consumed in community networks are much lesser than the datacenter 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 datacenter.

B. Datacenter power consumption

In classic datacenters power P_{DC} is consumed not only in hosts, but also for intra-datacenter communication, P_{DC_comm}, cooling, $P_{cooling}$, lighting, etc which are affecting the PUE parameter within a datacenter, as in (3).

$$P_{DC} = \sum_{numberofhosts} P_{host} + P_{DC\_comm} + P_{cooling} \tag{3}$$

For cooling, embedding simple fans or chiller systems in datacenter aisles are the most common techniques nowadays. 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. Doyle, et al. [4] analyzed the power drawn for different cooling techniques. The energy overhead of all the non-IT devices such as cooling, lighting, etc. is typically modeled as the $\frac{(PUE-1)}{U}$ coefficient of the overall power of the resource [3], [15], where U represents the utilization of the resource.

In communication within a datacenter, switches and communication links that connect the hosts are the major power consumption sources. Moreover, the network topology impacts the power usage profile. Here we study the power consumption of a hierarchical topology, which is easily scalable. We assume a 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.

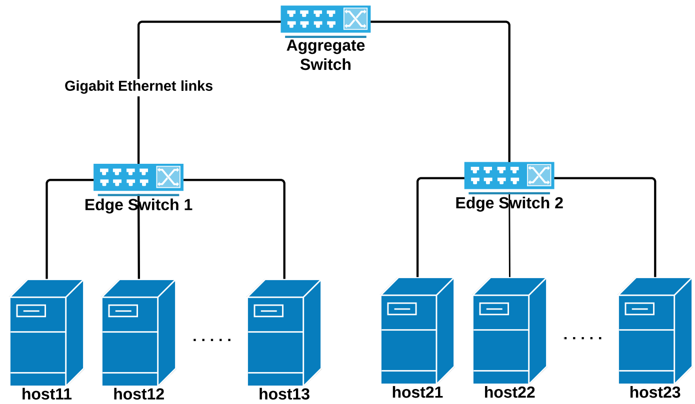


Fig. 2: Intra-datacenter communication model

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 fat-tree, the power consumed for this purpose is calculated as in (4). P_{switch} and P_{link} stand for power drawn by the switch and the link respectively.

$$P_{DC\_comm} = \sum_{i=1}^{l-1} (P_{switch}(i) + P_{link(i)}) \tag{4}$$

Referring to (4), the depth of the tree, l, directly influences the power efficiency of the datacenter. The tree depth is determined by the number of hosts. The larger the datacenter is, the more the number of switches and links required to connect the hosts and the deeper the tree is. Therefore, smaller distributed datacenters, serving the users independently, are more power efficient than a single mega-datacenter model, following a complete tree intra-datacenter topology. Loosely paraphrasing, in small datacenters, the tree depth is smaller, since the number of switches and links required to connect the hosts within a datacenter is directly related to the number of hosts. Therefore, the path should be traversed to reach a host contains less hops comparing to a deeper tree with more hosts.

C. P2P-cloud Power consumption in communication infrastructure

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(WN) power consumed to transmit data [16].

To transmit data from a source to the destination, different wireless routing protocols such as Dynamic Source Routing (DSR) or Optimized Link State Routing (OLSR) may be applied. Each protocol may impose distinct amount of overhead and produce different number of packets. Analysing each protocol separately is tricky. Here, for brevity, we examine the worst case which imposes the highest overhead and produces the largest amount of packets, i.e. flooding in the network via broadcasting. The power consumed for communication, P_{WN}^{comm}, is as in (5). Where, $N_{neighbours}^{hopi}$ stands for the number of i hop-far neighbours of a node, and $P_{WN}^{broadcast}$ represents broadcast power drawn in wireless.

$$P_{WN}^{comm} = \sum_{i=1}^{num_{hops}} N_{neighbours}^{hop_i} \times P_{WN}^{broadcast} + \sum_{i=1}^{N-1} P_{WN}^{receive}(i) \tag{5}$$

This equation addresses the relation of power and vicinity diameter as well as density; the energy grows in order of magnitude of $O(n/log(n)^{log(n)})$ where n indicates the number of nodes within the vicinity. This is derived from the vicinity density definition, which is expressed as the number of nodes divided by the vicinity diameter. Vicinity diameter is studied in Section V-F.

D. Internet Power consumption

P2P-clouds for inter-vicinity communication and classic datacenters for communication with users rely on Internet. Thus, to analyze the energy consumption of these systems, we should be aware of Internet energy consumption as well. Power drawn in Internet is subject to the hardware and distances exploited. Internet infrastructures are classified as core, distribution and access. 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. [17] conducted a study on the prevalent Internet hardware energy consumption. We rely on this study for the Internet power consumption part of our analysis by driving the model in (6). In this model, $P_{Internet}$ stands for Internet power consumption which is a combination of power drawn in core $P_{core} = P_{core}^{router} + P_{core}^{link}$, distribution $P_{dist} = P_{dist}^{router} + P_{dist}^{link}$ and access $P_{access} = P_{access}^{router} + P_{access}^{link}$ levels. P_{core}^{router}, P_{dist}^{router} and P_{access}^{router} denote router power consumption in core, distribution and access level respectively; whereas, P_{core}^{link}, P_{dist}^{link} and P_{access}^{link} illustrate the link power drawn in different levels.

$$P_{Internet} = P_{core} \times n_{core}^{hops} + P_{dist} \times n_{dist}^{hops} + P_{access} \times n_{access}^{hops} \tag{6}$$

In this work, we compare the energy consumption of typical MapReduce jobs on P2P-cloud and classic datacenter. Running MapReduce on P2P-cloud conforms to the idea of desktop grid as in [7].

IV. MapReduce Case Study

To scrutinize the energy consumed in the clouds, we analyze the energy consumed per MapReduce job, both in the datacenter and P2P-clouds. When a MapReduce request is sent to a datacenter, the scheduler decides which host 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 datacenter; 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 datacenter communication infrastructure includes VM and input data. To transmit the data, it should be split into messages that are suitable for transmission protocol, i.e. less than Maximum Transmission Unit(MTU_p), namely, 1500 bytes for Ethernet [18]. If the network interface is Ethernet, an overhead, O_p, of around 1% is expected [18]. 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, the output of the first phase, 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 number of packets to be transmitted is as in (7). $Size_{VM}$ and n_{host} denote the VM size and the number of hosts assigned to the job respectively. The energy consumed to transmit the required data for a job, as shown in (8) is the multiplication of power drawn for the communication, the number of messages per MapReduce job as depicted in (7) and the time to transmit a packet, i.e. t_{send_packet}, since $energy = power \times time$.

$$E_{intra\_DC}^{MR} = P_{DC\_comm}^{intra\_DC} \times t_{send\_packet} \times N_{msgs}^{MR} \tag{8}$$

t_{send_packet} is computed as the division of the packet size by transmission rate in the network $\frac{MTU_p}{R_{intra\_DC}}$. For instance, exploiting Gigabit Ethernet, t_{send_packet} is less than 0.5 ms [18].

The energy drained within each host is $\sum_{n_t} P_{host} \times t_{task}$ for each phase. 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 [19], 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 Internet as illustrated in (6). If the latency in each element and layer is known, the energy consumed in Internet for routing the MapReduce job data is measurable. This latency is related to the number of packets that should be transmitted over Internet, which is equal to $\frac{Size_{input}+Size_{output}}{MTU_p-O_p} \times MTU_p$. The only data to be exchanged over Internet in this case is the input and output data.

To analyze the energy consumed in the P2P-cloud per MapReduce job, we should consider two different scenarios. A case where jobs are assigned to the hosts within a vicinity, i.e. intra-vicinity, and the second case for inter-vicinity responses. In 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. On the other hand, in case of intra-vicinity responses, VM, input and intermediate output data need only to be sent to a host via wireless network. To exemplify, considering IEEE 802.11a

$$N_{msgs}^{MR} = MTU_p \times \frac{Size_{input} + n_{hosts} \times Size_{VM} + \sum_{i=1}^{n_t} Size_t(i) + Size_{intermediate\_output} + Size_{output}}{MTU_p - O_p} \tag{7}$$

TABLE I: VM Specifications

Type	Cores	Memory(GB) (GB)	Storage(GB) (GB)	number of mappers	number of reducers
Small	1	1	1	1	1
Medium	1	3.75	4	1	1
Large1	2	7.5	32	1	1
Large2	2	7.5	32	2	2

wireless infrastructure and IPv4 packets, the transmission rate, R_{intra_P2P} is 52Mbps and time to send a packet, t_{P2P}, is less than 10ms. In this case the number of messages to transmit over the community network follows (7) considering wireless MTU, protocol overhead and packet loss. Overall, the energy required to accomplish a MapReduce job on community for the intra-vicinity mode is given in (9). t_{P2P} implies the response time of the hosts in P2P-cloud.

$$\begin{aligned} E_{intra\_P2P}^{MR} = {} & P_{WN}^{comm}(Size_{message}) \times t_{P2P} \times N_{MR}^{msgs} \\ & + \sum_{i=1}^{numberofphases} (E[t_{task}] \times \sum_{n_t} P_{host}) \end{aligned} \tag{9}$$

As stated, the MapReduce workload is composed of input data, intermediate output data and VMs that contain the computing platform. A remarkable amount of energy is consumed to transmit the VM packets over community network. To alleviate the burden of VM transmission, in this work we introduce the caching mechanism for saving most prevalent VMs in community nodes, i.e. P2P-cloud with cache. In this way, we can save the energy required to transmit the VM over the community each time.

V. Evaluation

A. Experiment Setup and Scenarios

We aim to analyse the energy consumption on different cloud models under the MapReduce workload with the following configuration. For the P2P-cloud nodes we rely on the Clommunity [20] which employs the Jetway JBC362F36W with Intel Atom N2600 CPU with the maximum power of 20W, as well as the Dell OPtiplex 7010 desktop machines. Datacenter hosts are set to be HP Pro Liant Ml110G3 Pentium D930. For the HP machines power model is derived from the SPECpower_ssj2008 benchmark [21]. The community cloud infrastructure is modelled as wireless network which employs flooding as routing strategy, i.e. the worst case energy consumption scenario. Each wireless antenna consumes the maximum of 5.5 watts. For the switches in the LAN, we apply the power model introduced in [22]; Internet energy consumption values are derived from [17]. Four VM types as shown in Table I are exerted. For most scenarios, we assumed a typical workload of input data size of 15 GB, overall intermediate output size is 30% and the final output size is 20% of the original input. For the sake of comparison through this evaluation, we take small VMs to execute the tasks, unless it is explicitly mentioned. We study our main metric, i.e. energy consumption in the following scenarios:

A. **P2P-cloud without cache:** the base P2P-cloud scenario, assuming that the entire contents of workloads have to be downloaded via wireless, but are always available within the vicinity.
B. **P2P-cloud with cache:** same as above scenario, but enhanced with caching locally to nodes the most popular VMs and data files within the vicinity, thus reducing the amount of repeatedly downloaded information. Note that in this scenario and the scenario above we assume that resource scarcity never occurs.
C. **P2P-cloud with inter-vicinity responses:** the worst case P2P-cloud scenario, the base one but the content is not available within vicinities, thus accounting for inter-vicinity communication and extra costs.
D. **P2P-cloud with cache and inter-vicinity responses:** same as above, extended with local caching of VMs and data files, thus reducing the amount of repeatedly downloaded information.
E. **Classic datacenter:** For comparison against the classic datacenter scenario, where users access the datacenter exclusively through wired networks, we exploit the datacenter model with 4 rows of 32 clusters each with 32 hosts for the datacenter model.

B. Idle case energy consumption

Comparing the energy consumed when the machines are idle, we find that the energy consumed in idle state for the datacenter model is higher than that of the P2P-clouds. In datacenter model all communication devices as well as hosts consume power, i.e. static power, while in the community cloud machines are on not exclusively to serve the cloud. Hosts are on basically to process the users individual processes, no idle states are defined for the machines. Moreover, examined P2P-cloud devices', i.e. Jetway JBC362F36W, energy consumption is negligible. The nominal power for such devices are 20 watts and they dissipate 10 watts in the average utilization. The only static power drawn in the community clouds is because of the wireless network interfaces for intra-vicinity communication. What is more, in the datacenter model, cooling power is imposed to the power cost, while in P2P-cloud, it is not required to set up a cooling system. For the cooling model we refer to [4]. The simple fan cooling model is applied to the datacenter model.

C. P2P-cloud Energy Consumption

In Figure 3, we show the energy consumption for each of the defined scenarios as the workloads vary across two parameters, VM size and input data size. Naturally, energy

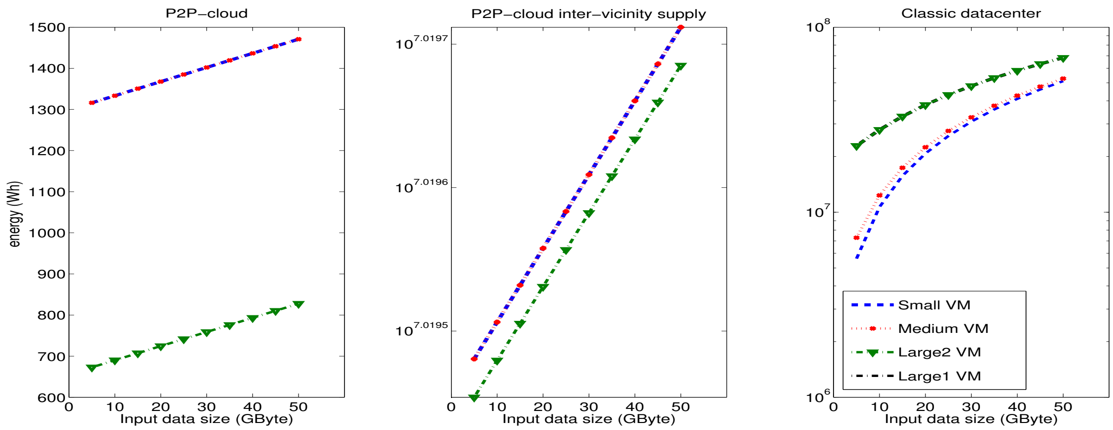


Fig. 3: Energy consumption for various inputs across scenarios

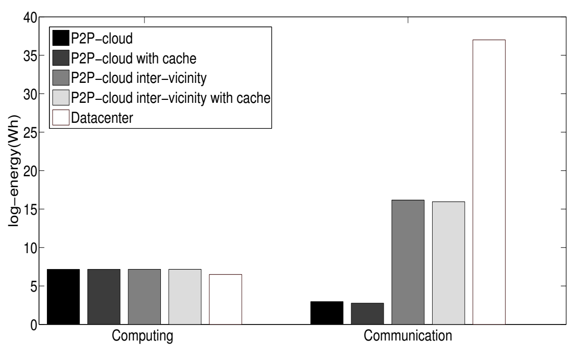


Fig. 4: Compute vs. communication energy consumption

consumption increases for workloads executing larger VMs and when processing larger input data files. Comparing to the classic cloud, P2P-cloud consumes quite less energy as long as the jobs are performed locally. Generally, the energy required to accomplish jobs in datacenter model exceeds that of the P2P-cloud in any cases if the input size is big enough or the VM is large. However, we should bare in mind, this energy saving occurs by sacrificing the performance.

As shown in Figure 3, the energy consumption in P2P-cloud in case of providing the service within the vicinity is much less than the case of inter-vicinity scenario, since in the inter-vicinity service provisioning we should transmit the input, output data and in some cases the VMs through Internet, which is the most energy hungry element of the P2P-cloud system. In general, the communication energy is fluctuating more P2P-clouds, while the processing energy is more varying in classic datacenters.

Figure 4 outlines the energy consumption in computing and communication part for small VMs with the input size of 20GB. This Figure proves that the energy consumption of P2P-cloud in communication part is varying more, since we can see different values for different scenarios.

D. VM size effect

As shown in Table I we consider three different types of VMs with different capabilities of processing MapReduce tasks. Figure 3 highlights the effect of VM size in MapReduce task processing in three scenarios. As depicted, the energy consumption in P2P-cloud intra-vicinity processing is neutral to VM size, but is dependent of the MapReduce task processing slots available in the VM. Including more slots in a VM, we save more energy, since less communication overhead is induced. The energy consumption of communication in P2P-cloud constitutes an enormous portion of the consumption and even more than computation cost. Although increasing the level of parallelism within a VM can improve the energy saving, it should be bared in mind that in P2P-cloud the processing power of the nodes are very limited and we cannot create large VMs there. Nevertheless, increasing the task collocation in classic datacenter hosts can be a more practical solution for energy saving purposes. As shown, energy consumption of inter-vicinity scenario is independent of the VM size as long as the VM images are available in the serving vicinities, since the input and output data transmission energy dominates the process energy consumption.

Increasingly, Figure 3 reveals the importance of choosing the right VM according to the input size besides choosing the appropriate platform. To exemplify, in a classic datacenter for the input size of less than 10 GByte, processing on small VMs is the most energy efficient choice due to the process power saving of small VMs.

E. Input-(intermediate) output Proportionality

Here we study the relation of intermediate output and output size of the MapReduce applications on the energy consumption to get an insight into the appropriate VM as well as system to run different MapReduce applications. Figure 5 illustrates the importance of VM selection for applications with smaller

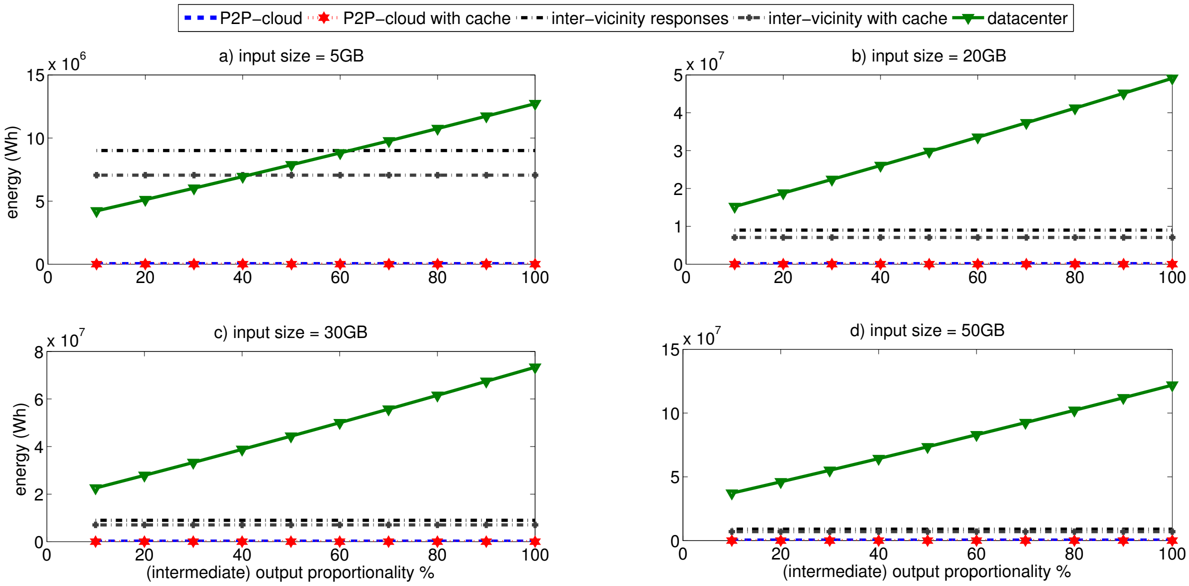


Fig. 5: Energy consumption of applications with different input-output sizes running on small VMs

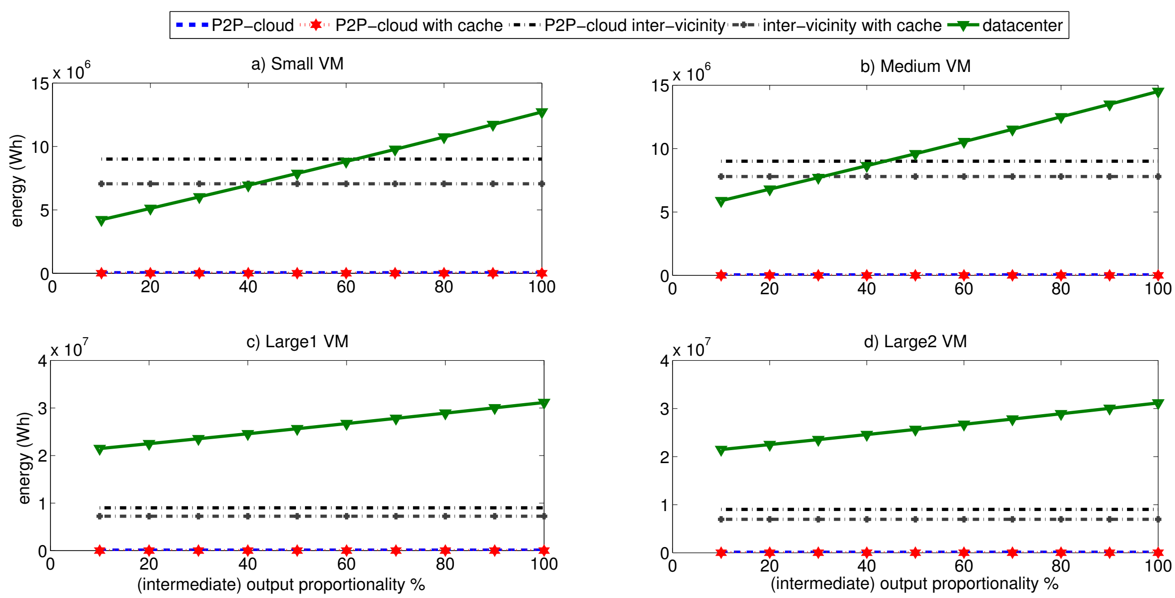


Fig. 6: Energy consumption of a 5GB input application running on different VMs across scenarios

input and output sizes. As shown in Figure 5 in cases that input size is small, i.e. 5GB and the output is less than 40% of input data, datacenter model outperforms the inter-vicinity scenario.

Figure 5 focuses on small VM. To be more precise, we draw the energy consumption for small inputs across different scenarios including different VMs in Figure 6 because the intermeidate-output has to be exchanged among vicinities in this case. As depicted in Figure 6, in small and medium VMs there is a cross point among datacenter energy consumption and inter-vicinity responding in P2P-cloud, Figure 6.a , Figure 6.b . However, for the large VMs, energy consumption of datacenter always exceeds the P2P-cloud scenarios even for the small input size, Figure 6.c , Figure 6.d.

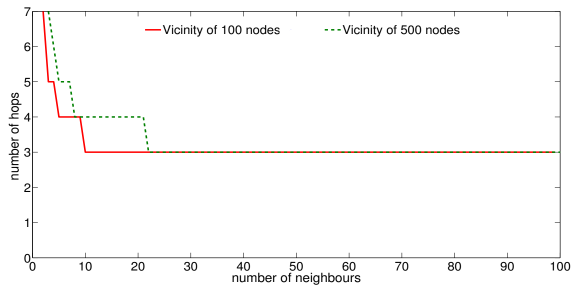
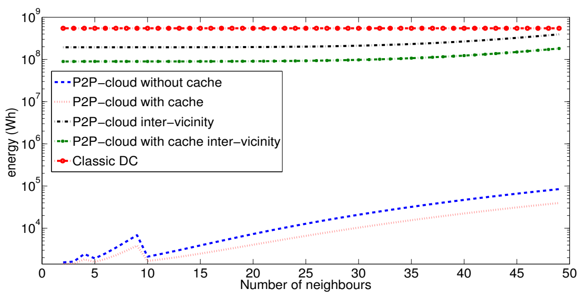
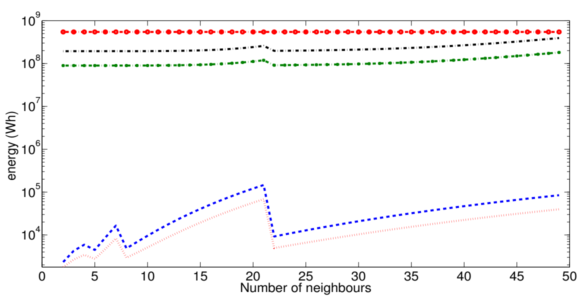


Fig. 7: Impact of number of neighbours in vicinity diameter on average hops between two nodes.



a) Vicinity size = 100



b) Vicinity size = 500

Fig. 8: Vicinity Density effect in community networks

F. Vicinity Density

Here we exert the logarithmic vicinity diameter model which implies the average distance of two nodes in the vicinity as $O(log^n_{neighbourCount})$ where n denotes the scale of the system. Figure 7 shows that with the number of neighbors of at least 10, P2P-cloud scenarios can keep the average number of hops between two nodes in the vicinity, where there are 100 nodes overall in the vicinity. Convergence to three hops for a vicinity of 500 nodes occurs in around 30 neighbours. Although three hops is very effective, increasing the number of neighbours not only leads to higher energy consumption due to multiple unaddressed recipients, but also does not provide additional gains in message latency. Nonetheless, adding more nodes increases the resource availability in each vicinity. Therefore, there is a trade-off between energy efficiency and resource availability.

In Figure 8, we depict energy consumption for typical workload presented earlier for all the scenarios described, with two different vicinity sizes: 100 and 500. P2P-cloud with caching, our proposal, is clearly the winner, with orders of magnitude less energy consumed, in both scenarios. Figure 8 also reveals the influence of the vicinity density, i.e., the number of neighbors accessible to each node. The P2P-cloud with caching is always the winner regardless of the vicinity density. The fluctuation in the graph for small number of neighbors is because of the estimation and round up error in the logarithmic vicinity diameter model, but by reaching the efficient average hop count, i.e. three for aforementioned scenarios, energy consumption rises gradually as the vicinity becomes denser.

G. Impact of Cache Scale

By addressing the implications of cache size in P2P-cloud, in Figure 9, we illustrate how the energy consumption varies when the P2P-cloud scenarios (scenarios: A, Figure 9.a and C, Figure 9.b) are enhanced with caching (scenarios: B, Figure 9.a and D, Figure 9.b), and with different scale parameters (the higher the scale factor, the higher the popularity and probability of finding less common VMs). This results in lower energy consumption as the number of neighbors increases in each vicinity, since the network becomes denser, i.e. more interconnected.

H. Discussion

From the obtained results we conclude that if the job is processed in the P2P-cloud, we can save more energy in almost all cases- except for the small input-output sizes running on small VMs, Figure 5, Figure 6. However, it is more energy conservative, with orders of magnitude, if we manage to execute the tasks in the intra-vicinity mode. Practically, this is not always possible, since the resources in each vicinity are very limited. Increasingly, although P2P-cloud outperforms classic datacenter model in terms of energy saving, it increases the execution time drastically, which impacts the performance. Therefore, P2P-cloud cannot be a good option for the tasks with closer deadline. But fortunately, this is rarely the case for MapReduce tasks.

Nonetheless, in our analysis and evaluation we ignored replication. However, multiplying all the aforementioned equations and results by the replication factor, we can include the replication energy consumption as considered in [3].

VI. Conclusion and Future Work

In this paper we compared the energy consumption in classic datacenter model with P2P-clouds. We scrutinized the main sources of energy consumption in both systems under various scenarios and found out the most and least energy efficient elements in each system. Our analysis revealed that P2P-cloud

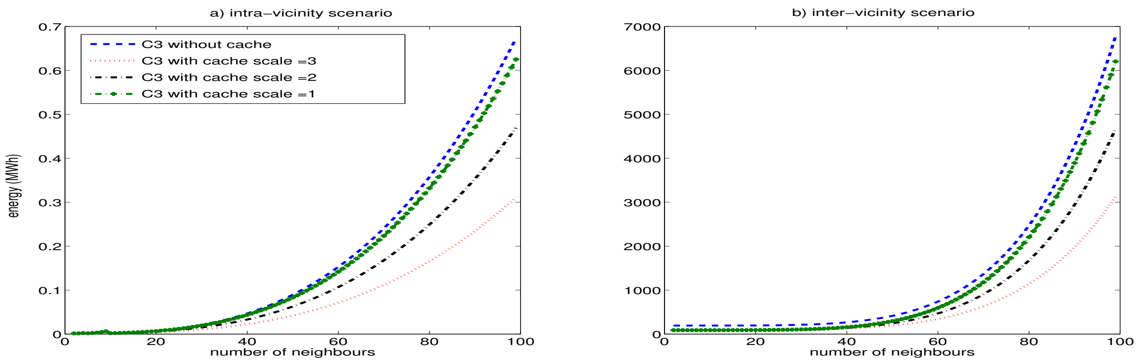


Fig. 9: Impact of cache scale in energy consumption

outperforms the classic datacenter model in terms of energy efficiency, as long as the jobs are served locally. Nonetheless, there is room to improve the efficiency of P2P-cloud via energy concerned scheduling and resource management mechanisms.

In this work we came up with the P2P-cloud in intra-vicinity responses as the most energy efficient solution. However, the effect of exerting P2P-cloud on quality of service was not studied. As a future work in this line, we intend to elaborate the quality of service for services offered by P2P-cloud, while considering energy efficiency mechanisms.

Acknowledgments

This work was partially supported by the European Community through the projects A Community Networking Cloud in a Box (Clommunity), FP7-317879, and also by Spanish government under contract TIN2013-47245-C2-1-R, and also by the Generalitat de Catalunya as a Consolidated Research Group 2014-SGR-881. This work was supported by Portuguese national funds through FCT Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2014.

References

[1] A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya *et al.*, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, no. 2, pp. 47–111, 2011.

[2] 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.

[3] 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.

[4] J. Doyle, R. Shorten, and D. O'Mahony, "Stratus: Load balancing the cloud for carbon emissions control," *IEEE Transaction on Cloud Computing*, 2013.

[5] F. F. Moghaddam, M. Cheriet, and K. K. Nguyen, "Low carbon virtual private clouds," in *Cloud Computing (CLOUD)*, 2011, pp. 259–266.

[6] 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 (CNBuB 2013), within IEEE WiMob*, Lyon, France, Oct. 2013.

[7] 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.

[8] X. Zhiqun, C. Duan, H. Zhiyuan, and S. Qunying, "Emerging of telco cloud," *Communications, China*, vol. 10, no. 6, pp. 79–85, 2013.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] C. Belady, A. Rawson, J. Pfleuger, and T. Cader, "Green grid data center power efficiency metrics: Pue and dcie," Technical report, Green Grid, Tech. Rep., 2008.

[14] G. Varsamopoulos and S. K. Gupta, "Energy proportionality and the future: Metrics and directions," in *Parallel Processing Workshops (ICPPW)*. IEEE, 2010, pp. 461–467.

[15] 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.

[16] 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.

[17] 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.

[18] C. Hornig, "A standard for the transmission of ip datagrams over ethernet networks," 1984.

[19] T. A. de Ruiter, "A workload model for mapreduce," *MSc thesis at TU Delft.*, 2012.

[20] Clommunity project, http://wiki.clommunity-project.eu/.

[21] Spec power benchmark, https://www.spec.org/benchmarks.html.

[22] P. Mahadevan, S. Banerjee, and P. Sharma, "Energy proportionality of an enterprise network," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*. ACM, 2010, pp. 53–60.