Research Proposal

Economics Inspired Energy Aware Service Provisioning in P2P Assisted Cloud Ecosystems

by

Leila Sharifi

July 2015

Abstract

Energy has emerged as a first class computing resource in modern systems. The trend has primarily led to the strong focus on reducing energy consumption of data centers, coupled with the growing awareness of the adverse impact on the environment due to data centers has led to a strong focus on energy management for server class systems.

The focus on energy management has been cross-cutting across various computing disciplines including computer architecture (hardware design), hypervisors, operating systems and system software.

In this work we intend to address the energy aware service provisioning, exerting economics-inspired mechanisms. Toward this goal, we should tackle the following challenges which are addressed in this work. To frame an energy aware service provisioning mechanism in P2P-assisted cloud, first we need to compare the energy consumption of each individual service in P2P-cloud and data centers. However,in the procedure of decreasing the energy consumption for cloud services we may be trapped with this pitfall that energy consumption decreases remarkably, but the performance is violated at the same time. Users may leave a system if they do not get 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. Therefore, wee need to formulate a performance aware energy analysis metric, conceptualized across the service provisioning stack.

Afterwards, we sketch a framework to analyse the energy effectiveness in P2P-assisted cloud platform to choose the right service platform according to the performance and energy characteristics mapping from the hardware oblivious, top level to the particular hardware setting in the bottom layer of the stack. Then, we introduce an economics-inspired mechanism to increase the energy effectiveness in the P2P-assisted cloud platform as well as moving toward a greener ICT for ICT for green platform.

Contents

Chapter 1

Introduction

Energy and associated environmental costs (cooling, carbon footprint, etc.) of IT services constitute a remarkable portion of dynamic cost. Indeed, there is a need for an energy aware economic model which includes service pricing, resource allocation and scheduling. Since the energy price is going to predominate for the services, it is required to device an energy-based pricing mechanism for each service. To this end, we need to formulate a per job energy consumption estimation technique to schedule resources in an energy efficient manner.

Although there is a growing body of work centered on the energy aware resource management, allocation, scheduling and pricing [1–4], they mainly considered the whole system energy measurement, estimation, improvement and optimization. There is only limited work focusing on the energy issues per job [3, 5–7]. However, they only aim at reducing total energy consumption in the infrastructure without taking into account the energy-related behavior of each individual job, its performance and its price, i.e., how expensive and useful is the energy employed for the observed job performance or progress.

Energy-based job pricing confronts some more challenges further to the system wide energy efficiency issues. In the system wide energy efficiency, the energy consumption of the resources are measurable simply by plugging the energy meter devices or exploiting the embedded sensors of the contemporary devices, e.g. Runtime Average Power Limit (RAPL) counters in recent Intel CPUs. Nonetheless, it is nontrivial to measure the energy consumed per job, since we cannot embed a physical sensor in a job or plug a metering device to it. Therefore, estimation is still the only option in this case. Estimation results in a more complicated model since it has to deal with uncertainty and error. Moreover, the need for a general approach that assumes an unpredictable workload behavior aggravates the problem. Increasingly, non-energy proportional hardware infrastructure adds excess complexity to the agenda in the multi-tenant ecosystems.

By estimating the energy consumption of each job, we focus on assigning it to the hosts which incur the lowest energy price. To this end, we need an energy aware resource manager which is aware of the power sources. The more diverse the energy providers, the greater the variety of user choices. Smart grids, naturally fulfil this goal. Diverse energy sources of smart grid improves the availability, sustainability and environment friendliness of the cloud services. Hence, combing the cloud infrastructure with smart grid can improve the economic model of both systems.

1.1 Problem Formulation

We notice several requirements for an economic model for the community cloud systems which are not well developed so far. Energy efficiency of community clouds in comparison with classic cloud model is still a matter of controversy. Furthermore, energy efficiency has not been a priority concern in the end-user incentive list.

This will tend to change in the midterm future as domestic prices of electricity rise and the profiles of energy sources are factored in the prices (e.g. by computing utility providers/distributors).

The main objective of this work is to come up with an economic model for the community clouds which is centered on the energy based pricing of the services that embed the energy efficiency in user incentive list and reduce the carbon footprint to be more environment friendly.

The distinguishing point of our model is considering the energy from a consumer perspective, i.e. per job, in lieu of the coarse grain provider vantage point. Note that We do not intend to reduce the energy consumption. Our ultimate goal is to increase the energy efficiency through the pricing mechanism. For this purpose, a P2P assisted cloud, as studied in [P2, P5, P4], outperforms classic data center oriented cloud architecture due to the diverse range of processing elements scattered all through the system, which can accomplish certain sort of tasks with lesser energy dissipation.

By estimating the energy consumption of each job, we focus on assigning it to the resources which incur the lowest energy price. To this end, we need an energy aware resource manager which is aware of the power sources. The more diverse the energy providers, the greater the variety of user choices. Smart grid, naturally fulfils this goal. Diverse energy sources of smart grid improves the availability, sustainability and environment friendliness of the cloud services. Hence, combining the cloud infrastructure with smart grid can improve the economic model of both systems.

1.2 Research Questions

To formulate an economic model for energy aware service provisioning the following questions should be addressed.

Q1- Is it energy efficient to switch to community cloud?
Studies on [P2, P5] reveal that there is no straight forward answer to this question, since the answer not only depends on service specification, but also partially depends on the hardware setting, the service is running on. Therefore, to answer this question we need a framework to analyse energy consumption of a service in different platforms, as we discuss in Chapter 3.

We should note that decreasing energy may result in performance plunge. Therefore, a performance aware energy analysis metric is needed in the introduced framework. This metric should be able to attribute energy and performance across the service provisioning stack. The details of this issue are discussed in Chapter 2.

Associated publications forming and addressing this question are [P2, P5, P4].

Q2- How can we exert cheap, green, distributed energy sources?
Energy and Information and Communication Technology(ICT), as two pillars of the contemporary life, are reshaping themselves based on ubiquitous society architecture to improve their service quality. Within the reforming process, integration of two systems can contribute to a greener ubiquitous society by equipping them with the concept of energy conservativeness, and leveraging renewable energy sources. We outline the idea of Cloud of Energy(CoE), Chapter 4, which fosters the adoption of green energy and green cloud by combining these two systems. CoE introduces an integrated framework of everything as a service(XaaS) to facilitate the service exchange, not only across the computing and electricity grid hierarchy, but also among these two systems via an economic middleware. This middleware embraces service pricing, resource allocation and scheduling as we discuss in Section 4.2. Related publications are [P1, P3].

1.3 Background and Hypothesis

In this section, we outline the elaborated cloud model as well as terms and hypothesis employed all through this document. We initially address the relevant cloud topologies: i) the classic data center, ii) the peer-to-peer cloud that is a precursor of the community cloud, and iii) cloud federation. Then, we address virtualization in the context of data centers as well as in peer-to-peer deployments.

1.3.1 Cloud Platforms

Here, we address initially the relevant cloud topologies: i) the classic data center, ii) the peer-to-peer cloud that is a precursor of the community cloud, and iii) cloud federation.

1.3.1.1 Classic Data Center

In the classic data center model, from which the idea of cloud computing stems from, a gigantic data center embraces a number of clusters of hosts constituting a powerful computing or storage capacity. The internal organization and hierarchy of the data center can follow a number of variants typically aiming at reducing latency and energy consumption in processing and internal traffic.

1.3.1.2 P2P-cloud

A P2P-assisted cloud or P2P-cloud comprises a number of vicinities, each one typically composed of a set of commodity hosts, including Internet of Things boards, laptops and desktop PCs, connected via a wireless communication platform as depicted in Figure 1.1. The main goal of a P2P-cloud is to take advantage of smaller distributed datacenter hosts as well as exploiting the commodity hardware of community networks.

As a special case, 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 [8]. The P2P-clouds we address in this paper have the vision of a cloud deployment in community networks: a

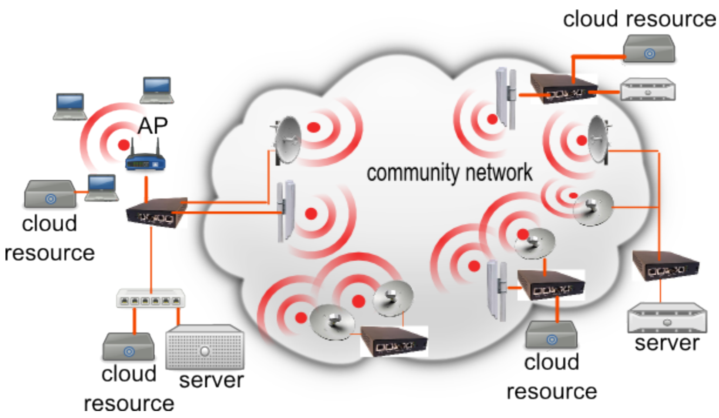


Figure 1.1: P2P-cloud intra-vicinity model

cloud hosted on community-owned computing and communication resources providing a diverse range of services.

Comparing P2P-cloud with desktop grids [9], we find out that desktop grids are a peer-to-peer volunteer computing platform. However, P2P-cloud services are not confined 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 above mentioned concepts. To exemplify, P2P-cloud hosts may be mobile or static. P2P-cloud reinforces the concept of telco-cloud [10], 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. As an example, a Raspberry Pi device, running light weight Internet of Things user applications can devote its unused capacity to run community 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 lesser energy compared 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 1.1. Each vicinity can access the others via Internet; this is known as inter-vicinity communication.

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 [11–13].

Previous studies have revealed 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[14]. 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.

On an even more decentralized scale, Fog Computing, as a new paradigm of wireless data transfer via distributed devices of Internet of Things (IoT), supports the idea of P2P-cloud. Fog computing introduces a hierarchical distributed architecture extended from the network edge to the core that provides a geo-distributed platform to improve location awareness and latency, by pushing the service provisioning to the edge of the network. This architecture perfectly matches the new data traffic paradigm of IoT, where data is big in the number of sources rather than the volume and helps in smarter big data processing.

However, fog computing is not a substitute for the cloud, it is a complement. By controlling data at various edge points, fog computing integrates core cloud services with those of a truly distributed data center platform. This infrastructure maintains the concept of the cloud while incorporating the power of fog Computing at the edge.

1.3.1.3 Cloud Federation

The cloud is mostly about the elasticity and flexibility and the network architecture is a key component that helps drive these properties. The challenges of supporting business continuity in a cloud environment are not limited to a physical area or the datacenter alone. 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 data centers, alongside with bringing high availability and disaster recovery, provide the opportunity to have different energy sources.

Federated cloud conforms to the same architecture of a distributed datacenter. 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 datacenter model. Therefore, power efficiency of federated cloud and distributed datacenter are expected to be very close.

Federation of P2P-cloud and data centers through the concept of the fog, elevates the popularity of cloud systems due to the advantages of reduced latency, higher availability and cheaper services and better quality of service.

Service prices can be reduced by pushing the computing toward the commodity devices at the edge of the network; however, data centers still support the services in the backbone in case of failure or if a service demands specific computing requirements which can be better provided via the data center servers, e.g. parallel data processing in specific MapReduce scenarios [P2]. In interactive applications, P2P-cloud platform can decrease the latency compare to the data centers by local service provisioning in a geo-distributed platform.

1.3.2 Virtualization

Here, we address virtualization as a driving force of cloud computing, first in the context of data centers, and then integrated with peer-to-peer deployments.

1.3.2.1 Virtualization in Data Centers

With the advent of server-side computing as a service, provisioning resource guarantees and isolation in multi-tenant environment became of utmost importance. It became imperative that these infrastructures satisfy the goal of application isolation and resource efficiency. In order to achieve this, the infrastructure economics must allow servers to be shared among multiple users and at the same time guarantee operational isolation of applications. Virtualization is the most widely adopted solution to guarantee these goals. Consolidation by using virtualization leads to application isolation, better resource utilization and lower operational costs.

Typically, there are three types of virualization: Container virtualization, Kernel virtualization and Hypervisor based virtualization. Each have their own hosts of benefits and demerits. Typically cloud based solutions for virtualization rely on Hypervisor based virtualization such as Xen since it can support different flavours of operating systems such as linux, windows etc. Additionally, with the proliferation of hardware virtualization in most modern architectures, this allows guest operating systems to run unmodified. However, Hypervisor based virtualization can suffer from performance degradation as they incur an additional overhead of VM management by the Hypervisor. Container virtualization on the other hand can execute applications at near native speed since they have no additional layer of routing and share the same kernel. Nonetheless, since the kernel is shared, they can only run guest operating systems that support the host kernel. On the other hand, Kernel virtualization is becoming an attractive alternative since each guest can have its own kernel and the host contains a modified kernel with extensions to manage and run multiple VMs.

1.3.2.2 Virtualization in P2P context

Although multiple virtualization techniques exist, virutalization in the context of P2P presents specific challenges that need to be addressed. P2P is typically comprised of commodity machines that are not server grade and as a result do not have high computational capabilities. They are limited by the available resources and it becomes imperative to ensure that virtualization does not impose a high overhead on these resource limited machines. Hypervisor based virtualization techniques like Xen impose a high overhead because of an additional level of routing at the expense of running any operating system. Container virtualization on the other hand has minimal overhead and executes guest applications at near native speeds. Since they share the kernel with the host OS, it has limited support in terms of multiple operating systems yet provides consolidation with resource provisioning guarantees at near native speed. Container virtualization is thus a better fit for such environments with limited resources. KVM is also an attractive alternative for resource limited environments since it imposes minimal overhead. However, each guest can have its own kernel and this has a performance impact when spawning a new VM. LXC has a lesser overhead from this perspective. Despite the host of relative advantages and disadvantages, LXC appears to be a reasonable choice of virtualization technique to adopt in a P2P environment.

Virtual machine density refers to the number of virtual machines a physical host can maintain, while providing enough compute resources for every virtual machine to perform well. It depends on multiple factors such as: server hardware, virtualization software, service type and workload diversity. These varying factors make it difficult to come up with an absolute number for virtual machine density across all scenarios. Driving up the VM density reduces the cost incurred but at the same time introduces additional challenges in guaranteeing performance because of contention in shared resources. Typically the bottleneck manifests at the memory subsystem and the amount of available memory. Commodity machines are scarce is such resources and as such the system will benefit from conservative provisioning in order to provide best-effort guarantees.

1.4 Publications

List of the relevant publications is as below:

[P1] Sharifi, Leila, Felix Freitag, and Luis Veiga. "Combing Smart Grid with community clouds: Next generation integrated service platform." In Smart Grid Communications (SmartGridComm), 2014.

[P2] Sharifi, Leila, Navaneeth Rameshan, Felix Freitag, and Luis Veiga. "Energy Efficiency Dilemma: P2P-cloud vs. datacenter." In CloudCom2014, **Best Paper Candidate**.

[P3] Sharifi, Leila, Felix Freitag, and Luis Veiga. "Envisioning Cloud of Energy." Submitted to Smart Grid Communications (SmartGridComm), 2015,**Under Review**.

[P4] Sharifi, Leila, Jose Simao, Navaneeth Rameshan, Felix Freitag, and Luis Veiga. "A Framework to Analyse Energy Effectiveness in P2P Assisted Cloud Ecosystems." Submitted to IEEE Transactions on Cloud Computing, **Under Review**.

[P5] Sharifi, Leila, Llorence Cerda-Alabern, Felix Freitag, and Luís Veiga. "Energy Efficient Cloud Service Provisioning: Keeping Data Center Granularity in Perspective." Submitted to Journal of Grid Computing, Special Issue on Green Cloud Computing, **Under Review**.

1.5 Outline

This document is organised as shown in Figure 1.2. In this chapter, i.e. Chapter 1, we stated our problem and defined the associated research questions and outlined the system model that we study our problem in as well as the hypothesis. Next chapter, Chapter 2, formulates the performance aware energy analysis metric required across the service provisioning stack and the analysis framework is formed in Chapter 3. Afterwards, we introduce a possible solution to exert distributed renewable energy sources of smart grid as well as distributed processing elements of P2P-cloud assisting mass producers to achieve a greener ecosystem in Chapter 4.

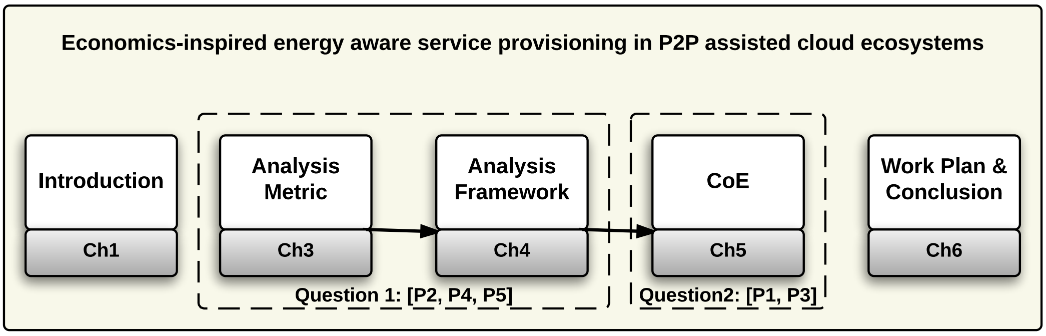


Figure 1.2: Outline

Chapter 2

Energy Analysis Metric

Energy has emerged as a first class computing resource in modern systems. The trend has primarily led to the strong focus on reducing energy consumption of data centers, coupled with the growing awareness of the adverse impact on the environment due to data centers has led to a strong focus on energy management for server class systems.

The focus on energy management has been cross-cutting across various computing disciplines including computer architecture (hardware design), hypervisors, operating systems and system software [15]. Figure 2.1 captures the various techniques developed to reduce energy consumption across the service provisioning stack.

In the procedure of decreasing the energy consumption for cloud services we may end up with this pitfall that energy consumption decreases remarkably, but the performance is violated at the same time. Users may leave a system if they do not get 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.

2.1 Background and Related Work

In this section we introduce the service provisioning stack. Moreover, the overview of previous research on power modelling of all layers of the stack is given in this section.

2.1.1 Service Provisioning Stack

Three main components involved in service provisioning are application, operating system and hardware. In virtualized platforms, operating system is replaced by the Virtual Machine (VM), as demonstrated in Figure 2.1. In each layer of this stack power and performance may be attributed to the different metrics. Namely, in application layer, performance is generally translated to latency, whereas in VMs it is mapped to SLA metrics and throughput is the interpretation of performance in the hardware level.

All the same, across the stack we need a translation to hardware agnostic power model in application layer, and partly in VM/OS level, which should be mapped to a hardware aware model at the bottom of the stack in run time. In the following section we review

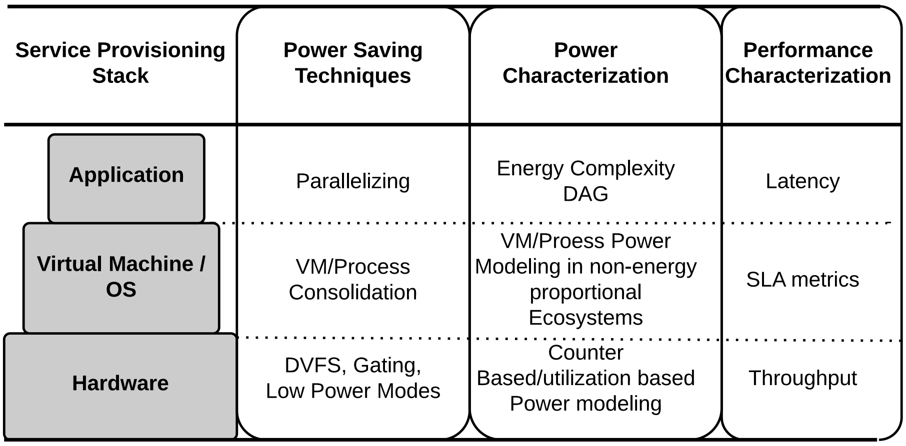


FIGURE 2.1: Energy and performance conceptualization across the service provisioning stack

the state of the art power characterization techniques across the service provisioning stack.

2.1.2 Power Characterization

Research in power modeling can be broadly classified into (i) Simulator-based, (ii) CPU Utilization-based (iii) Event or performance counters based and (iv) Coarse-grained. Early power management research exerted analytic power models based on voltage and frequency [16], which are fast, but only provide rough estimates. Coarsegrained estimates based on the type and state (active, off) of the processor have been used in [17]. However, with the increase in the dynamic power range of servers, a more accurate power prediction method is needed.

Power models based on readily available system parameters like CPU utilization [18] are possibly the simplest to use for algorithm design. A CPU utilization based model is currently the most popular power estimation model used in practice [19]. CPU utilization can possibly be estimated roughly using the computational complexity of an algorithm. However, different applications make differing use of various CPU units and other system resources like memory and a CPU utilization model is not accurate across wide application categories.

Interestingly, the workload-sensitive nature of CPU-based models has been recently cited as a reason to go back to using detailed event counters in [7] for predicting processor and memory power usage under voltage scaling. Application-aware power modeling has the potential to assist energy aware algorithmic engineering. In [20], the authors create power profiles for each application and use it to estimate the power drawn by a consolidated server hosting the applications. WattApp [21] exploits power profiles for an application and estimates the power consumed with changes in workload as well. A good comparison of various system-level power models is presented in [22]. However, all these techniques are measurement-based, whereas algorithmic engineering needs energy models that are based on first principles. A good comparison of various system-level power models is presented in [22].

The first asymptotic energy model for algorithms is presented in [23]. As is the norm with traditional algorithm design, the energy model is asymptotic. A couple of simulation results were presented, where it was shown that algorithms designed for the "traditional" algorithmic model can be transformed into algorithms whose memory accesses are highly parallelized and thus, consume much less energy than the naive implementation. These simulations led to design of energy optimal (in the asymptotic sense) algorithms for certain basic problems such as sorting and matrix transpose. However, implementations needs to be aware of the exact constants used in the complexity model for practical implementation. Model in [23] has similarities with the cache oblivious model [24]. In this model, the goal is to minimize the number of cache misses (equivalent to the number of I/Os), while ensuring that the algorithm is also work optimal.

2.1.3 Hardware Power Modeling

As current platforms do not provide fine-grained power measurement capabilities, power models are the first step to enabling dynamic power management for power proportionality on all levels of a system. Currently, the approach closest to hardware-based monitoring is the running average power limit (RAPL) feature available for the Intel Sandy Bridge and Ivy Bridge CPUs [25], which allows for monitoring the power consumption of the whole CPU package.

As this feature is not available on other CPUs, power models typically rely on a number of performance counters. For instance work in [26] leverages 5 counters, including the instructions per cycle (IPC) counter, and rely on a regression model for estimation. Similar work has been performed by Counters in [27] additionally considers different CPU frequencies, but not multi-core architectures. Other work starts with all available counters and then try to reduce their number [28] by analyzing the correlation between counters of different architectures and power dissipation. Usually the accuracy of the models is validated by comparing estimates with the measures of a power meter when running benchmarks in isolation [29]. Power modeling often considers learning techniques such as sampling [30] that assume the proportionality of system events to power consumption. Measurements of a hardware power meter are gathered and subsequently used, together with a set of normalized estimated values, in various regression models, which are so far mostly linear [31]. However, in [32, 33], it is stated that linear power models depending on the CPU load are not sufficient anymore and that more parameters have to be considered. Study in [31] shows that, especially in multi-core systems, linear models lead to a much higher mean relative error for CPU power consumption and cannot easily be improved by applying more complex techniques. Linear models rely on the independence of the covered features, which is not realistic in current systems. Polynomial/exponential regression can cover these dependencies and, as shown in [34], a quadratic solution better fits the power modeling of multi-core systems. The described systems must however isolate processor features, such as HyperThreading and TurboBoost, to avoid hidden states. HAPPY [35] introduces a hyperthread-aware power model that differentiates between the cases where either single or both hardware threads of a core are in use. The most recent work in this line is BitWatts [36], which introduces a counter based power model for each individual frequency.

2.1.4 VM Power Modeling

In data centers, the efficiency of VM consolidation, power dependent cost modeling, and power provisioning are highly dependent on accurate power models. Such models are particularly needed because it is not possible to attach a power meter to a virtual machine. In general, VMs can be monitored as black-box systems for coarse-grained scheduling decisions. However, for fine-grained scheduling decisions—e.g., with heterogeneous hardware— finer-grained estimation at sub-system level is required and might even need to step inside the VM. So far, fine-grained power estimation of VMs required profiling each application separately. To exemplify, WattApp [21], which relies on application throughput instead of performance counters as a basis for the power model. PMapper [37] maps resources using a centralized step-wise decision algorithm in lieu of application power estimation.

To generalize power estimation, some systems like JouleMeter [32] assume that each VM only hosts a single application and thus treat VMs as black boxes. In a multi-VM system, they try to compute the resource usage of each VM in isolation and feed the resulting values in a power model. Bertran et al. [7] propose an approach uses a sampling phase to gather data related to performance-monitoring counters (PMCs) and compute energy models from these samples. With the gathered energy models, it is possible to predict the power consumption of a process, and therefore apply it to estimate the power consumption of the entire VM. Another example is VMeter [5], which estimates the consumption of all active VMs on a system. A linear model is used to compute the VMs' power consumption with the help of available statistics (processor utilization and I/O accesses) from each physical node. The total power consumption is subsequently computed by summing the VMs' consumption with the power consumed by the infrastructure. Janacek et al. [38] exploit a linear power model to compute the server consumption with postmortem analysis. The computed power consumption is then mapped to VMs depending on their load. This technique is not effective when runtime information is required.

As afformentioned, energy consumption of the host per job embraces the static power consumption, independent of the resource utilization, and the dynamic power, which is degraded not only proportional to the VM's allocated resources but also on account of the overhead caused in the hypervisor, and the interference due to collocation. Estimating this overhead is complicated since the pattern of the hypervisor overhead is tightly coupled with the number of VMs, the type of resources each VM asks for, and the number of times the switching occurs between VMs and hypervisor. Thus, for a more accurate estimation, further to individual VM's energy, VM interference energy overhead should also be estimated. Some estimation methods have been proposed in the state of the art: e.g. [5, 39, 40]. In [41] the authors argue that, in virtualized environments, energy monitoring has to be integrated within the VM as well as the hypervisor. They assume that each device driver is able to expose the power consumption of the corresponding device as well as an energy-aware guest operating system and is limited to integer applications.

Work in [42] introduces an interference coefficient, defined to model the energy interference. The major contribution of this work is to estimate the energy interference according to the previous knowledge of standalone application running on the same machine. They model interference as a separate implicit task. Moreover, an energy efficient collocation management policy is introduced in this work that is modeled as an optimization problem solvable by data mining techniques. All the VMs running on the same machine are known as a collection. The energy consumption of a collection is the sum of idle energy consumed for the longest VM run, dynamic energy consumed by each VM if they were run in isolated environment, and the energy depleted due to interference between each VM pair. The interference energy can be positive or negative depending on the intersection of resources between each VM pair. Interference energy is estimated as the coefficient of the summation of idle and isolated run for each VM. On the other hand, performance is measured as the delay, which is measured by modeling the system as a M/M/1 queue and calculating the imaginary interference tasks response time as the delay due to interference.

In data centers, the efficiency of VM consolidation, power dependent cost modeling, and power provisioning are highly dependent on accurate power models. Such models are particularly needed because it is not possible to attach a power meter to a virtual machine. In general, VMs can be monitored as black-box systems for coarse-grained scheduling decisions. However, for fine-grained scheduling decisions—e.g., with heterogeneous hardware— finer-grained estimation at sub-system level is required and might even need to step inside the VM. So far, fine-grained power estimation of VMs required profiling each application separately. To exemplify, WattApp [21], which relies on application throughput instead of performance counters as a basis for the power model. PMapper [37] maps resources using a centralized step-wise decision algorithm in lieu of application power estimation.

To generalize power estimation, some systems like JouleMeter [32] assume that each VM only hosts a single application and thus treat VMs as black boxes. In a multi-VM system, they try to compute the resource usage of each VM in isolation and feed the resulting values in a power model. Bertran et al. [7] propose an approach uses a sampling phase to gather data related to performance-monitoring counters (PMCs) and compute energy models from these samples. With the gathered energy models, it is possible to predict the power consumption of a process, and therefore apply it to estimate the power consumption of the entire VM. Another example is VMeter [5], which estimates the consumption of all active VMs on a system. A linear model is used to compute the VMs' power consumption with the help of available statistics (processor utilization and I/O accesses) from each physical node. The total power consumption is subsequently computed by summing the VMs' consumption with the power consumed by the infrastructure. Janacek et al. [38] exploit a linear power model to compute the server consumption with postmortem analysis. The computed power consumption is then mapped to VMs depending on their load. This technique is not effective when runtime information is required.

As afformentioned, energy consumption of the host per job embraces the static power consumption, independent of the resource utilization, and the dynamic power, which is degraded not only proportional to the VM's allocated resources but also on account of the overhead caused in the hypervisor, and the interference due to collocation. Estimating this overhead is complicated since the pattern of the hypervisor overhead is tightly coupled with the number of VMs, the type of resources each VM asks for, and the number of times the switching occurs between VMs and hypervisor. Thus, for a more accurate estimation, further to individual VM's energy, VM interference energy overhead should also be estimated. Some estimation methods have been proposed in the state of the art: e.g. [5, 39, 40]. In [41] the authors argue that, in virtualized environments, energy monitoring has to be integrated within the VM as well as the hypervisor. They assume that each device driver is able to expose the power consumption of the corresponding device as well as an energy-aware guest operating system and is limited to integer applications.

Work in [42] introduces an interference coefficient, defined to model the energy interference. The major contribution of this work is to estimate the energy interference according to the previous knowledge of standalone application running on the same machine. They model interference as a separate implicit task. Moreover, an energy efficient collocation management policy is introduced in this work that is modeled as an optimization problem solvable by data mining techniques. All the VMs running on the same machine are known as a collection. The energy consumption of a collection is the sum of idle energy consumed for the longest VM run, dynamic energy consumed by each VM if they were run in isolated environment, and the energy depleted due to interference between each VM pair. The interference energy can be positive or negative depending on the intersection of resources between each VM pair. Interference energy is estimated as the coefficient of the summation of idle and isolated run for each VM. On the other hand, performance is measured as the delay, which is measured by modeling the system as a M/M/1 queue and calculating the imaginary interference tasks response time as the delay due to interference.

2.1.5 Application Power Characterization

To the best of our knowledge, so far, there is little effort on application energy characterization. A line of work is toward profiling applications to figure out the energy consumption pattern of a particular application. In [43], a counter based application resource usage profiling which is followed by a mechanism to map it to energy consumption is proposed. In [44], a fine grained application energy profiling is proposed to enable application developers to make energy efficient choices. The most recent work in this line is [45], which compares two well-known application profiling tools, SLURM[1] and Score-P[2] available in Linux.

Tangential to this goal, recently a line of work is attempting to profile the application energy consumption for mobile devices [46, 47]. They try to characterize the diverse resources of mobile devices such as GPS, WiFi, CPU, Memory and storage requirements of individual mobile application. However, all the proposed techniques are measurement based, while we need a model based on the principals for application to fulfill the hardware agnostic requirement at the application layer of the stack.

In [23] a new complexity model is introduced to account for the energy used by an algorithm. Based on an abstract memory model (which was inspired by the popular DDR3 memory model), they present a simple energy model that is a (weighted) sum of the time complexity of the algorithm and the number of 'parallel' I/O accesses made by the algorithm. They derive this simple model from a more complicated model that better models the ground truth and present some experimental justification for their model. The simplicity and applicability of this energy model is the main contribution of the

[1] https://computing.llnl.gov/linux/slurm/slurm.html
[2] http://www.vi-hps.org/projects/score-p/

work. In their next work [48], they experimentally validate the algorithm energy complexity model derived. This energy complexity model is asymptotic which is expected in a hardware agnostic conceptualization.

2.2 Energy Effectiveness

As afformentioned, if the service is not delivered as expected, it may tarnish provider's reputation. Thus, it is required to obtain a service with desirable response time as well as acceptable throughput, availability and consistency level. Attaining high QoS may impose more energy consumption. Therefore, we should strive to alleviate the burden of high service energy. To this end, the energy efficiency is introduced in [49] as $\frac{Performance}{Energy}$. However, in this metric, there is no mechanism to guarantee the performance, and all sensitive and non-sensitive services are treated equally. In this definition, there is no mechanism to control the performance. For instance if we over provision the performance, for a particular service, we probably have to spend more energy, while gaining nothing in exchange of increased performance, since it is not sensible by the user. However, energy efficiency ratio may increase in this scenario. In the next section, we introduce energy efficiency metric to surmount the enumerated issues.

Increasingly, the most efficient servers nowadays, consume at least 20-30 percent of their nominal power in the idle case, and deviate from linear proportionality property noticeably according to the SPECPower_ssj2008. Hence, Idle to Peak Ratio (IPR) and Linear Deviation Ratio (LDR), for the current power model, are still remarkably higher than the ideal case. Higher IPR encourages the server consolidation for the sake of power saving; however, this is not always a solution. Utilizing a server to its 100% capacity may affect the applications performance tremendously, thus reducing actual energy efficiency of jobs, and also does not contribute to power saving in cases that LDR is unequal to one and when the interference overhead exceeds the proportion of static power.

Moreover, collocation of applications has its own challenges. Workload intensity is often highly dynamic. The power profile of the datacenter hardware is inherently heterogeneous; this makes the optimal performance gain problem more complicated. The nonlinearity and in some cases unpredictability of the energy efficiency profile currently aggravates the complexity of energy efficient collocation management.

Therefore, a performance aware energy analysis metric is required for a fair analysis of systems and techniques. 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 (2.1), is a speculative metric that quantifies the degree to which the ecosystem is successful in decreasing energy dissipated while the performance is not significantly violated.

$$\mathfrak{E} = \alpha \times \frac{E^*}{\hat{E}} + (1-\alpha).min(1, \frac{\hat{\mathcal{P}}}{\mathcal{P}^*}) \tag{2.1}$$

In (2.1), $\mathfrak{E}$ introduces the energy effectiveness, $\hat{E}$ and $\hat{\mathcal{P}}$ stand for the estimated or measured energy consumption and performance of the considered service on the running platform. E^* factorizes to the energy consumed to provide the service in an energy proportional system with a linear power model, representing the equality of utilization and associated power dissipation ($P(U) = U$), this is the minimum reachable energy consumption. $\mathcal{P}^*$ is quantified based on the Service Level Objectives(SLO) and Service Level Agreement(SLA) parameters depending on the interpretation of the performance on each layer of the service stack. Quantifying the SLA metrics is extensively studied in [50]. In other words, $\mathcal{P}^*$ represents the desirable performance conceptualized in the associated service stack layer.

Moreover, it is necessary to handle the trade off among these tightly coupled parameters to achieve an efficient mechanism. Intuitively, an adaptive model, covering the system and user requirements, is appropriate for this purpose, because the parameters are tunable in such model. The model supports 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.

2.2.1 Vulnerability Factor

Further to energy effectiveness, we define **Vulnerability Factor**, $\mathcal{V}$, which embodies to the range of variability in the energy effectiveness as $\mathcal{V} = \frac{\partial \mathfrak{E}}{\partial \alpha}$. Namely, $\mathcal{V}$ represents the slope of the $\mathfrak{E}$ equation when α varies in range of 0-1. The higher the $\mathcal{V}$, the more influence adaptiveness factor has in $\mathfrak{E}$ value, and the mare important is to set it properly. $\mathcal{V}$ can be determined in the SLAs according to the user incentives (previously addressed for cycle-sharing [51]) and service requirements (previously addressed for virtual machines [52] and Java applications [53]).

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 in case of 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 [23] to find out more.

2.3 Hardware Power Model

Power is majorly drawn in communication and processing hardware, during the service provisioning life-cycle. In this section, we study different approaches toward power modeling across the service stack.

Linear Power Model 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.

$$P_{host} = P_s + (P_{Max} - P_s) \times U \tag{2.2}$$

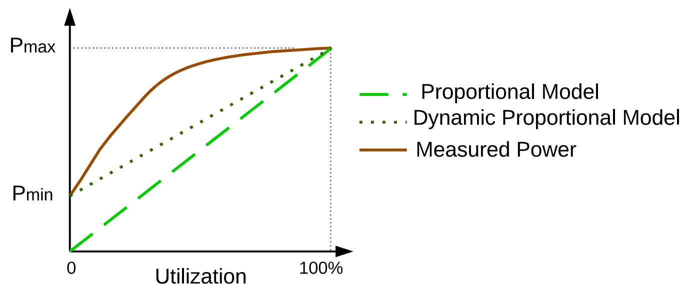


FIGURE 2.2: Energy Proportionality

In (2.2), a linear correlation among the utilization level and the power drain is assumed, 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 the next section. ers in a data center is above 100 watts.

2.3.1 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 (2.3).

$$LDR = max\frac{P(U) - (P_s + P_d)}{P_s + P_d} \tag{2.3}$$

IPR is the indicator of idle to peak power consumption as illustrated in (2.4).

$$IPR = \frac{P_{idle}}{P_{Max}} \tag{2.4}$$

To measure how far a system power model is from the ideal (energy proportional) one, Proportionality Gap(PG) [54] 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 (2.5). Therefore, having proportionality gap values for a given device, we can reconstruct the power model of the device.

$$PG(U) = \frac{P(U) - (P_{Max} \times U)}{P_{Max}} \tag{2.5}$$

Given the state of the art hardware, designing hardware which is fully energy proportional remains an open challenge, power model of a non-energy proportional system is illustrated in Figure 2.2. However, even in the absence of redesigned hardware, we can approximate the behavior of energy proportional systems by leveraging combined power saving mechanisms [55] and engaging heterogeneous commodity devices combined with powerful server machines in lieu of homogeneous server hardware platform [54].

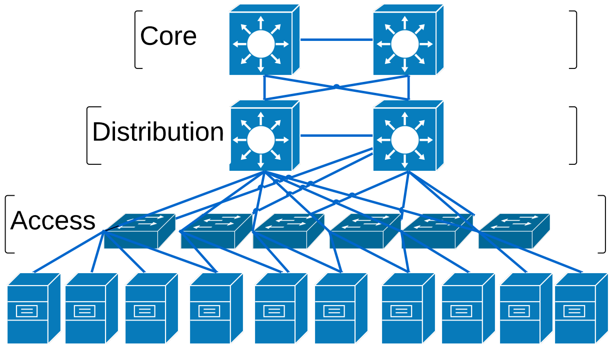


FIGURE 2.3: Hierarchical three-tier data center topolgy

2.3.2 Communication Power Model

Data center communication power: 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-tire, hierarchical topology. The motivation behind formulating hierarchical model is that it can be easily generalized to numerous intra-data center topologies, e.g. Fat-Tree[56] , VL2[57], BCube[58], PCube[59], etc. The tree depth is defined based on the path messages should traverse within the data center in each layer. For the topologies which deviate from this property, e.g. CamCube[60], 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 2.3.

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 (2.6). P_{switch} stands for power drawn by the switch. Additionally, we added P_{host} to each level consumption to generalize our model.

$$P^{intra}_{DC\_comm} = \sum_{i=1}^{l-1} (P_{switch}(i) + P_{host}(i)) \tag{2.6}$$

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 (2.6), 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 more 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 [61] and FlatNet [62], obtain shorter path via less switches. Topologies providing smaller network diameter are also more energy efficient due to shorter average path should 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, since 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 should be traversed to reach a host

P2P-cloud communication power modeling: 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[3], with more than $20,000$ active nodes, Athens Wireless Metropolitan Network[4], FunkFeuer[5], Freifunk[6], 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 [63].

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.

We characterize the power consumption in the P2P-cloud by means of experimental measurements in a production wireless community network. The network consists of around 50 802.11an-based nodes. It is deployed as part of the *Quick Mesh Project* (QMP)[7] and EU CONFINE project[8]. We shall refer to this network as *QMPSU*, which is part of a larger Community Network having more than 20.000 operative nodes called Guifi.net[9]. An experimental evaluation of QMPSU can be found in [64], and a monitoring page is available in Interne[10].

Typically, QMPSU users have an outdoor router with a wifi 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 premises network as depicted in Figure 2.4.

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

$$P_{WN} = 2\,P_{\text{AP}} + 4.78\,P_{\text{router}}, \tag{2.7}$$

[3] http://guifi.net/en
[4] http://www.awmn.net
[5] http://www.funkfeuer.at
[6] http://freifunk.net
[7] http://qmp.cat
[8] http://confine-project.eu/
[9] http://guifi.net/en
[10] http://dsg.ac.upc.edu/qmpsu

FIGURE 2.4: QMPSU connectivity

where P_{AP} and P_{router} are the power consumption of the AP and outdoor routers, respectively.

The most common outdoor router used in QMPSU is the Ubiquiti NanoStation M5 (NS)[11]. As indoor AP we have considered the TP-LINK WDR4300 [12].

Internet Power consumption: P2P-clouds for inter-vicinity communication and classic data centers 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. [65] 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 (2.8). In this model, $P_{Internet}$ stands for Internet power consumption which is a combination of power drawn in each level $L = \{core, distribution, access\}$. P_*^{router} denotes router power consumption in layer $*$, and n_*^{hops} indicates the number of hops should be traversed at $*$ layer.

$$P_{Internet} = \frac{1}{\varphi} \times \sum_{* \in L} P_* \times n_*^{hops} \tag{2.8}$$

The concept of oversubscription, φ, exist 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. Oversubscription for the home users is 40:1 and for the business connection is around 20:1 in the current Internet.

2.4 VM Power Estimation

As afformentioned, direct VM power measurement is not possible, therefore, VM power modeling is essential to estimate VM power consumption. Models for power estimation

[11] http://www.ubnt.com/downloads/datasheets/nanostationm/nsm_ds_web.pdf
[12] http://www.tp-link.com/lk/products/details/?model=TL-WDR4300

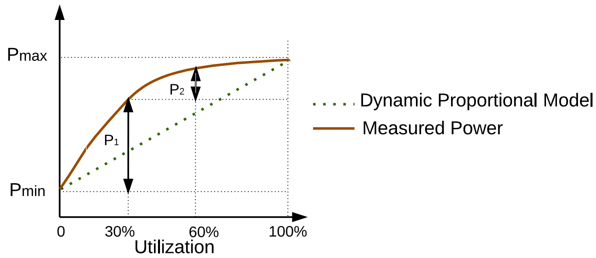


FIGURE 2.5: VM power modeling issues in non-energy proportional systems

have been majorly studied at the level of processors, and less extensively in the context of virtualization.

Besides the hypervisor and interference overhead in multi-tenant systems, the non-energy proportional hardware adds more complexity to the VM power modeling agenda. In non-energy proportional hardware platform, since the hardware power model is non-linear, two identical VMs, sharing the same hardware, may end up with different dynamic power usage estimation during the runtime, which may lead to unfair energy based service charging, and planning. Figure 2.5, visualizes such a case. In this scenario, there are two identical VMs, i.e. VM_1 and VM_2, collocated on a host with the power model demonstrated in the Figure. If we only run VM_1, the dynamic power estimated for this VM will be P_1, whereas running the second identical VM on the same machine predicted as $P_2 < P_1$. Therefore, in case of collocation, there should be a strategy to divide the dynamic power fairly among the running VMs.

Proposed Solution: To address the fairness issue introduced in the previous section we propose the weighted division VM power model. In this model as illustrated in (2.9), a particular VM's power consumption, $P_{VM}(i)$ is calculated according to the relative utilization, i.e. $\frac{u_i}{U}$, contributed by that particular VM. In this equation, u_i represents the utilization incurred by VM i, and U denotes the overall machine utilization.

$$P_{VM}(i) = \frac{u_i P(U)}{U} \tag{2.9}$$

2.5 Application Power Modelling

Application energy characterization faces more challenges compare to the VM and hardware challenges. Application Energy model should be accurate enough in a course grained view toward energy characterization and increasingly needs to be hardware oblivious. Fulfilling these requirements needs to sketch a model that attributes the application requirements to a set of parameters that represent the tentative resource utilization in run time.

Typical state of the art approach as mentioned in related work, is application profiling which fails to meet hardware agnosticness.

The closest work to the approach to our proposal, i.e. analytical model for application power characterization is [23]. However, this study is centered on the algorithms and is only studied for a limited set of algorithms. Therefore, a generalized model derived from

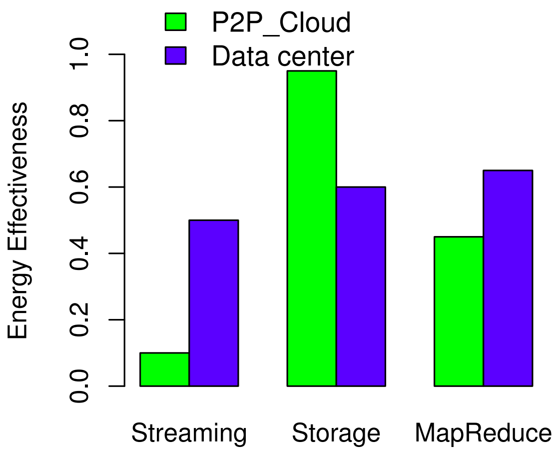


FIGURE 2.6: Energy effectiveness of typical cloud applications

the proposed model is required to build up a framework for algorithm energy complexity. Besides, application energy analysis in a hardware oblivious setting needs to take into account data flow among the algorithms as the basic components of an application. Note that in distributed application settings such as MapReduce, flow complexity broadens its extends to network communication.

2.6 Evaluation

In this section we study the energy effectiveness introduced in this chapter for a set of typical cloud applications, i.e. storage, streaming and MapReduce. However, this experiment is done in a particular hardware setting, and cannot be generalized in the higher layers of the stack, i.e. application and virtualization layer. Note that in this study we set $\alpha = 0.5$, since the aim of this experiment is not addressing the adaptiveness. Effect of α and vulnerability factor are elaborated in the next chapter.

Figure 2.6 illustrates the energy effectiveness of different application in P2P-cloud and classic data center systems. As shown in the figure, identical hardware setting results in different values for energy effectiveness of each application, due to the particular characteristics of each application, which includes the minimum amount of energy it uses as well as the performance requirements of the application. More interestingly, we see the difference in energy effectiveness of the same application in different hardware settings. As shown in Figure 2.6, storage service has higher energy effectiveness when is run on P2P-cloud, while streaming and Mapreduce perform better on data centers, since they are more process intensive, and P2P platform cannot provide powerful processing capability.

2.7 Summary

In this chapter we addressed the formulation of a performance aware energy analysis metric by introducing energy effectiveness, which can be specified in each layer of service provisioning stack, i.e. application, virtual machine/OS and hardware, from a course-grained, asymptotic, hardware agnostic conceptualization on top of the stack to an accurate, fine-grained, hardware dependent formulation on the bottom layer. We introduced power modeling in communication and process elements of different cloud platforms, and discussed the added complexity in power modeling rooted in multi-tenancy as the cornerstone of cloud service provisioning. Besides, we characterized the performance metric in each layer of the stack to enable energy effectiveness calculation with different granularity across the stack.

In the next section we form a service analysis framework, leveraging the energy effectiveness metric we introduced here, to improve energy effectiveness by efficient service platform selection.

Chapter 3

Analysis Framework

P2P-clouds embrace vast sums of ubiquitous commodity ICT resources, which introduce an opportunity to scale the cloud service provisioning beyond the borders of giant cloud service providers such as Amazon that rely on gigantic data centers. However, since energy consumption is becoming crucial in industrial world including IT sector, emerging technologies should be energy efficient enough to be able to survive in the new economics paradigm. To understand if P2P-cloud as an emerging cloud paradigm meets the above condition, in this chapter , we compare the energy dissipation of P2P-cloud and data centers through an analytical model and assess it in a particular setting.

Thus, to sketch a comprehensive view of energy consumption within a service life cycle, we need a hardware agnostic framework to cope with the hardware diversity. This framework can be customized to any hardware platform to outline the energy consumption in that particular setting. Leveraging such a framework assists the resource management module and broker to make energy aware decisions for resource allocation in the federated environment of P2P-clouds and data centers.

3.1 Related Work

Previous work [66, P2, P5] reveals that, in the contest between classic data centers and P2P-clouds, the latter can compete with the classic datacenter model in terms of energy efficiency for specific services, as long as the jobs are served mostly locally. Nonetheless, there is no straightforward global answer for this question, since energy consumption depends on a diverse range of factors on service provisioning stack, from hardware specifications to the service characteristics and execution platform.

To the best of our knowledge, there is limited work addresses the energy consumption analysis in P2P platforms. In [66] a high level model of P2P and data center energy consumption is introduced, and [67] compared streaming service in nano-data centers with gigantic ones in terms of energy consumption. In this chapter we introduce an analytical framework to characterize service energy consumption in a P2P assisted cloud platform.

3.2 Service Energy Analysis

In this section we analyse the energy effectiveness across a range of cloud services falling in different categories of communication, process and storage intensive services.

3.2.1 Storage as a Service

Hadoop File System(HDFS) [68], as the most prevalent distributed file system leveraged in data centers, is a block based file system that divides a file by default into blocks of 64MB. These blocks are stored across the cluster of one or several machines which are referred to as DataNodes. To store the file blocks, HDFS chooses the target DataNodes randomly for each block. Thus, retrieving a file may require cooperation of multiple machines. NameNode is the machine that facilitates this coordination by storing all metadata for the file system. To open a file, a client contacts the NameNode and retrieves a list of locations for the blocks that comprise the file to identify the DataNodes holding each block. Client then directly reads the files from the DataNodes, possibly in parallel. To make the data robust to failure and increase the HDFS reliability, data is replicated in different DataNodes, by default with replication factor of 3. However, NameNode is still a single point of failure in HDFS.

On the other hand, in P2P-cloud instance we have, i.e. cloud on top of the CONFINE community network [69], storage service is provided via Tahoe-LAFS decentralized storage system. Akin to HDFS, Tahoe-LAFS cluster is embodied to client nodes, storage nodes and an introducer as the single coordinator node which plays the same role as the NameNode in HDFS. Storage nodes announce their presence to the introducer; therefore, when a client node intends to store data, it connects to the introducer to get the list of present storage nodes. When the client uploads a file to the storage cluster, a unique public/private key pair is generated for that file, and the file is encrypted, erasure coded and distributed across storage nodes (with enough storage space) [69]. The erasure coding parameters determine how many servers are used to store each file which is denoted as N, and how many of them are necessary for the files to be available, K . The default parameters in Tahoe-LAFS are K=3 and N=10 (3-of-10). The location of erasure coded shares is decided by a server selection algorithm that hashes the private key of the file in order to generate a distinct server permutation. To download a file, the client asks all known storage nodes to list the pieces of that particular file if they hold any, then client chooses which nodes to request for each piece based on various metrics such as latency, node load, etc.

Therefore, generally, to offer storage service on a distributed system we need a decentralized storage system installed on top of the infrastructure. A decentralized storage system embodies to a set of storage nodes, client nodes and coordinators. Storage nodes are coordinated by the coordinator nodes which are aware of each individual node, e.g. NameNode and Introducer in HDFS and Tahoe-LAFS.

Energy consumption of storage service factorizes to the communication, coordination and storage nodes energy dissipation.

$$E_{SaaS} = r \times (E_{communication} + E_{coordinator} + E_{storage}) \tag{3.1}$$

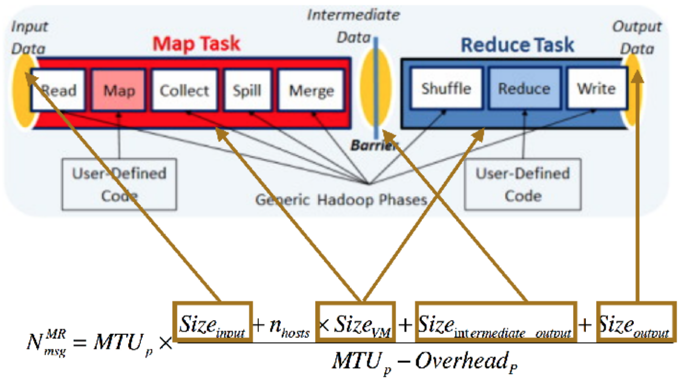


FIGURE 3.1: Mapreduce data flow

Communication energy,$E_{communication}$ is modeled in Chapter 2, and $E_{coordination}$ denotes the energy consumed in the coordinator host, following the host power model formulated in the previous chapter. r represents the replication factor which is by default set to 3 in HDFS.

$E_{storage}$ depends on the drive technology, for the SSD drives this value is only proportional to the data size and is trivial compare to the other parts energy, while for the HDD drives based 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. There is some effort to model disk power usage on HDD technology [70] as $E_{storage} = \frac{P_{HDD}}{\tau_{HDD}}$.

3.2.2 MapReduce as a Service

To scrutinize the energy consumed in the clouds, we analyze the energy consumed per MapReduce job, both in the datacenter and P2P-clouds, as visualized in Figure 3.1. When a MapReduce request is sent to a data center, 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 data center communication infrastructure includes VM and input data with size, $Size_{input}$. 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 size of data to be transmitted is following (3.2).

$$Size_{data}^{MR} = Size_{input} + n_{hosts} \times Size_{VM} + \sum_{i=1}^{n_t} Size_t(i) + r \times Size_{intermediate\_output} + Size_{output} \tag{3.2}$$

$Size_{VM}$ 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.

The energy consumed to transmit the required data for a job, as shown in (3.3) is the multiplication of power drawn for the communication, the amount of data should be transmitted, as depicted in (3.2), over the network throughput, τ_{DC}.

$$E^{MR}_{intra\_DC\_comm} = P^{intra\_DC}_{DC\_comm} \times \frac{Size^{MR}_{data}}{\tau_{DC}} \tag{3.3}$$

Network throughput is a factor of network infrastructure and communication protocol. Exploiting Gigabit Ethernet, 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 in which Map, collect, split, and Reduce are basically 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 [71], 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 (2.8). 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 following (3.4). Here, PUE defines Power Usage Efficiency of the data center.

$$E^{MR}_{intra\_DC} = PUE \times [E^{MR}_{intra\_DC\_comm} + \sum_{i=1}^{numberofphases} (E[t_{task}] \times \sum_{n_t} P_{host})] \tag{3.4}$$

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 scenario, 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 are needed only to be sent to a host via 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 in QMPSU. In this case the amount of data to transmit over the community network follows (3.2). Overall, the energy required to accomplish a MapReduce job on community for the intra-vicinity mode is given in (3.5). t_{P2P} implies the response time of the hosts in P2P-cloud.

$$E^{MR}_{intra\_P2P} = P^{comm}_{WN} \frac{Size^{MR}_{data}}{\tau_{intra\_P2P}} + \sum_{i=1}^{numberofphases} (E[t_{task}] \times \sum_{n_t} P_{host}) \tag{3.5}$$

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 dissipated due to the utilization of resources induced by the task.

3.2.3 Streaming as a Service

Video streaming service can provide either online streaming, i.e. content being encoded on the fly, or offline streaming, i.e. serving previously encoded and stored content. 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 P2P-cloud and data center models, since the data should be retrieved from network accessible storage. Video decoding, on the other hand, is always done in the end user side. Hence, applying power aware video decoding mechanisms [72] contributes to a more energy efficient service provisioning at end user level.

Nonetheless, in cloud assisted live video streaming, e.g. Amazon CloudFront live video streaming[1], video rendering is done in the data center servers. For instance, CloudFront uses the Adobe Flash Media live encoder.

For offline video streaming, as in (3.6), the energy is dissipated in three parts, retrieving the frames form the NAS, $E_{storage} + E^{intra-DC}_{transmit}$,and transmitting them over Internet, $E^{Internet}_{transmit}$. On the user side, this frames should be buffered and decoded to play on the screen. Since in all scenarios we target the same end user, we presume that the end user energy consumption is a constant amount for all given scenarios, E_u.

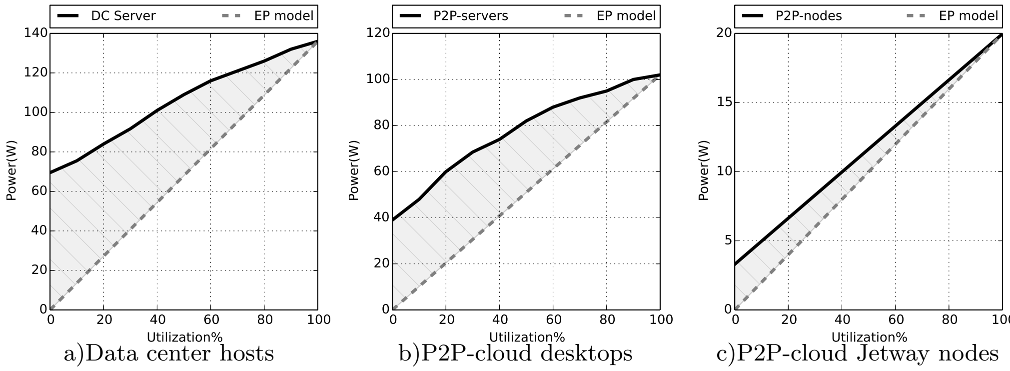
$$E^{OD}_{streaming} = E_{storage} + E^{intra-DC}_{transmit} + E^{Internet}_{transmit} + E_u \tag{3.6}$$

In (3.6), $E_{storage}$ refers to energy dissipated for reads and writes from/to disk, which is explained in Section 3.2.1.

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 both intra data center, Internet communication and P2P transmission are the same as 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 should model the video encoding energy consumption $E_{encoding}$ and replace it with the $E_{storage}$ in (3.6). For the rest of the processes we can follow the offline streaming model. Video coding tightly couples with the video format; nevertheless, we can assume that encoding energy is larger or equal to decoding energy in a particular hardware platform due to the exhaustive, extra stage of complicated motion compensation recognition process should be traversed in encoding process. Here we consider H264 video format. H264 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 compare to the I and P frame on the same machine, since B frame relies on bi-directional differential coding of the values through the JPEG coding process, $E^{I}_{encoding} \leq E^{P}_{encoding} \leq E^{B}_{encoding}$.

[1] http://docs.aws.amazon.com/AmazonCloudFront/



a)Data center hosts b)P2P-cloud desktops c)P2P-cloud Jetway nodes

FIGURE 3.2: Host Power Model

3.3 Evaluation

We aim to analyse the energy consumption on different cloud models under the video streaming, storage and MapReduce workload with the following configuration.

3.3.1 Experiment Setup

Processing power: For data center, host power model is derived through loading as Sandy bridge server machine with different loads, collecting performance counters and applying regression, the corresponding power to utilization values are depicted in Figure 3.2.a.

For the P2P-cloud nodes we employ the Jetway JBC362F36W with Intel Atom N2600 CPU with the maximum power of 20W, whereas there are Dell Optiplex 7010 machines with Core i7 CPU, 4 cores supporting up to 8 parallel threads. We derive a power model for these devices, as shown in Figure 3.2.b,c by utilizing the machines in different levels, using Stress[2]. The portion of Dell machines are 20% in the testbed while we have 80% of Jetway devices.

Data center Switch Power Model: Different topologies covering switch centric and server centric have been studied and simulated using power consumption values of switches available in the market. For the core switch, we opt Cisco Nexus 5596T, which has 32 ports of 10 Gbit Ethernet, and supports optical networking owing to SPF+ ports. It typically dissipate 900 watts, while the maximum power is 1100 watts.

For the distribution and access layer switches we rely on Cisco Nexus 2232TM switch which has 32 ports of 1 and 10 Gbit Ethernet with the over subscription of 4:1. Its maximum power consumption is 386 watts; nonetheless, it draws 280-350 watts, typically. For commodity switches, we employ Cisco Catalyst 37590-48TS which consumes the maximum of 75 watts and provides 48 ports. For all the switches, since 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.

[2]http://linux.die.net/man/1/stress

TABLE 3.1: Wireless Infrastructure Power Consumption

Power(watts)	**NS**	**TP-LINK**
Static	3.7	3.9
UDP-Max	5.0	5.4
TCP-Max	5.2	6.1

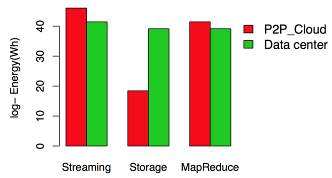


FIGURE 3.3: Service energy consumption in different platforms

The power consumption of a server's port is set to be 3W [73]. Internet energy consumption values are derived from [65], which characterizes the metro power of around 10.25 watts and core power of less than 0.15 watts per connection through fast Ethernet link with the over subscription of 40:1.

For MapReduce 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.

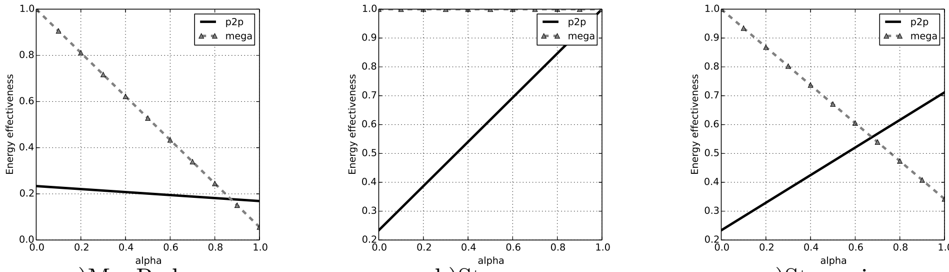
P2P Communication Power Modeling: We elaborated the power consumption of the P2P communication infrastructure described in 2.3. We have observed a similar power consumption in the TP-LINK and NS, which falls in the range of around 4 W (static power) to 6 W (maximum throughput), the measured values are given in Table 3.1. However, all devices are far from transmitting at the maximum throughput during a typical transmission, experimental measurements show an average throughput between nodes and their gateway of 10.9 Mbps (see [64]), which be estimated as the average throughput between any pair of nodes. Thus, we conclude that 5 W is a good rule of thumb as power consumption for all networking devices in QMPSU. Substituting in (2.7) we have that the average consumption in QMPSU is 33.9 watts.

Regarding the round trip time (RTT), experimental measurements in QMPSU give an average RTT of each node to the gateway of 18.3 ms, with standard deviation σ = 50.6 ms. We shall use these values as estimation for end-to-end RTT delays in the network.

3.3.2 Results and Discussion

Figure 3.3 compares the energy consumption of different services provided in the same hardware setting. As shown in this figure, different services perform better in different platforms, in terms of energy consumption. This result confirms that there is no straight forward answer for the most energy efficient service platform, P2P-cloud vs. data center.

Figure 3.4 certifies that if we focus on energy effectiveness, i.e. performance aware energy analysis metric, rather than pure energy consumption analysis, more complexity is added to the agenda by putting different values on performance.



a)MapReduce b)Storage c)Streaming

FIGURE 3.4: Energy effectiveness

TABLE 3.2: 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

Moreover, moving from hardware agnostic toward hardware aware analysis, particular hardware setting may result in opposite results in the comparison. In the next section, we study the MapReduce in different hardware setting and show that P2P-cloud can save more energy for this service, in contrast to what is shown in Figure 3.3.

3.3.3 MapReduce Case Study

In this section we study MapReduce service provisioning in clouds in more details in particular circumstances.

3.3.3.1 Experiment Setup and Scenarios

We 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 [74] 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[3]. 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 [75]; Internet energy consumption values are derived from [65]. Four VM types as shown in Table 3.2 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:

[3] https://www.spec.org/benchmarks.html

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.

3.3.3.2 P2P-cloud Energy Consumption

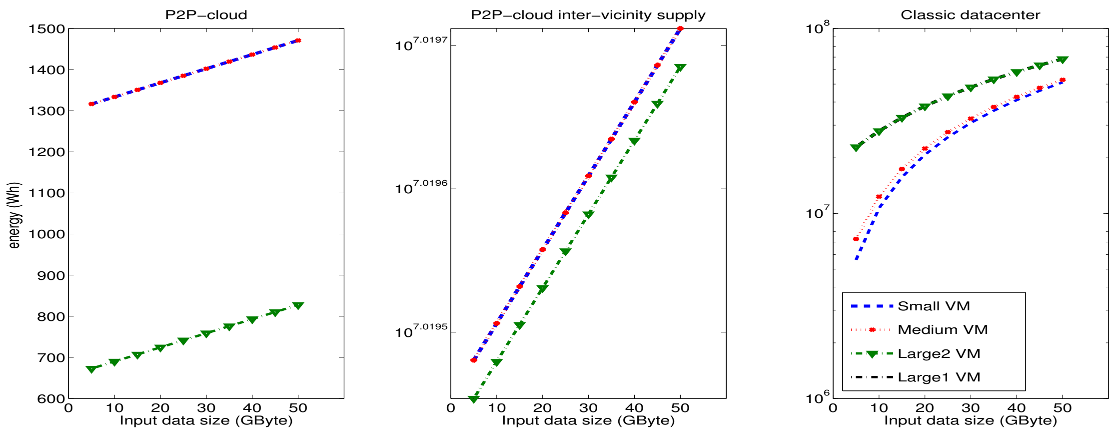


FIGURE 3.5: Energy consumption for various inputs across scenarios

In Figure 3.5, 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 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 3.5, 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

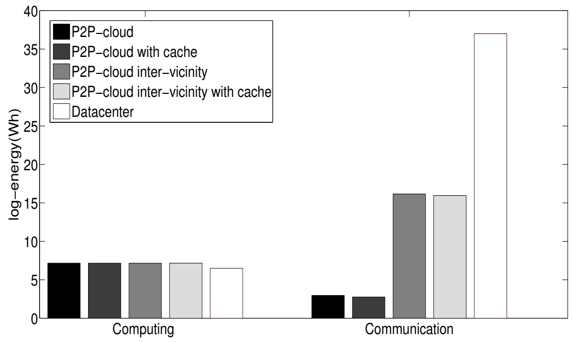


FIGURE 3.6: Compute vs. communication energy consumption

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 3.6 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.

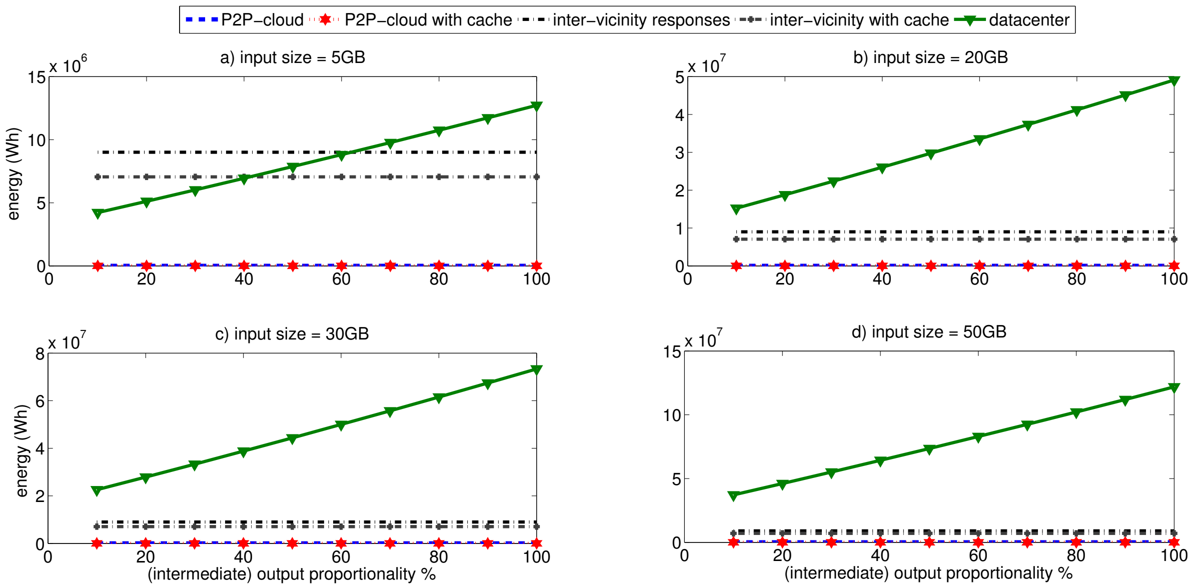


FIGURE 3.7: Energy consumption of applications with different input-output sizes running on small VMs

3.3.3.3 VM size effect

As shown in Table 3.2 we consider three different types of VMs with different capabilities of processing MapReduce tasks. Figure 3.5 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

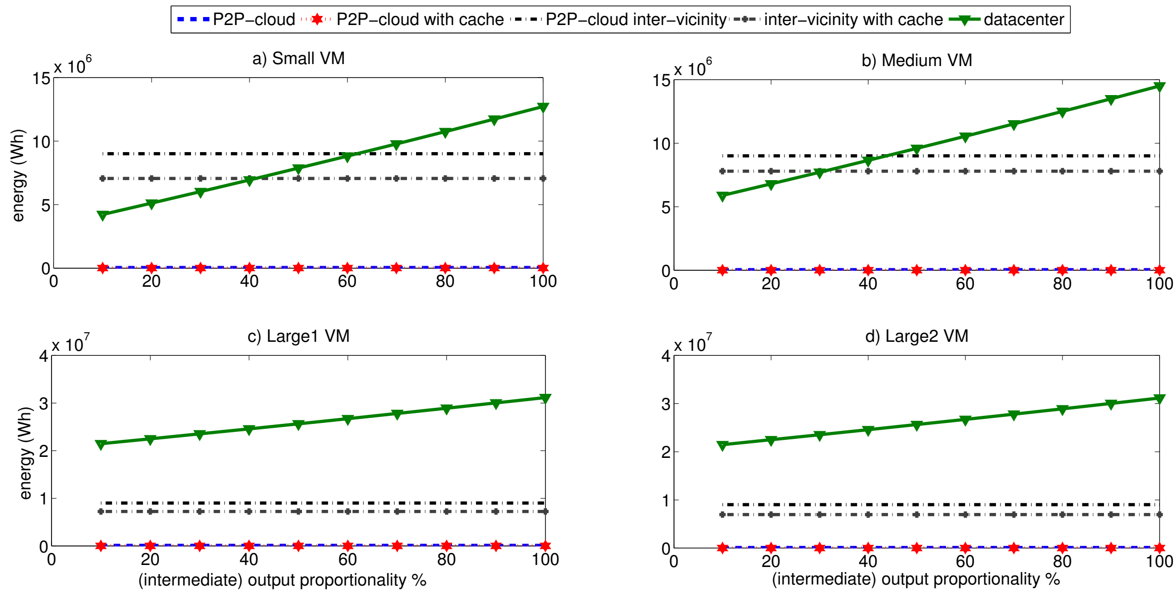


FIGURE 3.8: Energy consumption of a 5GB input application running on different VMs across scenarios

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 3.5 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.

3.3.3.4 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 3.7 illustrates the importance of VM selection for applications with smaller input and output sizes. As shown in Figure 3.7 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 3.7 focuses on small VM. To be more precise, we draw the energy consumption for small inputs across different scenarios including different VMs in Figure 3.8 because

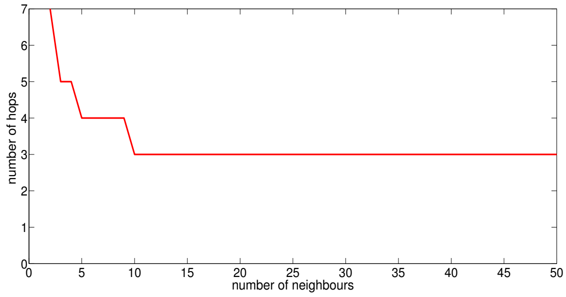
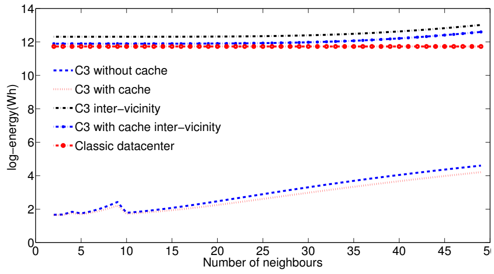
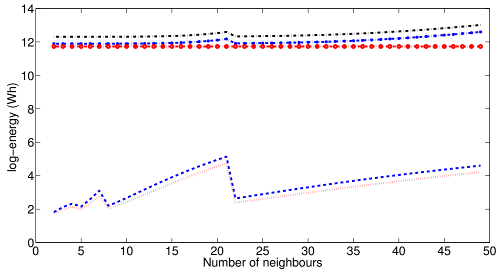


FIGURE 3.9: Impact of number of neighbours in vicinity diameter on average hops between two nodes.



a) Vicinity size = 100



b) Vicinity size = 500

FIGURE 3.10: Vicinity Density effect in community networks

the intermeidate-output has to be exchanged among vicinities in this case. As depicted in Figure 3.8, in small and medium VMs there is a cross point among datacenter energy consumption and inter-vicinity responding in P2P-cloud, Figure 3.8.a , Figure 3.8.b . However, for the large VMs, energy consumption of datacenter always exceeds the P2P-cloud scenarios even for the small input size, Figure 3.8.c , Figure 3.8.d.

3.3.3.5 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 3.9 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 3.10, 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 3.10 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

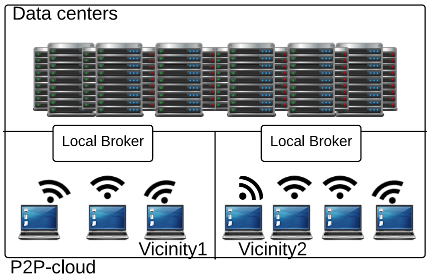


FIGURE 3.11: P2P-assisted Cloud Architecture

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.

3.4 Energy Aware Platform Selection

Generally, each service consumes energy on hosts and communication infrastructure. Host energy factors to CPU, memory, storage and other I/O devices. Hence, being able to analyse each service to understand the hot spots in resource consumption for a particular service, we can derive a conclusion about which platform suits better to that service.

Afterwards, defining local brokers equipped with an insight into the energy consumption pattern of each individual service, it can decide in an energy-aware service platform selection and resource allocation in the federated cloud architecture explained, as shown in Figure 3.11.

In algorithm 1, we outline how a broker can take into account the energy awareness in resource allocation process for each service. In this algorithm, the computing and communication energy effectiveness difference of the service in P2P and data center platforms are calculated, and service is allocated to the appropriate platform accordingly if enough resources are available.

Algorithm 1 service assignment policy

function SERVICE MANAGEMENT($service, \alpha$) $\triangleright$ α represents the effectiveness factor
    $\Delta E_{compute} \leftarrow N_{host}^{P2P} \times E_{host}^{P2P}(U) - N_{host}^{DC} \times E_{host}^{DC}(U)$
    $\Delta E_{communic} \leftarrow N_{hops}^{P2P} \times E_{communic}^{P2P} - N_{hops}^{DC} \times E_{communic}^{DC}$
    if $(\Delta E_{compute} + \Delta E_{communic} \leq 0)$ **then**
        if (P2P resource available and energy effectiveness(α)) **then**
            return Direct to P2P
        end if
    end if
    return Direct to data center
end function

3.5 Summary

In this chapter we introduced a hardware agnostic framework to analyse energy effectiveness both from hardware and service vantage point. Characterizing the energy model for a particular setting and exerting this framework, a broker can decide for a more energy effective service allocation, which considers energy conservativeness while still meets the quality of service requirements.

Moreover, we compared the energy consumption in classic datacenter model with P2P-clouds. We scrutinized the main sources of energy consumption in both systems and assessed their energy effectiveness.

We extended the P2P-cloud in intra-vicinity, previously presented in [P2], as a response to a more energy efficient solution to assist cloud ecosystems. The effect of exerting P2P-cloud on quality of service was studied on services for file transfer, video streaming, and MapReduce jobs. Our MapReduce case study indicates that the hardware specifications may completely turn the table in favor of a specific platform which revokes the possibility of finding an straight forward answer to our major question in this research, i.e. "Is it energy efficient to switch to community cloud?", and highlights the necessity of a general framework to eork as a middleware between the service stack layers. This middleware framework maps the hardware-agnostic, corse-grained service level model to a particular hardware setting, to refine the model and come up with an answer for our major question, afformentioned.

Chapter 4

Combining P2P Assisted Cloud and Smart Grid

As carbon footprint rate rises in recent years, and is predicted that the global carbon emission will reach 1430 megatonnes by 2020 [76]. being energy efficient and moving toward green energy sources are essential for environmental sustainability. Information and Communication Technology (ICT) plays a leading role in this context by its potential for providing a large scale real time controller to improve decision making and developing environmental information systems [77]. Moreover, investments in energy saving technologies are compensated financially, particularly when carbon tax is applied to energy price. However, this ICT infrastructure itself is a source of energy consumption. For instance, cloud computing energy consumption will increase to 1,963 billion kWh by 2020 and the associated CO2 equivalent emissions of 1,034 megatonnes will be expected [76].

Therefore, green ICT and ICT for green are not mutually exclusive, both are important and they complement each other [78]. Hence, the challenge for the future lies in synthesising, not only ICT for green, but also green ICT, to achieve a more sustainable service platform.

The electricity industry attempts to transform itself from a centralized, producer controlled network to a more consumer interactive and decentralized one via smart grid. Smart grid intends to achieve grid's full potential and prepares a cleaner and more efficient, reliable, resilient and responsive electric system. A smart grid system needs a large scale infrastructure for collecting and communicating data; likewise, it must have access to flexible, network-scattered computational power, network bandwidth, and storage capacity, due to distributed nature of data sources.

Akin to smart grid, ubiquitous P2P society is a collaborative effort in which infrastructure and services are shared among several individuals and/or organizations forming a specific community with common concerns. Ubiquitous society envisions a world in which services are accessible from anywhere, anytime, by anyone and anything[1]. These goals are partially intersected with the cloud vision, which introduces pervasive service provisioning. Therefore, we name the ubiquitous P2P society as P2P-cloud.

[1] http://www.itu.int/WORLD2006/forum/ubiquitous_network_society.html

Since Energy and ICT are two pillars of modern life that advance hand in hand, in line with the goals of ubiquitous society, in this chapter, we propose Cloud of Energy(CoE) system, which considers everything as a service (XaaS), as introduced in the idea of clouds [79], e.g. Infrastructure as a Service, Platform as a Service, Software as a Service. In tandem with this trend, Energy as a Service is added to the agenda in CoE. Smart grid and P2P-cloud are both large scale distributed systems involving vast sums of common specifications: self service, metered, elastic resources, multi-tenant, and access via the network are cases in point. Thus, CoE combines P2P-cloud, including sensors, commodity desktop machines and IoT boards, with the smart grid, to provide energy efficient services and also contributes to smart energy system's computing and communication platform.

There is a growing body of work centered on exploiting the cloud and peer to peer platforms for the smart grid computing [82, 84**?** , 85]. In a cloud computing environment, flexible data centers offer scalable computing, storage and network resources to any Internet-enabled device on demand. Moreover, P2P-cloud can manage the massive amount of data from distributed sources of consumption, generation and network nodes. On the other hand, diverse energy sources of smart grid improves the availability, sustainability and environment friendliness of the ubiquitous network society services.

The main contribution of this part is introducing CoE architecture as an integrated energy and computing platform, Section 4.2. CoE aims to design a service framework that incentivies all range of service producers, offering services from computing to energy, in range of small prosumers to giant providers, to serve in a greener marketplace, through an economic middleware, outlined in Section 4.2.3. We analyse the feasibility of the proposed architecture in Section 4.3.

4.1 Background and Related Work

The electricity industry attempts to transform itself from a centralized, producer controlled network to a more consumer interactive and decentralized one via smart grid. Smart grid enables the industry's best ideas for grid modernization to achieve their full potential and prepares a cleaner and more efficient, reliable, resilient and responsive electric system.

A smart grid system requires a monitoring capability suitable for wide area deployments. It needs a large scale infrastructure for collecting and communicating data; likewise, it must have access to flexible (possibly network-scattered) computational power, network bandwidth, and storage capacity. The distributed nature of data sources, the possibility that data may need to be collected from multiple (competitive) power producing and transport enterprises, and this need for timely state estimation, all make the system more complicated.

Akin to smart grid, P2P-cloud is a collaborative effort in which infrastructure and services are shared among several organizations form a specific community with common concerns. Smart grid and P2P-cloud are both large scale distributed systems involving vast sums of common specifications: self service, metered resources, multi-tenant, elastic resources, and access via the network are cases in common.

The economic models invented for smart grid and P2P-cloud systems are inspiring for each other, on account of the similarities of these two systems. Primarily, because in both P2P-cloud and smart grid, consumers can be providers as well. Moreover, they both follow the pay-as-you-go mechanism and the computational tasks are not batched; hence, there is no waiting time. This enables a time-critical model of computation.

We can leverage the ad hoc and elastic nature of clouds to benefit the smart grid at the economy of scale expected from cloud computing, while efficiently utilizing power as we scale up. On the other hand, combining smart grid and community clouds in a symbiotic relationship can be mutually beneficial in fostering adoption of both.

Some previous work [80–85] sketches a smart grid communication and information platform that relies on a cloud system. To employ such solutions, the smart grid should establish its own cloud system or must use public cloud infrastructure. In both cases, the smart grid should spend enormous amount of money for communication and data provisioning. Since the P2P-cloud is a community of the available end user resources, employing commodity hardware, no extra resource investment is necessary to manage a smart grid by exerting the P2P-cloud.

To this aim, we can integrate the pricing mechanism of both systems. The users supply the P2P-cloud resources for the smart grid, charge it according to their contribution and energy consumption and the community users have the opportunity to choose the platform with least energy cost to execute their services. This synergetic solution encourages the user to share as much resources as possible in the P2P-cloud, and eventuates to end user utility expense reduction, as well. Furthermore, the P2P-cloud facilitates the energy efficiency issues by employing the efficient energy provisioning capabilities of the smart grid.

At the same time and increasingly so, the need to reduce carbon footprint has greatly raised investment in heterogeneous renewable sources of energy such as water, waves, wind and sun for energy efficient smart grid. As stated in [86] the smart grid infrastructure is a combination of smart energy, information and communication subsystems. Utilizing both information and communication systems, the smart grid accomplishes precise matching of supply to demand and offers incentives to appropriate consumer behavior. These changes affect the energy waste and the carbon footprint of the grid, making it smarter and greener.

Analogously, in the context of computing, replacing expensive, gigantic, cloud datacenters by inexpensive nano-datacenters of the P2P-cloud, constructed of commodity hardware, would be a huge step towards energy efficient systems. Previously, some studies, e.g. [87], compared the smart grid to the Internet. In the next section we survey the smart grid and P2P-cloud potential conjunction points.

4.1.1 Smart Grid and P2P-cloud collaboration potential

There is a growing body of work centered on exploiting the cloud and peer to peer platforms for the smart grid computing [82, 84, 85]. In a cloud computing environment, flexible datacenters offer scalable computing, storage and network resources to any Internet-enabled device on demand. Moreover, P2P communication platforms can manage the massive amount of data from distributed sources of consumption, generation and network nodes. On the other hand, diverse energy sources of smart grid improves the availability, sustainability and environment friendliness of the cloud services.

In [88], Niyato, et al. proposed a cooperative game based approach to manage the virtual machines of a cloud in a more energy efficient way by being aware of smart grid resources. In [89], analysing the power flow of datacenters, authors formulated a service request routing mechanism that considers the load balancing of distributed datacenters, which leads to energy consumption balance in datacenters and helps to grid energy management.

Moreover, there are some studies[90] on how to leverage a Peer-to-Peer platform as the ICT infrastructure of Smart grid. For instance, the CoSSMic project[91] aims to develop the ICT tools needed to facilitate the sharing of renewable energy within a neighbourhood. Cisco also proposed the combined platform of fog and cloud computing for smart grid data processing[92]. P2P clouds[93] and ClouT[94] approached this issue in a more general view by targeting the Internet of things enabled smart homes and cities.

A Community cloud-enabled smart grid can benefit from the following advantages:

- **Facilitating the development:** It is easier to develop community cloud especially in the urban areas which facilitates the development of smart grid as well.
- **Providing communication and computing platform:** P2P-cloud provides both communication and computing platform while classic cloud relies on Internet for communication.
- **P2P-cloud offers user-enabled control mechanism:** a user can control the applications whereas they are open source or each user can develop her own applications employing APIs such as REST.
- **Hierarchical data processing:** Smart grid data analysis on time series data perfectly matches the parallel data analysis. Data analysis algorithms can run on subsets of data, i.e. a subset of users' data chosen according to the locality property, stored on different machines, and aggregate them into the final result set through hierarchical, multi-level processing. As with the distributed storage, the distributed parallel processing is harnessing the network of commodity hardware to its fullest, in which the amount of available memory and computing power is abundant. Moreover, aggregation gives the possibility to anonymize data, which is a safe and secure way to retrieve business intelligence information to personalize the services without jeopardizing the end user privacy.

Nonetheless, smart grid can provide various energy sources for the community services. Charging according to the energy price, users are more concerned about the energy sources and prices, therefore, we make a broad range of choices for the users via providing the users with smart grid resource availability data. In the next section we introduce an economic model that facilitates the mutual collaboration of these systems.

4.2 Cloud of Energy

Smart grid aware ICT service provisioning can foster the idea of green ICT by better employment of energy sources. On the other hand, there are some endeavors to leverage ICT platform for smart grid communication and information subsystems. Besides, with the idea of Internet of Energy, Internet not only can serve as the communication infrastructure for the smart grid, but also the distributed mechanisms designed to manage the Internet and tackle the administration issues can inspire the solution space of smart grid challenges, which is called Internet thinking of smart grid [95]. All the same, the cloud is already proposed as the information subsystem for the smart grid, in the state of the art studies [82, 84, 85, 96]. Previous work suggests also how P2P-cloud can be leveraged as the information subsystem at the smart micro-grid level [P1].

Partly inspired by Internet of Things (IoT), Internet of Energy (IoE) [97] is about providing energy as a service in a more efficient way by dynamically adjusting resources to deliver energy at the lower cost and the higher quality possible in the context of smart grid.

In line with the idea of Internet of Energy, we define Cloud of Energy (CoE). CoE outlines how involving customers in future ubiquitous society-driven energy conservation efforts can both foster the adoption of green energy, as well as green cloud due to the increasing energy awareness of society. The rationale is to get users into the loop, not only to guide them how to use the services, but also to involve them directly in the whole cycle of control, production and provisioning of energy. Ubiquitous society makes it possible to combine informational support with fostering intrinsic motivation of users, all over the generation, provisioning and control stack by acquiring immediate feedback on society state.

Moreover, a large-scale distributed management system is required that can process huge amounts of event data and operate in real time. It should be able to manage the interface with infrastructures such as service market platforms that support the cooperation of various players. It, thus, helps to automatically balance highly fluctuating supply and demand, in a reliable and cost-effective manner. Relying on crowd sourcing [98] in a ubiquitous society, we can obtain needed services by soliciting contributions from the society rather than from traditional suppliers.

4.2.1 Challenges

There are some differences in cloud and smart grid services that should be taken into account in CoE planning. To design a comprehensive model for integration, we need to face the following challenges which stem from the natural differences of computing and energy systems.

- **Flow Management**: data flow management is way more flexible than energy flow management. In other words, we can encapsulate and label data easily, while it is not easy to route the electrons in the same way. Thus, implementing VPC is easier than developing a VPP.

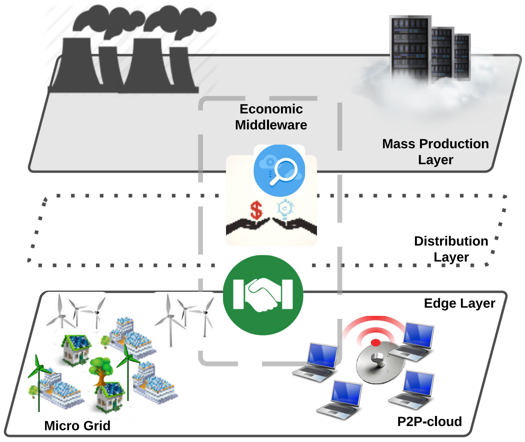


FIGURE 4.1: CoE Architecture

- **Storable Services**: in smart grid, batteries can save energy. Therefore, energy service can be stored instead of instantaneously offered to the demands, while it is not possible to store the computing service.
- **Stochastic behavior**: both systems are conforming to a stochastic behavior due to resource fluctuation and highly evolving topology, regarding origin of requests and availability of resources. In other words, due to the unpredictable collaboration paradigm of end users in the cloud, the system depicts a stochastic behavior. Likewise, in the smart grid energy provisioning system, we observe a stochastic behavior of renewable energy resources participating in the system, which is tightly coupled with the weather condition of each geographical region. However, the demand paradigm in the smart grid is more predictable than the cloud (more remarkable difference between peak and low usage). The electricity consumption pattern is almost fully determined in advance in the smart grid. The peak demand time is almost predictable in the grid system, while it is not as easy to foresee the demand pattern in a distributed computing environment.
- **Service Diversity**: the diversity of provided services in the computing platform is vaster than in the smart grid. This leads to the more complicated QoS and management mechanisms in the clouds.

4.2.2 CoE Architecture

CoE is inspired by the idea of federating ubiquitous P2P network platform and the classic distributed data centers to form a multi-layer interactive architecture. CoE fulfils hierarchical control system goals in the integrated system of XaaS that supports both computing and energy service provisioning.

CoE offers establishing Virtual Power Plant (VPP) and Virtual Private Cloud (VPC) for each vicinity through the local broker. VPP leverages existing grid networks to tailor electricity supply and demand services for a customer. VPP maximizes value for both the end user and the distribution utility using a set of software-based dynamic systems to deliver value in real time, and can react quickly to changing customer load conditions.

All the same, Virtual Private Cloud (VPC) is a cost-effective solution to expand the presence into the public cloud instead of expanding private infrastructure. With its pool

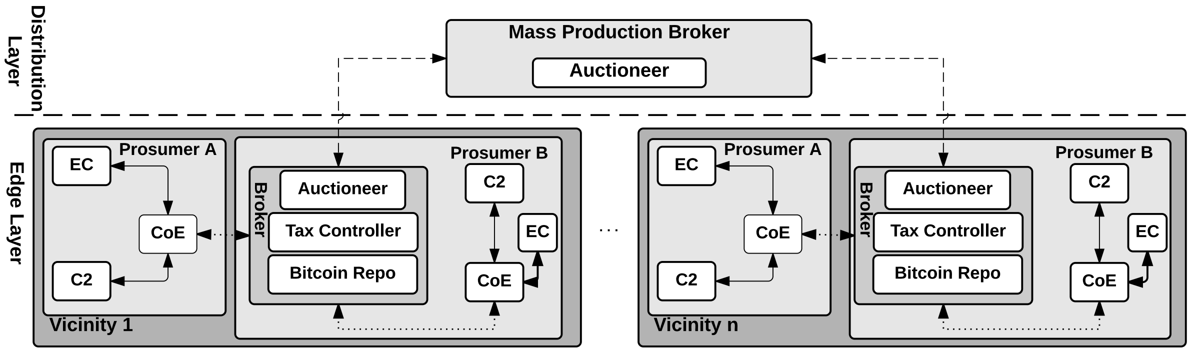


FIGURE 4.2: Economic Middleware Architecture

of highly available compute, storage, and networking resources, VPC fits well in scenarios involving variable or bursting workloads, test and development, and next generation mobile applications.

In CoE, there is a pool of providers, i.e. energy and computing service providers, including prosumers in the edge layer, and mass producers in the higher layer. CoE layered architecture assures quality of service via improving resource availability in edge-layer by the support from the mass production layer. A layered architecture of CoE is illustrated in Figure 4.1. Horizontal layers represent a hierarchical division of the service providers. Prosumers, i.e. consumers and retail service producers, at the bottom layer constitute the edge layer locally under the concept of vicinity, as illustrated in Figure 4.2. Classic cloud service providers and mass energy providers are categorized as the mass service providers in the highest level. The lower layers promote energy efficiency in resources usage and the employment of greener sources of energy. Meanwhile, the higher levels can ensure resource availability and cope with power variations in edge-layer power output.

In the CoE architecture, hierarchical brokers are responsible for managing the market in different layers. These brokers are cross layer agents that are in charge of hosting auctions and providing feedback to the layers below and above, in the economic middleware, as demonstrated in Figure 4.2. In this architecture, there is a bidirectional information flow. While wholesale brokers are statically placed, local controller/broker agents can be dynamically placed in any prosumer location providing that the prosumer can obtain the computing and energy requirements for the broker. In broker placement the priority is with the source which has excess energy generated. To reinforce the fault tolerance of the distributed system, we store the data in distributed data storage accessible to all the prosumer agents in the vicinity if they have access to the token. Dynamic local controller placement contributes to the energy efficient data processing and movement, which is the key for a sustainable system.

4.2.3 Economic Middleware

An Economic Middleware acts as an interface to facilitate smart electricity and ubiquitous computing service trading. This middleware, as shown in Figure 4.2 includes the following components:

Energy Controller(EC) module exists in each prosumer side, which is able to predict and measure the energy consumption of each individual appliance at home. All EC units are connected to the energy provider through a communication infrastructure such as a community network [99].

Computing Controller(C2) in each prosumer of ubiquitous society plays the same role of EC for the computing services.

Local broker, further to hosting auctions is responsible for defining tax rate based on the bids it receives.
If the demand and supply do not match and the vicinity encounters resource scarcity, the broker decreases tax rate, through tax controller, to make the external resources more affordable for end users. Moreover, the broker should submit the bids for the higher level broker, to obtain the resources for excess demand of the vicinity. A bitcoin repository component is responsible to keep the bitcoin balance of the vicinity which is necessary for trading with mass production broker, in the outside world. Bitcoin [100] is an online payment system, in which trade parties can transact directly without the interference of any intermediary, through bitcoin.

Mass production broker is in charge of setting up auctions among different service providers for the demands submitted by the local brokers.

4.2.4 Agent Based CoE Service Composition

The CoE platform, as illustrated in Figure 4.1, can be modeled with the concept of multi agents. Multi agent systems are the most suitable platform to model distributed collaborative systems requirements based on their properties and functionality, allowing them to implement intelligence in the smart grid control due to their social ability, flexibility, self-healing features and economic agent support [101].

Environment: In the CoE agent based model, we have nested environments through the horizontal hierarchy of the architecture, which amount to a set of producers and consumers, and brokers. Looking closer, prosumers make a rich, heterogeneous environment which is controlled by coordinators, in order to drive the prosumers behavior and represent the interest of a group of prosumers on the market.

Agents: In CoE, agents include prosumers, brokers in different levels, service providers and mass producers of electricity and cloud services. Prosumer agents produce services in the retail level and are the end users of the services, at the same time. Each prosumer is equipped with a cloud and electricity controller, to regulate and control its demand and supply.

Broker agents in different layers can decide what strategies to employ both on the market and prosumers. For that we can apply a Stackelberg game, which is a hierarchical game where players of this game are leaders and followers across the hierarchy. The Stackelberg leader is the wholesale market broker and the local brokers should follow its strategy in the market. However, each broker can run its own double auction mechanism to supply the demands locally. This property gives the authority to the autonomous local brokers to run their own strategy as long as it does not violate the wholesale market's framework. This promotes decentralization, better scalability and speed of adjustment to varying local conditions, while bounding global imbalances.

Utility and cloud service providers can trade the mass provider services on their behalf via the mass production broker.

Market Rules: Since energy and computer systems provide two different services, to integrate these two systems, in our market model, we need a metric that can measure the contribution of each service in an understandable scale for the other. Moreover, a universal metric facilitates the collaboration of the two systems. Virtual money seems to be an appropriate metric for this end. Defining local currency in the micro-grid-community level, we can incentivise the users to collaborate in the system by sharing the resources, i.e. energy and computing by earning credits. The idea behind defining a local currency is to drive and improve the coordination of users within a community, to promote the community among the others by elevating the value of their local currency against the other communities. Moreover, this mechanism helps in load balancing by changing the value of local currency, by allowing arbitration.

When local supply exceeds the local demand, the local broker can assign bitcoin [100] generation tasks to the prosumeres offering resources, in exchange of certain amount of local currency based credit in their account. Therefore, the available resources are not effectively lost and can be re-acquired later from mass producers, if supply is scarce, in the vicinity. This is specially useful when the energy powering the idle resources is green energy that is being under-utilized. Thus, we can in a novel way, effectively attempt at preserving resources and energy for later demand.

Thus, local brokers, to provide resources from outside the vicinity, can only rely on some outside currency, i.e. the bitcoin generated in the vicinity when there are excess resources of electricity and computing in the vicinity (as an ideal universal replacement to any legal tender or precious metal). Afterwards, to deliver the service to the end user, local broker charges the users based on the community currency value equivalent to the amount of bitcoin and the associated conversion taxes.

Furthermore, to keep the system constraints, we define the exchange tax, which is an extra amount that should be drawn from the requesters' credit due to service provisioning. To exemplify, communities geographically far will set higher exchange rates to assure the quality of service, i.e. reduced latency, lower transmission loss and more energy efficient service provisioning. Note that the Virtual money defined here deviates from the state of the art concept in terms that it does not necessarily follow the conservation property.

In the next section we assess the feasibility of the proposed CoE system.

4.3 Evaluation

We study the challenges of rolling out the CoE and elaborate the feasibility of the proposed architecture by answering several questions across this section.

4.3.1 A Comparison on Smart Micro Grid and P2P-cloud

As discussed so far, the design goals of the P2P-cloud appear to be nearly identical to those of the smart grid; the similarities and differences of smart grid and P2P-cloud are listed on Table 4.1. Both of them attain the basic requirements of a modern society in a large scale and distributed manner, namely electricity, communication and computing. Both infrastructures are conforming to a stochastic behavior due to resource fluctuation and highly evolving topology, regarding origin of requests and availability of resources.

Table 4.1: Smart Grid and P2P-cloud similarities and differences

Features and Properties	**P2P-cloud**	**Smart Grid**
Scale	large	
Evolution rate	high	
Time Dependency	Time critical	
Billing mechanism	pay-as-you-go	
Resource behavior	Fluctuating resources	
Suppliers	Distributed	
Market behavior	non monopolistic	
Transparency	Consumers are unaware of the underlying complexity	
demand	Distributed and unpredictable	
Service Cost	Cost Effective	
Resource management	Distributed	
Hardware Costs	**Cheap**	**Expensive**
Range of services	**Diverse**	**Limited**
Storage Support	**Full support**	**some degree**
Availability	**Important**	**Critical**
(Byzantine) Fault Tolerance	Yes	
Scalability	Critical	
Reliability	Critical	
Consistency	Critical	
Data Security	Critical	

Loosely paraphrasing, due to the unpredictable collaboration paradigm of end users in the P2P-cloud, the system depicts a stochastic behavior.

Likewise, in the smart grid energy provisioning system we observe a stochastic behavior of renewable energy resources participating in the system, which is tightly coupled with the weather condition of each geographical region. To exemplify, in a windy day, wind farms generate a lot of energy, while the solar panels reach their extreme productivity on a perfect sunny day. Both smart grid and P2P-cloud follow the bidirectional flow property, since most of the nodes collaborating in the distributed set of users and suppliers, serve as prosumers, i.e. PROducers and conSUMERs concurrently, to respond to the distributed demand for energy and information. Although the collaborative distributed systems supply the demands in a distributed manner, consumers are unaware of the underlying network complexity.

The most remarkable property of the both systems is "pay-as-you-go" mechanism employed in these systems, that eradicates the heavy investment for the centralized infrastructure and revokes the supplier monopolies thanks to the decentralized, collaborative structure.

For both systems the fundamental goal is to effectively integrate a number of separately administered existing networks into a common utility network. The common secondary

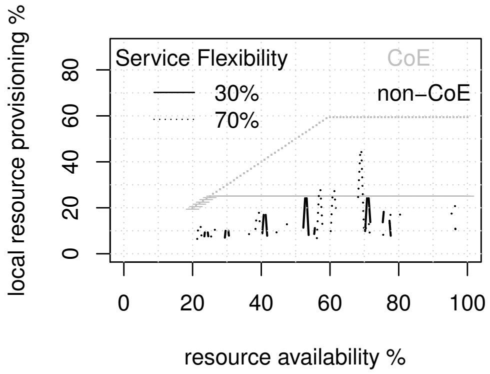


Figure 4.3: Collaboration

design goals are: 1) to tolerate loss of individual components, 2) to support different underlying infrastructure types, 3) to allow distributed resource management, 4) to be cost effective, 5) to allow easy endpoint attachment and 5) to be accountable for resource usage. To thrive a collaborative distributed network, we should consider a cost effective design as the most striking issue, while bearing in mind the fundamental design principles of a distributed system including scalability, reliability, availability, consistency, fault tolerance, distributed resource management and data security.

Moreover, some of the problems in the P2P-cloud, that of aggregation of stochastic sources, distributed resource management, multiple time scales of control, and user incentivization, are similar to that faced in the smart grid.

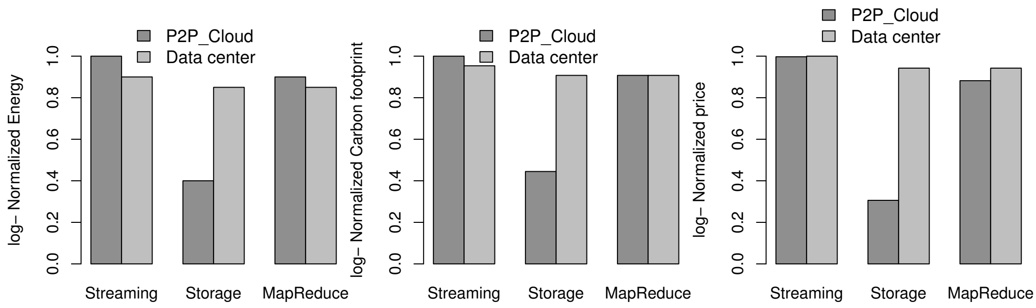
Nonetheless, it is not all about the similarities, there are some differences as well. The demand paradigm in the smart grid is more predictable than the P2P-cloud (more remarkable difference between peak and low usage); the electricity consumption pattern is almost fully determined beforehand in the smart grid. The peak demand time is almost predictable in the grid system, while it is not as easy to foresee the demand pattern in a distributed computing environment. Even so, many global services experience predictable peak and low periods for each time zone.

Furthermore, the diversity of provided services in the P2P-cloud is much higher than in the smart grid. This leads to the more complicated QoS and management mechanisms in the P2P-cloud. Additionally, the computing hardware costs follow a downward trend, while the hardware expenses of the smart grid rises day to day.

4.3.2 Is bi-level architecture incentivize the collaboration?

Defining cost as the main incentive, CoE can improve the collaboration among the prosumers, through the credit earning mechanism. Figure 4.3 illustrates that more resources are provided within the vicinity in CoE compared to the random resource allocation mechanism.

Here, we only consider flexible service provisioning in the edge to assure the quality of service due to the uncertainty of renewable retail generators. Both electricity and



a) Energy consumption b) Carbon emission c) Price

FIGURE 4.4: Energy, Carbon and Price of different services

computing services can be classified as rigid and flexible. While rigid service needs real time resource provisioning, flexible services can be scheduled for a later time, and is more flexible.

As illustrated in Figure 4.3, local resource provisioning depends on service flexibility and resource availability in the vicinity. Here we studied two service models, with 30% and 70% flexibility. The results show that the more resources available in the vicinity, the higher collaboration of prosumers occur in CoE compare to the random collaboration case. The collaboration in non-CoE case, however, is weekly correlated to the resource availability in the vicinity. Note that in CoE we do not consider the possibility of the inter-vicinity collaboration, since there is a significant transmission loss and quality of service degradation in this case.

Implication 1: Increasing the resource availability at the edge layer of the CoE should be considered as a priority to attain the smart grid objectives.

4.3.3 How much energy can be saved in CoE?

Figure 4.4.a depicts how much energy can be saved by smart service provisioning in CoE[2]. We see that some cloud services such as storage as a service in the P2P-cloud, i.e. edge layer, is more energy efficient compare to the data center case, while two other services are better to obtain via data centers in the higher layer. Therefore, combination of edge devices and data centers result in a more energy efficient services providing that resources are allocated in an energy efficient manner. For this end, a framework is required to characterize the energy efficacy of each individual service in both platforms. A decision support system can help afterwards according to the analysis results.

Increasingly, in Figure 4.4.b, the carbon emission of different services are compared. We assume that the prosumers are equipped with the solar roof tops, which emit 41 g/kWh and data centers equipped with 50% of renewable solar energy produced by solar PV at utility level and generate 48 g/kWh of $CO2$ in average and 50% of brown energy inducing 802 g/kWh of carbon footprint in an average case, according to [102]. This figure reveals the fact that, carbon emission as an incentive, besides energy consumption may turn the table in more cases in favor of P2P-cloud, due to the lower emission rate of prosumer level renewable energy generators.

[2]Experiment setup is the same as what we described in the previous chapter.

Implication 2: carbon emission rate is a better metric than energy consumption to quantify the efficacy of the system in fulfilling smart grid objectives.

4.3.4 How much cost will be saved?

In the state-of-the-art mechanisms, computer services are priced regardless of the energy consumption cost. However, energy aware service provisioning can save remarkably in the provider costs, since energy is a major part of dynamic price in the cloud service provisioning. Exerting CoE, we have a better chance of directing services to the appropriate layer of provisioning, and saving energy cost as a consequence.

Besides, CoE provides an opportunity to share the infrastructure and data required in smart grid and cloud instead of duplicating the resources. Namely, in case of carbon based charging, finding a cheap energy source will be significantly important. In such a case, being renewable energy sources aware can help saving in dynamic cost. CoE as an integrated architecture will obtain the smart grid data to the brokers across the hierarchy, instead of duplicating this data in two separate systems of cloud and smart grid.

Figure 4.4.c illustrates the cost of energy in a carbon based energy pricing, which assigns the same price to all the energy sources and applies carbon taxes according to the carbon-footprint portion attributed to the electricity source. As shown in this figure, in all cases, P2P-cloud service provisioning leads to cost saving. Nevertheless, we should bare in mind that there is limited resource availability for local resource provisioning and the quality of service may not be obtained in local service providing.

4.3.5 Is implementation complexity warranted?

CoE reveals that integration facilitates a diverse range of service exchange. However, integration may incur more complexity to the economic layer in the system due to the different nature of each system such as uncertainty level, storablity, flow management complexity, etc. This added complexity should be warranted with the advantages of integration, e.g. more effective marketplace. To attain CoE goals, we need a robust economic model which can manage the demand and supply in a multi-variable marketplace.

Nonetheless, if we aim at greening the ICT while exerting ICT for green, CoE can be a good candidate to reduce carbon emission, save energy and cost as a consequence of smart service provisioning.

4.4 Summary

In this chapter we introduced Cloud of Energy (CoE). CoE envisions the service provisioning framework of the future that provides everything as a service via an integrated cloud and smart electricity grid platform in horizontal and vertical dimensions. CoE facilitates the resource management in each of smart grid and cloud through their hierarchy. It also expedites the horizontal integration of different services via their shared economic incentives. The economic layer acts as a middleware to translate a service in every concept, e.g. energy and computing, to the common incentive scale of money. Integration elevates the collaboration of diverse range of providers and consumers, requesting for different services. Moreover, an integrated system is more efficient and greener, since it avoids unnecessary redundancy in the common sub-systems, such as shared data, computing and communication infrastructure, etc. Also the integration leads to greener system since it provides increased energy awareness. However, this is just the very first step in introducing the idea and still there are several open questions that should be investigated more profoundly. For instance, a mechanism should be designed for energy aware pervasive resource trading in the hierarchical broker system of introduced middleware.

Chapter 5

Work Plan and Conclusion

5.1 Work Plan

Tentative outcomes of the work and the research questions are listed below. The correlation of each question and the tentative contributions are illustrated in Figure 5.1.

5.1.1 Research Questions

Question1: Is it energy efficient to switch to community cloud?

Question2: Which metric should be applied for the analysis?

Question3: What are the requirements of the energy consumption analysis framework?

Question4: How can we exert cheap, green, distributed energy sources?

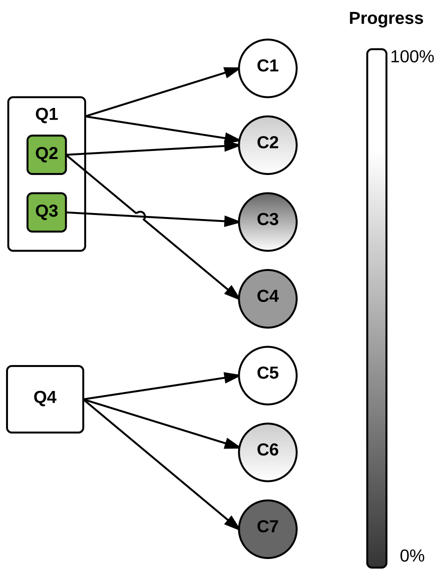


FIGURE 5.1: Correlation of the questions and milestones

5.1.2 Milestones

Contribution1: Analysing the energy efficiency of P2P-cloud.

Contribution2: Comparing the energy consumption of P2P-cloud and classic cloud models.

Contribution3: A framework to analyse energy consumption in P2P-assisted cloud ecosystems.

Contribution4: A metric for performance-aware energy analysis across the service provisioning stack.

Contribution5: Study the viability of smart micro-grid and P2P-cloud integration.

Contribution6: A framework to integrate smart grid and cloud service provisioning.

Contribution7: Dynamic energy-based pricing.

To accomplish this work, we have been relying on the schedule in Table 5.1.2.

Milestone	Outcome	Deadline	Status
C1	CloudComm 2014	December 2014	Published
C2	Journal of Grid Computing	December 2014	Under Review
C3	IEEE Transactions on Cloud Computing	April 2015	Under Review
C4	CCGrid	October 2015	In progress
C5	SmartgridComm2014	November 2014	Published
C6	SmartgridComm2015	November 2015	Under Review
C7	ACM e-energy	January 2016	Initial Stage
PhD Thesis	Thesis	March 2016	In progress

5.2 Conclusion

In this proposal, we explained the work plan to devise an economics inspired energy aware service provisioning in P2P assisted cloud ecosystems. Work plan is designed to address two major questions enumerated as following. First we need to find out if it is energy efficient to move toward P2P-clouds. Addressing this question requires a framework to compare energy consumption for each service, as sketched in Chapter 3.

Nonetheless, this analysis framework may be trapped with tremendous performance degradation. Therefore, a performance aware energy analytic metric is needed to tackle with this issue. We introduced energy effectiveness metric conceptualized to the energy and performance requirements of each layer across the service provisioning stack, i.e. application, VM/OS, hardware. Energy effectiveness can assess how successful is the ecosystem from a particular perspective based on different granularity of information.

Moreover, we introduced the idea of Cloud of Energy to make the ecosystem greener. Since energy and Information and Communication Technology (ICT), as two driving forces of the contemporary life, are reshaping themselves based on ubiquitous society architecture to improve their service quality. Within the reforming process, integration of two systems can contribute to a greener ubiquitous society by equipping them with the concept of energy conservativeness, and leveraging renewable energy sources. In this work we outlined the idea of Cloud of Energy (CoE) which fosters the adoption of green energy and green cloud by integrating these two systems. CoE introduces an integrated framework of everything as a service to facilitate the service exchange, not only across the computing and electricity grid hierarchy, but also among them via an economic middleware.

References

[1] Cheng Wang, Neda Nasiriani, George Kesidis, Bhuvan Urgaonkar, Qian Wang, Lydia Y Chen, Aayush Gupta, and Robert Birke. Recouping energy costs from cloud tenants: Tenant demand response aware pricing design. In *ACM e-energy 2015*, 2015.

[2] Atefeh Khosravi, Saurabh Kumar Garg, and Rajkumar Buyya. Energy and carbon-efficient placement of virtual machines in distributed cloud data centers. In *Euro-Par 2013 Parallel Processing*, pages 317–328. Springer, 2013.

[3] Nakku Kim, Jungwook Cho, and Euiseong Seo. Energy-credit scheduler: An energy-aware virtual machine scheduler for cloud systems. *Future Generation Computer Systems*, 2012. ISSN 0167-739X. doi: http://dx.doi.org/10.1016/j.future.2012.05.019.

[4] Xibo Jin, Fa Zhang, Lin Wang, Songlin Hu, Biyu Zhou, and Zhiyong Liu. A joint optimization of operational cost and performance interference in cloud data centers. *arXiv preprint arXiv:1404.2842*, 2014.

[5] Ata EH Bohra and Vipin Chaudhary. Vmeter: Power modelling for virtualized clouds. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1–8. Ieee, 2010.

[6] Rakpong Kaewpuang, Sivadon Chaisiri, Dusit Niyato, B Lee, and Ping Wang. Cooperative virtual machine management in smart grid environment. 2013.

[7] Ramon Bertran, Marc Gonzàlez, Xavier Martorell, Nacho Navarro, and Eduard Ayguadé. Counter-based power modeling methods: Top-down vs. bottom-up. *The Computer Journal*, 56(2):198–213, 2013.

[8] Amin M. Khan, Leila Sharifi, Luís Veiga, and Leandro 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, October 2013.

[9] SungJin Choi, HongSoo Kim, EunJoung Byun, MaengSoon Baik, SungSuk Kim, ChanYeol Park, and ChongSun Hwang. Characterizing and classifying desktop grid. In *Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on*, pages 743–748. IEEE, 2007.

[10] Xu Zhiqun, Chen Duan, Hu Zhiyuan, and Sun Qunying. Emerging of telco cloud. *Communications, China*, 10(6):79–85, 2013.

[11] Yunhao Liu, Li Xiao, Xiaomei Liu, Lionel M Ni, and Xiaodong Zhang. Location awareness in unstructured peer-to-peer systems. *Parallel and Distributed Systems, IEEE Transactions on*, 16(2):163–174, 2005.

[12] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7(1-4):72–93, 2005.

[13] Vana Kalogeraki, Dimitrios Gunopulos, and Demetrios Zeinalipour-Yazti. A local search mechanism for peer-to-peer networks. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 300–307. ACM, 2002.

[14] Mohamed Hefeeda and Osama Saleh. Traffic modeling and proportional partial caching for peer-to-peer systems. *Networking, IEEE/ACM Transactions on*, 16 (6):1447–1460, 2008.

[15] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, Albert Zomaya, et al. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*, 82(2):47–111, 2011.

[16] Frank Bellosa. The benefits of event-driven energy accounting in power-sensitive systems. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, pages 37–42. ACM, 2000.

[17] Michael A Bender, Gerth Stølting Brodal, Rolf Fagerberg, Riko Jacob, and Elias Vicari. Optimal sparse matrix dense vector multiplication in the i/o-model. *Theory of Computing Systems*, 47(4):934–962, 2010.

[18] Nikhil Bansal, Tracy Kimbrel, and Kirk Pruhs. Speed scaling to manage energy and temperature. *Journal of the ACM (JACM)*, 54(1):3, 2007.

[19] Michael T Goodrich. Data-oblivious external-memory algorithms for the compaction, selection, and sorting of outsourced data. In *Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures*, pages 379–388. ACM, 2011.

[20] Tibor Horvath, Tarek Abdelzaher, Kevin Skadron, and Xue Liu. Dynamic voltage scaling in multitier web servers with end-to-end delay control. *Computers, IEEE Transactions on*, 56(4):444–458, 2007.

[21] Ricardo Koller, Akshat Verma, and Anindya Neogi. Wattapp: an application aware power meter for shared data centers. In *Proceedings of the 7th international conference on Autonomic computing*, pages 31–40. ACM, 2010.

[22] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.

[23] Swapnoneel Roy, Atri Rudra, and Akshat Verma. An energy complexity model for algorithms. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 283–304. ACM, 2013.

[24] Vijay Anand Korthikanti and Gul Agha. Towards optimizing energy costs of algorithms for shared memory architectures. In *Proceedings of the twenty-second annual ACM symposium on Parallelism in algorithms and architectures*, pages 157–165. ACM, 2010.

[25] Marcus Hähnel, Björn Döbel, Marcus Völp, and Hermann Härtig. Measuring energy consumption for short code paths using rapl. *ACM SIGMETRICS Performance Evaluation Review*, 40(3):13–17, 2012.

[26] Tao Li and Lizy Kurian John. Run-time modeling and estimation of operating system power consumption. *ACM SIGMETRICS Performance Evaluation Review*, 31(1):160–171, 2003.

[27] Gilberto Contreras and Margaret Martonosi. Power prediction for intel xscale® processors using performance monitoring unit events. In *Low Power Electronics and Design, 2005. ISLPED'05. Proceedings of the 2005 International Symposium on*, pages 221–226. IEEE, 2005.

[28] Michael D Powell, Arijit Biswas, Joel S Emer, Shubhendu S Mukherjee, Basit R Sheikh, and Shrirang Yardi. Camp: A technique to estimate per-structure power at run-time using a few simple parameters. In *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, pages 289–300. IEEE, 2009.

[29] Shinan Wang, Hui Chen, and Weisong Shi. Span: A software power analyzer for multicore computer systems. *Sustainable Computing: Informatics and Systems*, 1 (1):23–34, 2011.

[30] Ramon Bertran, Yolanda Becerra, David Carrera, Vicenç Beltran, Marc Gonzàlez, Xavier Martorell, Nacho Navarro, Jordi Torres, and Eduard Ayguadé. Energy accounting for shared virtualized environments under dvfs using pmc-based power models. *Future Generation Computer Systems*, 28(2):457–468, 2012.

[31] John C McCullough, Yuvraj Agarwal, Jaideep Chandrashekar, Sathyanarayan Kuppuswamy, Alex C Snoeren, and Rajesh K Gupta. Evaluating the effectiveness of model-based power characterization. In *USENIX Annual Technical Conf*, volume 20, 2011.

[32] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A Bhattacharya. Virtual machine power metering and provisioning. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 39–50. ACM, 2010.

[33] Daniel Versick, Ingolf Waßmann, and Djamshid Tavangarian. Power consumption estimation of cpu and peripheral components in virtual machines. *ACM SIGAPP Applied Computing Review*, 13(3):17–25, 2013.

[34] W Lloyd Bircher and Lizy K John. Complete system power estimation: A trickle-down approach based on performance events. In *Performance Analysis of Systems & Software, 2007. ISPASS 2007. IEEE International Symposium on*, pages 158–168. IEEE, 2007.

[35] Yan Zhai, Xiao Zhang, Stephane Eranian, Lingjia Tang, and Jason Mars. Happy: Hyperthread-aware power profiling dynamically. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)(Philadelphia, PA*, pages 211–217, 2014.

[36] Maxime Colmant, Mascha Kurpicz, Pascal Felber, Loïc Huertas, Romain Rouvoy, and Anita Sobe. Process-level power estimation in vm-based systems. In *European Conference on Computer Systems (EuroSys)*, page 14. ACM, 2015.

[37] Akshat Verma, Puneet Ahuja, and Anindya Neogi. pmapper: power and migration cost aware application placement in virtualized systems. In *Middleware 2008*, pages 243–264. Springer, 2008.

[38] Stefan Janacek, Kiril Schroder, Gunnar Schomaker, Wolfgang Nebel, Marco Ruschen, and Gunter Pistoor. Modeling and approaching a cost transparent, specific data center power consumption. In *Energy Aware Computing, 2012 International Conference on*, pages 1–6. IEEE, 2012.

[39] James W Smith, Ali Khajeh-Hosseini, Jonathan Stuart Ward, and Ian Sommerville. Cloudmonitor: Profiling power usage. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 947–948. IEEE, 2012.

[40] Qingwen Chen, Paola Grosso, Karel van der Veldt, Cees de Laat, Rutger Hofman, and Henri Bal. Profiling energy consumption of vms for green cloud computing. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, pages 768–775. IEEE, 2011.

[41] Jan Stoess, Christian Lang, and Frank Bellosa. Energy management for hypervisor-based virtual machines. In *USENIX annual technical conference*, pages 1–14, 2007.

[42] Madhurima Pore, Zahra Abbasi, Sandeep KS Gupta, and Georgios Varsamopoulos. Energy aware colocation of workload in data centers. In *High Performance Computing (HiPC), 2012 19th International Conference on*, pages 1–6. IEEE, 2012.

[43] Simon Schubert, Dragan Kostic, Willy Zwaenepoel, and Kang G Shin. Profiling software for energy consumption. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 515–522. IEEE, 2012.

[44] Hui Chen, Youhuizi Li, and Weisong Shi. Fine-grained power management using process-level profiling. *Sustainable Computing: Informatics and Systems*, 2(1): 33–42, 2012.

[45] Fahimeh Alizadeh Moghaddam, Thomas Geenen, Patricia Lago, and Paola Grosso. A user perspective on energy profiling tools in large scale computing environments. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2015*, pages 1–5. IEEE, 2015.

[46] Luca Ardito, Giuseppe Procaccianti, Marco Torchiano, and Giuseppe Migliore. Profiling power consumption on mobile devices. 2013.

[47] Shuai Hao, Ding Li, William GJ Halfond, and Ramesh Govindan. Estimating mobile application energy consumption using program analysis. In *Software Engineering (ICSE), 2013 35th International Conference on*, pages 92–101. IEEE, 2013.

[48] Swapnoneel Roy, Atri Rudra, and Akshat Verma. Energy aware algorithmic engineering. In *Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2014 IEEE 22nd International Symposium on*, pages 321–330. IEEE, 2014.

[49] Josep Subirats and Jordi Guitart. Assessing and forecasting energy efficiency on cloud computing platforms. *Future Generation Computer Systems*, 2014.

[50] Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29 (4):1012–1023, 2013.

[51] Pedro da Silva Dias Rodrigues, Carlos Nuno da Cruz Ribeiro, and Luís 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, April 2010.

[52] José Simão and Luís 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, December 2013.

[53] José Simão and Luís Veiga. 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.

[54] Daniel Wong and Murali Annavaram. Knightshift: Scaling the energy proportionality wall through server-level heterogeneity. In *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*, pages 119–130. IEEE, 2012.

[55] Niraj Tolia, Zhikui Wang, Manish Marwah, Cullen Bash, Parthasarathy Ranganathan, and Xiaoyun Zhu. Delivering energy proportionality with non energy-proportional systems-optimizing the ensemble. *HotPower*, 8:2–2, 2008.

[56] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A scalable, commodity data center network architecture. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 63–74. ACM, 2008.

[57] Albert Greenberg, James R Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A Maltz, Parveen Patel, and Sudipta Sengupta. Vl2: a scalable and flexible data center network. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 51–62. ACM, 2009.

[58] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. Bcube: a high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Computer Communication Review*, 39(4):63–74, 2009.

[59] Lei Huang, Qin Jia, Xin Wang, Shuang Yang, and Baochun Li. Pcube: Improving power efficiency in data center networks. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 65–72. IEEE, 2011.

[60] Paolo Costa, Austin Donnelly, Greg O'Shea, and Antony Rowstron. Camcubeos: a key-based network stack for 3d torus cluster topologies. In *Proceedings of the 22nd international symposium on High-performance parallel and distributed computing*, pages 73–84. ACM, 2013.

[61] Dennis Abts, Michael R Marty, Philip M Wells, Peter Klausler, and Hong Liu. Energy proportional datacenter networks. In *ACM SIGARCH Computer Architecture News*, volume 38, pages 338–347. ACM, 2010.

[62] Dong Lin, Yang Liu, Mounir Hamdi, and Jogesh Muppala. Flatnet: Towards a flatter data center network. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 2499–2504. IEEE, 2012.

[63] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1548–1557. IEEE, 2001.

[64] Llorenç Cerdà-Alabern, Axel Neumann, and Pau Escrich. 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, November 3–8, 2013. ACM.

[65] Jayant Baliga, Kerry Hinton, and Rodney S Tucker. Energy consumption of the internet. In *Optical Internet, 2007 and the 2007 32nd Australian Conference on Optical Fibre Technology. COIN-ACOFT 2007. Joint International Conference on*, pages 1–3. IEEE, 2007.

[66] Sergiu Nedevschi, Sylvia Ratnasamy, Jitendra Padhye, and Intel Reserch. Hot data centers vs. cool peers. In *HotPower*, 2008.

[67] Vytautas Valancius, Nikolaos Laoutaris, Laurent Massoulié, Christophe Diot, and Pablo Rodriguez. Greening the internet with nano data centers. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 37–48. ACM, 2009. ISBN 978-1-60558-636-6. doi: 10.1145/1658939.1658944.

[68] Dhruba Borthakur. The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, 11(2007):21, 2007.

[69] M. Selimi, F. Freitag, R. Pueyo Centelles, and A. Moll. Distributed storage and service discovery for heterogeneous community network clouds. In *Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on*, pages 204–212, Dec 2014.

[70] Jayant Baliga, Robert WA Ayre, Kerry Hinton, and RodneyS Tucker. Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE*, 99(1):149–167, 2011.

[71] Thomas A de Ruiter. A workload model for mapreduce. *MSc thesis at TU Delft.*, 2012.

[72] Johan Pouwelse, Koen Langendoen, R Lagendijk, and Henk Sips. Power-aware video decoding. In *22nd Picture Coding Symposium, Seoul, Korea*, pages 303–306, 2001.

[73] László Gyarmati and Tuan Anh 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*, pages 183–186. ACM, 2010.

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

[75] Priya Mahadevan, Sujata Banerjee, and Puneet Sharma. Energy proportionality of an enterprise network. In *Proceedings of the first ACM SIGCOMM workshop on Green networking*, pages 53–60. ACM, 2010.

[76] Green space, http://www.greenpeace.org/international/global/international/planet-2/report/2010/3/make-it-green-cloud-computing.pdf.

[77] Friedemann Mattern, Thorsten Staake, and Markus Weiss. Ict for green: how computers can help us to conserve energy. In *Proceedings of the 1st international conference on energy-efficient computing and networking*, pages 1–10. ACM, 2010.

[78] Vlad Coroama and Lorenz M Hilty. Energy consumed vs. energy saved by ict–a closer look. In *Environmental Informatics and Industrial Environmental Protection: Concepts, Methods and Tools, 23rd International Conference on Informatics for Environmental Protection, Berlin*, pages 353–361, 2009.

[79] Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai, and Thomas Sandholm. What's inside the cloud? an architectural map of the cloud landscape. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pages 23–31. IEEE Computer Society, 2009.

[80] Ketan Maheshwari, Marcus Lim, Lydia Wang, Ken Birman, and Robbert van Renesse. Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud. *Innovative Smart Grid Technologies*, 2013.

[81] Kenneth P. Birman, Lakshmi Ganesh, and Robbert van Renesse. Running smart grid control software on cloud computing architectures. *Workshop on Computational Needs for the Next Generation Electric Grid*, 2010.

[82] Christoph Gerdes Sebnem Rusitschka, Kolja Eger. Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain. *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 483 – 488, 2010.

[83] et. al. Y. Simmhan. On using cloud platforms in a software architecture for smart energy grids. *IEEE International Conference on Cloud Computing (CloudCom)*, 2010.

[84] Akash K Singh. Smart grid cloud. *International Journal of Engineering Research and Applications*, pages 674 – 704, 2012.

[85] Alberto Leon-Garcia Amin Mohsenian-Rad. Coordination of cloud computing and smart power grids. *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 368 – 372, 2010.

[86] et al. Xi Fang. Smart grid—the new and improved power grid: A survey. *IEEE Communications Surveys and Tutorials*, 14(4):1 – 37, 2011.

[87] Srinivasan Keshav and Catherine Rosenberg. How internet concepts and technologies can help green and smarten the electrical grid. In *Proceedings of the first ACM SIGCOMM workshop on Green networking*, Green Networking '10, pages 35–40, 2010. ISBN 978-1-4503-0196-1.

[88] Rakpong Kaewpuang, Sivadon Chaisiri, Dusit Niyato, B Lee, and Ping Wang. Cooperative virtual machine management in smart grid environment. *IEEE Transaction on Services Computing*, 2013.

[89] A-H Mohsenian-Rad and Alberto Leon-Garcia. Coordination of cloud computing and smart power grids. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 368–372. IEEE, 2010.

[90] Peer energy cloud, http://www.peerenergycloud.de/.

[91] Cossmic, http://www.cossmic.eu/.

[92] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh 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*, pages 13–16. ACM, 2012.

[93] Pau Escrich, Roger Baig, Ivan Vilata, Axel Neumann, Marc Aymerich, Ester Lopez, Davide Vega, Roc Meseguer, Felix Freitag, and Leandro Navarro. Community home gateways for p2p clouds. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–2. IEEE, 2013.

[94] Kenji Tei and Levent Gurgen. Clout: Cloud of things for empowering the citizen clout in smart cities. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 369–370. IEEE, 2014.

[95] Srinivasan Keshav and Catherine Rosenberg. How internet concepts and technologies can help green and smarten the electrical grid. *ACM SIGCOMM Computer Communication Review*, 41(1):109–114, 2011.

[96] György Dán, Rakesh B Bobba, George Gross, and Roy H Campbell. Cloud computing for the power grid: From service composition to assured clouds. In *In Proceedings of 5th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud'13)*, 2013.

[97] Nicola Bui, Angelo P Castellani, Paolo Casari, and Michele Zorzi. The internet of energy: a web-enabled smart grid system. *Network, IEEE*, 26(4):39–45, 2012.

[98] Daren C Brabham. A model for leveraging online communities. *The participatory cultures handbook*, 120, 2012.

[99] Bart Braem, Chris Blondia, Christoph Barz, Henning Rogge, Felix Freitag, Leandro Navarro, Joseph Bonicioli, Stavros Papathanasiou, Pau Escrich, Roger Baig Viñas, et al. A case for research with and on community networks. *ACM SIGCOMM Computer Communication Review*, 43(3):68–73, 2013.

[100] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system.

[101] Jurica Babic. Agent-based modeling of electricity markets in a smart grid environment, 2014.

[102] IPCC Working Group. Ipcc wk. group iii – mitigation of climate change, annex ii i: Technology - specific cost and performance parameters. 2014.

[P1] Sharifi, Leila, Felix Freitag, and Luis Veiga. "Combing Smart Grid with community clouds: Next generation integrated service platform." In Smart Grid Communications (SmartGridComm), 2014.

[P2] Sharifi, Leila, Navaneeth Rameshan, Felix Freitag, and Luis Veiga. "Energy Efficiency Dilemma: P2P-cloud vs. datacenter." In CloudCom2014, **Best Paper Candidate**.

[P3] Sharifi, Leila, Felix Freitag, and Luis Veiga. "Envisioning Cloud of Energy." Submitted to Smart Grid Communications (SmartGridComm), 2015,**Under Review**.

[P4] Sharifi, Leila, Jose Simao, Navaneeth Rameshan, Felix Freitag, and Luis Veiga. "A Framework to Analyse Energy Effectiveness in P2P Assisted Cloud Ecosystems." Submitted to IEEE Transactions on Cloud Computing, **Under Review**.

[P5] Sharifi, Leila, Llorence Cerda-Alabern, Felix Freitag, and Luís Veiga. "Energy Efficient Cloud Service Provisioning: Keeping Data Center Granularity in Perspective." Submitted to Journal of Grid Computing, Special Issue on Green Cloud Computing, **Under Review**.