AEQUO: Enhancing the Energy Efficiency in Private Clouds Using Compute and Network Power Management Functions

Kai Spindler*, Sven Reißmann†, Ronny Trommer*, Christian Pape*, Sebastian Rieger* and Thomas Glotzbach‡
*Department of Applied Computer Science, Fulda University of Applied Sciences, Fulda, Germany
{kai.spindler, ronny.trommer, christian.pape, sebastian.rieger}@cs.hs-fulda.de
†Computing Centre, Fulda University of Applied Sciences, Fulda, Germany
sven.reissmann@rz.hs-fulda.de
‡Elektrotechnik und Informationstechnik, Hochschule Darmstadt University of Applied Sciences, Darmstadt, Germany
thomas.glotzbach@h-da.de

Abstract—**Today's data centers need a huge amount of energy for their operation. Private cloud infrastructures using virtualization technologies are the prevailing paradigm in modern data centers and their energy consumption and the corresponding ongoing operational costs are not negligible. Solutions that raise the energy efficiency allow reductions in these operational costs and optimizations of the utilization of the data center infrastructure. Also, renewable energy sources can help to provide the needed energy, but usually these sources are fluctuating. Therefore, the energy is not always available when needed and also not always produced near the point of use. Further, the storage of energy is not available in industrial scale. The following article examines the possibility to shift the energy consumption of virtual machines and presents a lightweight prototype that can be integrated in private cloud environments using standard OpenStack components and application programming interfaces. It optimizes the energy efficiency by observing the current utilization parameters of compute resources and by taking appropriate actions based on this data. Furthermore, we evaluated mechanisms to control the energy efficiency of network resources. This optimization will be carried out by an automated instance, possessing a comprehensive view on the data center assets, which relocates virtual machines and optimizes the network structure. The article is completed with an evaluation to measure power consumption of data center assets during virtual machine live-migration operations and also illustrates further areas of research.**

Keywords–Private Cloud; Energy Efficiency; Renewable Energy; Computer Networks; Power Management.

I. INTRODUCTION

Private or enterprise cloud solutions are currently gaining more and more momentum, mainly driven by the success of cloud-based services [1] and virtualization, but also by the ongoing eavesdropping scandals that hinder the use of public cloud providers for sensitive information. One of the major benefits of cloud-based services is formed by their scalability. This scalability is supported by the "elasticity" [2] of the underlying infrastructure that allows providers to support large-scale applications and services [3] for a vast number of mobile devices (e.g., smart phones, tablets) and users from all over the world. However, the improvement in scalability is achieved at the cost of larger data centers and growing energy consumption. Energy is not only needed to supply the IT infrastructure itself with electricity, but also for appropriate cooling. Hence, energy costs are one of the major challenges for current data centers.

Since cloud services are based on distributed systems, besides compute and storage, another essential resource is the network, enabling fast and decentralized access to the services over the Internet and especially the Web. This is also described as "broad network access" in [2]. To provide cloud and web-based services, efficient IT virtualization techniques and computer networks are necessary. These technologies in turn have an impact on the energy consumption and cost. Hence, adaptive power management based on the current requirements, i.e., the load on the applications and services, helps to increase the energy efficiency by turning components on and off or reducing their performance (e.g., throttling, energy saving functions). Such adaptive power management functions can also balance or consolidate the power consumption in private cloud environments. As cloud services are provided on an "on-demand" basis according to [2], an adaptive management based on the current load of the resources is supported by this major cloud paradigm.

In [1], we presented a solution to enhance the energy efficiency in OpenStack-based private cloud environments. This article elaborates on the implementation and concepts outlined in [1] and introduces a combination with renewable energy. A special focus is put on the efficient placement of virtual machines (VM) and the reduction of power required by network connections and components. Adaptive placement of VMs also permits a reduction of compute and storage power consumption by consolidating them on specific hosts, addressing the "resource pooling" requirement for cloud computing environments given in [2]. However, migration costs need to be considered. Hence, this article includes an evaluation of the power consumption of compute, storage and network components during VM migrations. A prototype that was implemented to monitor the energy efficiency (e.g., compute, storage and network utilization as well as temperature and thermal efficiency of the cooling) in cloud environments was presented in [1]. It includes throttling, enabling or disabling resources based on the current demand and given constraints (e.g., required fault tolerance, redundancy, quality of service parameters and network connectivity). The prototype uses standard cloud APIs (application programming interfaces) (i.e., OpenStack, Open Cloud Computing Interface (OCCI)). Therefore, it can easily be integrated in existing cloud infrastructures using standard OpenStack components.

The paper is laid out as follows. Section II gives an overview on private clouds based on OpenStack and describes the requirements for energy efficiency in such private cloud environments. Also, examples for existing techniques to enhance the energy efficiency in computer networks and references to related research projects are given. A major aspect of the research project behind this paper focuses on the use of renewable energy and to enhance the energy efficiency of distributed data centers. Therefore, Section III evaluates renewable energy fluctuation in Germany and defines requirements to leverage renewable energy sources for the energy-efficient use of resources in distributed data centers. Requirements for the implementation of our prototype, to enhance the energy efficiency by combining the state of the art techniques and extending them, are defined in Section IV. The implementation of our prototype and mechanisms to optimize the energy efficiency in private clouds are presented in Section V. Section VI describes an experimental testbed that was used for the evaluation of our concept and the implemented prototype, being presented in Section VII. Finally, Section VIII draws a conclusion, evaluates our research findings and outlines future work that will be pursued in the research project.

II. STATE OF THE ART

The following sections give an overview on the deployment of private clouds using OpenStack and examine the requirements for the energy efficiency of such environments. A special focus is drawn on the potential of energy-efficient computer networks. Additionally, related research projects are discussed.

A. *OpenStack-based Private Clouds*

The term cloud is an ambiguous concept and has been interpreted in many ways by vendors and customers of cloud services. One of the most sophisticated definitions is documented in NIST SP 800-145, expressing cloud computing as ”a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (i.e., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [2]. NIST identifies five essential characteristics, three service models, and four deployment models. Our work focuses on private cloud deployments with OpenStack, which is a software project that provides an open source implementation of technologies for building and operating public and private cloud environments using the ”Infrastructure as a Service” (IaaS) service model. In OpenStack, this infrastructure is built by offering networking resources (named Neutron), compute resources (Nova), and storage resources, i.e., object storage (Swift) and block storage (Cinder). Additionally, OpenStack offers many more services for management and orchestration, such as Horizon and Heat, its identity service Keystone, and a telemetry service called Ceilometer.

The IaaS service model in OpenStack is implemented by providing VMs, which can run as Nova instances on the compute nodes of an OpenStack environment. The placement of VMs, being one of the main objectives of our work, can be on a specific Nova node or may depend on various parameters of the environment. Also, the migration of a running VM from one compute node to another, as well as the starting or stopping of VMs depending on the current load is possible during the lifecycle of a service. This flexibility provides some interesting aspects in terms of resilience (i.e., by seamlessly moving VMs from one data center to another) but also in terms of energy efficiency as we will demonstrate in detail later in this paper.

B. *Energy Efficiency in Private Clouds*

In today’s rapidly-growing IT infrastructures, energy efficiency is no longer a secondary requirement, but has rather become one of the main objectives when planning and operating new data centers. One reason for this development is the common sensitization for an ecologically sustainable use of global resources. Furthermore, large-scale data centers consume enormous amounts of electrical power not only for running the IT systems, but also for cooling them. A measure for the ratio between the energy used by the computing equipment and the overall energy consumption of a data center is the power usage effectiveness (PUE), which takes into account, i.e., the energy needed for cooling and losses by (uninterruptible) power supplies [4]. At the same time, PUE has an impact on the operational overhead cost of a data center, hence its minimization is of great interest for today’s data center operators, which have to act economical while facing increasing energy costs [5].

It can be said that cloud computing by definition leads to energy efficiency through its operational concepts, which include a better utilization of physical resources, dynamic scaling based on the current load, and location-independent and efficient resource management. However, to take advantage of these concepts, the whole cloud infrastructure needs to be carefully adapted to the operators’ individual needs. For instance, resource pooling allows a cloud operator to consolidate multiple VMs providing various services on only a few physical hosts, hence increasing the efficiency of these hosts. At the same time, rapid elasticity and on-demand self-service concepts require the immediate and automatic availability of compute power if needed, therefore instant availability of additional resources is required [2].

The energy consumption of a VM running in OpenStack depends mainly on the energy requirements of its physical IaaS components, including compute (i.e., CPU (central processing unit), RAM (random-access memory)), storage (i.e., SAN (storage area network), NAS (network-attached storage), HDD (hard disk drive)), and networking components (i.e., routers, switches), but also on the distance between the components involved (e.g., the distance of the storage from the compute node). Consequently, the real power consumption ratio of a cloud service depends on the number of active compute, storage, and networking components needed to provide it. As VMs can be migrated from one physical host to another, it is possible to take advantage of fluctuating electricity prices or to adapt the load factor of a data center to climatic changes. This could be done not only by consolidating VMs in one data center, but also by sending the VMs to another geographical location, where operational costs are lower.

In OpenStack, the placement of VMs on a specific cloud computing fabric controller (Nova) is determined mainly by nova-scheduler [6]. While several techniques are offered for optimal VM placement, by default the so called Filter Scheduler is used. It supports the placement of a VM based on a physical location, available compute resources (e.g., CPU, RAM), or by its requirements for secondary resources, such as the availability of specific storage or network capabilities. Moreover, the Filter Scheduler addresses the operational requirements for resilience or consolidation of VMs by explicitly allowing a placement on different hosts or by grouping them on a single host. However, it does not take into account any energy efficiency parameters, neither for initial placement nor for the live-migration of VMs. Also, automatic migration of a VM in favor of load balancing or energy efficiency enhancements is not supported by nova-scheduler. Nevertheless, with its components for service orchestration (Heat) and telemetry (Ceilometer), OpenStack provides interfaces to manage VM migration that can be extended to evaluate energy consumption or cooling requirements.

C. Energy-efficient Computer Networks

Another aspect to take into account when measuring the energy consumption of a VM running in OpenStack is the networking equipment. Since nearly every current service or application is used over a network, the relevance of this aspect is rather obvious. In private clouds, the relevance is increased even further as cloud services are typically formed by a combination of different interdependent services in a data center. This can especially be observed in large cloud providers, where intra data center network traffic is several orders of magnitude higher than the traffic going out to the Internet [7].

From a theoretical point of view a network consists of multiple nodes, which are interconnected using links. Hence, a network and its resulting topology can be defined as a number of nodes and links. Looking at the power consumption, most of the links, especially in local area networks are passive, meaning that they do not consume individual power, but rather serve as a medium that carries electromagnetic or optical signals being generated at the nodes as a sender. Depending on the link characteristics (e.g., attenuation), signals might need to be refreshed, for example on long-distance links to allow the receiver to interpret the signal correctly. Optical and electromagnetic amplifiers can be applied to refresh the signal. Therefore, especially long-distance links can include amplifiers or special transceivers, e.g., directly attached to the cable. An overview on the cumulative energy consumption for the required amplifiers in long-distance optical networks is presented in [8]. In the model for the power consumption of a network, these transceivers and amplifiers can also be treated as nodes. In terms of power consumption these nodes are active components of the networks, as they individually draw power to refresh, send, receive and interpret the signals and the contained data. Nodes can interpret and modify the transfered data on different layers of the OSI or TCP/IP reference model.

The complexity of the protocols being interpreted in a network node, and the decapsulation necessary to get the corresponding protocol headers, are a major factor for the power consumption at the network nodes besides the power that is needed to send and receive the signals. Link characteristics (e.g., bandwidth, attenuation, length) define the power that is needed to send and receive the signals. All active components and their corresponding power consumption can be considered to enhance the energy efficiency of computer networks. Links have only an indirect influence on the power consumption. However, if links can be reduced, shortened or exchanged against media that support a higher energy efficiency, the power consumption of the network nodes can be lowered even further, though this is not always possible since locations of nodes and quality of links sometimes is implied by the physical location or local circumstances.

According to [9], computer networks typically account for 15–25% of the total energy consumption in data centers. The increasing number of users and the complexity of cloud services require a high bandwidth, which leads to increasing link speeds and, therefore, raises the power consumption of each switch port. This is also observed in [8], where the slope of energy consumption per bit at network devices like routers over the last years is lower than the slope of the continually increasing peak access bit rate. Accordingly, the total power consumption of networks is still increasing due to the increasing access bit rate and number of users regardless of improved networking equipment that consumes less watt per transfered bit. An overview on energy-efficient data center networks is given in [10]. Redundant links are required to assure resilience of the network, again increasing the power consumption. Because of the increased power consumption for higher bit rates and the number of redundant links, some researchers [11][12] already claim that the fraction of the energy consumed by the network in a data center is likely to rise to up to 50% in the near future. Concepts like Equal Cost Multipathing (ECMP) or Multipath TCP are available to utilize the equipment and redundant links up to the maximum capacity of the networks.

As today's networks are mostly not energy proportional [11][13], higher utilization of the network and its equipment leads to an increased energy efficiency of the network. However, variable bandwidth requirements (e.g., decreased utilization during nighttime) makes it economically reasonable to scale down the network as well [14]. For wired local area networks (LAN), which we primarily focus on, there are already some power management techniques being offered by network equipment providers. First and foremost, the LAN standard 802.3 was extended in 2012 to include 802.3az, also called Energy Efficient Ethernet (EEE) [15]. Since this extension is part of the regular 802.3-2012 standard, it is likely that in the near future all Ethernet equipment will support EEE. However, EEE was specifically designed for copper-based network links. With increasing bandwidth requirements, most links especially in the aggregation or core layer use optical links. While it is currently not possible to lower the link speeds of fiber physical transceivers (PHY) to reduce their energy consumption [16], the transceivers and hence optical links can be powered off if currently not needed [17]. Compared to copper PHYs, which support an idle state leveraging EEE, fiber PHYs unfortunately do not support an automatic wakeup, being related to the missing capability to lower the link speed [18]. Therefore, the power management of optical links currently needs an external power controller or network management system.

While network equipment manufacturers who include EEE in their products claim that 802.3az allows a reduction of the energy consumed by a single copper port by up to 81% [19], this benefit comes with the price of increased latency during the low power idle (LPI) phase [20]. Regarding the fact that currently data center network infrastructures are moving to 10 Gbit/s Ethernet and beyond, where power consumption per port is usually over 5 Watts [19], the power savings for the entire data center infrastructure are even higher. Furthermore, there are other vendor-specific power management functions of networking components (e.g., Cisco EnergyWise [5]) that are not covered by EEE. Also, as mentioned above, network power management techniques could be improved by temporarily powering off unused optical links or network functions. Unfortunately, such techniques also have a negative impact on the latency due to the power management and necessary wakeup cycles. Compared to power management functions of compute and storage resources (e.g., Advanced Power Management (APM), Advanced Configuration and Power Interface (ACPI)) that have constantly evolved over the last decades, power management functions for network components are relatively new and supposedly need to be improved due to energy efficiency requirements in the near future [12].

Existing solutions for energy-efficient networks concentrate on the reduction of the local power consumption on individual network components and ports, but they are typically unaware of the current global requirements in the entire network, especially when multiple network equipment providers are used. Therefore, their scope is rather limited and the energy efficiency optimization is rather isolated. As networks are a fundamental building block of private clouds and since ”broad network access” is also an essential characteristic of cloud services [2], a holistic approach to enhance energy efficiency in private clouds should take all three cloud infrastructure components (compute, storage and network resources) into account.

Some research projects, notably Stanford's ElasticTree [21] have identified this problem, but did not integrate it with an appropriate placement of VMs and especially did not discuss the requirements of private clouds. Also, these solutions only state that switching off network components and their functions might be an option, but do not leverage or present corresponding network power management functions.

By using a network controller in private or enterprise clouds that is aware of the entire network topology, such network power management functions could be implemented for example to disable currently unused links or to throttle link rates during off-peak times while still maintaining fault tolerance requirements, e.g., in multipath network environments. Additionally, such a controller could enable power saving modes, standby or disable networking functions to lower the power consumption of the nodes. Existing network power management, as described above, could be combined with this approach. Moreover, such a controller could also activate and deactivate entire networking components based on the current requirements to enhance the energy efficiency. Hence, energy-efficient computer networks could reduce the power consumption of the network nodes (and the number of active links) to the minimum, especially in off-peak times while still ensure fault-tolerance and high performance under load. Possible solutions are presented in the forthcoming sections of this paper.

D. Related Work

Energy-efficient placement of virtual machines in OpenStack private cloud environments is also discussed in [22][23][24]. However, these approaches do not consider an optimal placement of VMs with respect to temperature, cooling and network connectivity requirements. Furthermore, these publications focus more on the evaluation of different algorithmic approaches for an optimal placement of VMs while keeping the cost of migrations low, than on the integration, thereby this paper will not go into detail on the evaluation of different algorithms. Additionally, the extensions presented in these papers cannot be used with the current Juno (nor the previous Icehouse and Havana) Release of OpenStack.

A more generalized evaluation of an energy-efficient placement of VMs in cloud environments and relevant parameters is given in [25] and [26]. However, these contributions do not offer testbeds for OpenStack environments. A tool that allows for distributing virtual machines considering the migration cost is introduced in [27]. It includes a basic analysis of migration cost and the impact of live-migration for an application. The CÆSARA project [28] outlines an algorithm for the energy-efficient placement of virtual machines. Basic concept is the estimation of a server's energy consumption based on the running virtual machines' characteristics. Furthermore, a distributed algorithm used for virtual machine placement in large cloud environments is discussed in [29]. The idea here is that every server knows the CPU load of the other physical servers. Each server tries to comply with an upper and lower threshold for the CPU load and initiates the migration of virtual machines when these thresholds are violated. Also, bin packing algorithms that form the basic concept of virtual machine placement on physical servers are still subject of current scientific studies and research [30][31][32].

Our research also highlighted the lack of studies examining the relationship between energy consumption and communication distance. Instead, merely average estimations are determined in the form of energy consumption per download quantity. In [33], a holistic view on energy consumption of network transactions including also the embedded energy resources used to manufacture network devices is given. The focal point of this publication are transmissions to end customers (e.g., including Digital Subscriber Line Multiplexers (DSLAMs) and telephone lines). Excluding these costs, the energy demand stated in this study is 149 Wh/GB for embedded energy and 849 Wh/GB for the real transmission, so in sum 998 Wh/GB $\approx$ 0.1 kWh/GB. Similar values were determined in [34] by measuring transmissions at an international conference between Switzerland and Japan. In this publication, the authors relied upon pessimistic assumptions, so a realistic value of 0.2 kWh/GB was postulated, thus, a higher value than in other studies. Furthermore, a comprehensive study from the year 2009 by the German OFFIS institute [35] identified possible savings by load management across multiple data centers and forecasted an energy demand of 0.1 kWh/GB for the year 2014, which corresponds with recent studies.

Concerning energy-efficient computer networks, especially the ElasticTree project [21] presented interesting starting points and related work for power management and throttling of network components using OpenFlow. The ideas of ElasticTree were extended, e.g., in the ECODANE project [36] to include traffic engineering. Also, theoretical energy-aware optimizations of data center networks were presented in [37][9]. Requirements and constraints for energy-efficient placement of VMs regarding the network connectivity, were explored in [38][39][40]. However, these solutions do not include existing power management techniques like the ones we described in the previous sections for networking resources (e.g., [9][19][20]). Furthermore, these approaches do not include power management functions like ACPI and related solutions. In our work, we combine the existing power management mechanisms and the solutions that were discussed in the related work given in this section, and present a lightweight extension to leverage power management techniques in existing OpenStack enterprise clouds.

III. USING RENEWABLE ENERGY TO REDUCE THE POWER CONSUMPTION OF DISTRIBUTED DATA CENTERS

While data centers (DC) usually need a lot of energy for their operation, environmental compatibility plays an increasingly important role. A number of key figures, like the already mentioned power usage effectiveness (PUE), as well as the data center infrastructure efficiency (DCIE) and carbon usage effectiveness (CUE) [41] need to be considered, when planning new or optimizing the efficiency of existing data centers. A number of certification programs are available, to provide an incentive for organizations to build efficient IT infrastructure. As an example, ”Der Blaue Engel” (The Blue Angel) [42], a well known quality seal for environmental compatibility assigned by the German federal environment agency has recently enlarged its certification program for data centers. Criteria for the award include appropriate PUE monitoring, application of efficient hardware components, as well as meeting most of the data centers electricity demand from renewable energies, such as hydroelectric power, photovoltaics (PV), wind power, biomass energy, or from combined heat and power generation plants. In the future, it is conceivably that legal obligation will force large data centers to meet some of the requirements of such certification programs. As a result, an energy supply with the help of the renewable energy sources would be eligible. However, from the perspective of the energy suppliers one problem occurs.

The energy output of renewable energy sources is fluctuating. That means the energy is not always available when needed or vice versa. In addition, the energy is not always produced near the point of use (e.g., offshore wind energy [43]). First of all, this leads to the necessity to store the energy in between [44] or shift the consumption in time [45]. Until now, the storage of energy is just conditionally feasible, as it is expensive and not available in industrial scales. In contrast, the possibility to shift the energy consumption of data centers with the help of an intelligent energy management is viable. Regarding wind energy it is obvious that in the case of heavy winds at the shore, the produced energy has to move via overhead lines, which means that the power grid has to be extended and rebuilt in future [46]. While in the classical approach, the energy has to be moved from the power plant to the location of the data center, the following solution is focusing on the other way around. The services of the data center as consumer will be moved to the place of energy production. This is supported by the increasing interest in server, storage and network virtualization like software-defined networking, supporting current IT and cloud infrastructures. Therefore, migrations should be as fast and cheap as possible to benefit from the advantages of fluctuations in renewable energy sources and related savings of operational costs of local distributed data centers.

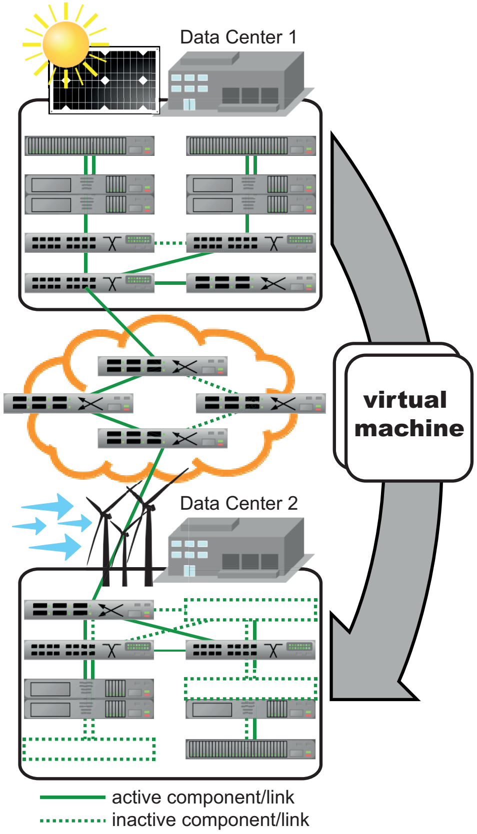

Figure 1. Benefits of renewable energy and a proper energy management for data centers.

Figure 1 shows an overview of an intelligent energy management in data centers with software-defined networks and renewable energies. Due to the current demand of the shown DC 2, not all available IT resources are actively used. Inactive components and connections are marked with dashed lines. If an energy surplus arises due to strong winds and available onsite wind turbines, inactive components can be immediately activated and virtual machines (virtual resources) or applications can be transferred from DC 1 to DC 2. Conversely, the virtual machine can be shifted from DC 2 to DC 1 if no wind energy is present but energy is supplied by photovoltaic power plants. Current server virtualization and infrastructure solutions enable this migration of IT resources in the background. As DC 2 uses a different Internet connection, the only noticeable difference to the users of the resettled services might be a higher or in ideal case a lower latency related to the new distance between user and service. The DC 2 uses a different Internet connection. Related to the new distance between user and service is just a higher or in ideal case a lower latency noticeable while using the service. Within Germany, these latency changes are generally below 50 milliseconds. Initially, a time series analysis of meteorological data has been carried out over three years' data to determine the potential of shifting services from one DC to another.

The meteorological data used was derived from the years 2011 to 2013 at three different locations. The weather conditions in terms of renewable energy depend mainly on the latitude. Therefore, the locations Cuxhaven (northern Germany), Frankfurt am Main (in the middle of Germany) and Munich (southern Germany) were picked. From these locations, measured values were used with an interval of 15 minutes. Overall, 105.120 individually measured values per time series were evaluated. The most important short and medium term sources for electrical power supply from renewable energies are wind and solar energy (photovoltaic). For this reason, the focus is put on three indicators for wind speed in m/s, the irradiance in W/m^2 and the ambient temperature in °C. Based on these indicators, it is possible to calculate the power

TABLE I. Overview of the installed capacity of wind and photo-voltaic power divided into northern, central and southern Germany [47][48].

Separate

Region	Wind [MW]	Wind [%]	PV [MV]	PV [%]
Northern Germany	17,305	44.2%	6,289	16.4%
Central Germany	16,854	43.0%	13,550	35.4%
Southern Germany	5,005	12.8%	18,402	48.1%

Total

Region	Wind	PV
Northern Germany	73.3%	26.7%
Central Germany	55.4%	44.6%
Southern Germany	21.4%	78.6%

(power time series) for wind and photovoltaic energy, using mathematical methods. Due to the existing north-south divide of wind speed and irradiance, there is a certain distribution of power plants. The wind speed has a strong increase from south to north. The biggest values of wind speed are measured on the German coast in the far north. Therefore, the biggest percentages of wind turbines are located in the north and in the plain regions of central Germany. The situation in photovoltaic is exactly the opposite. In southern Germany, you will find a larger irradiance and for this reason a bigger energy yield. Thus, most photovoltaic power plants are based in southern Germany. Table I shows an overview of the installed capacity of wind power and photovoltaic power divided into northern, central and southern Germany [47][48]. The calculated power time series of the two energy sources were converted into percentage values to make a conclusion out of current power distribution of the three places. The determined distributions of Table I were included.

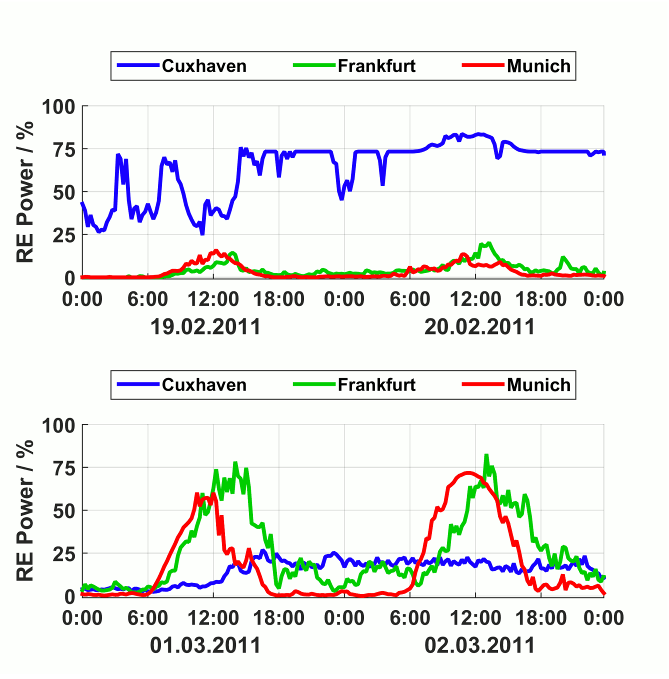

Figure 2. Percentage power outcome of renewable energies on four days.

Figure 2 shows an example of the daily distribution of power. In the upper chart it can be seen, that Cuxhaven has a lot bigger outcome of renewable energy sources than at the two other sides. At this time, the services of the DCs in southern and central Germany should be shifted to the DCs in northern Germany. In the lower part of the figure it is shown that, at least during the day, more power in Frankfurt and Munich is available. In that case a corresponding shift should occur.

TABLE II. Times when a displacement would be possible (data in hours and the percentage of total time (26,280h)).

To From	Cuxhaven	Frankfurt	Munich
Cuxhaven	-	581h (2.2%)	1,454h (5.5%)
Frankfurt	2,654h (10.1%)	-	1,093h (4.2%)
Munich	2,971h (11.3%)	771h (2.9%)	-

In Table II, the evaluation of the whole time series over three years is shown. In each case, two locations were compared to each other to calculate the sum of time in hours, to find a situation where a displacement would be possible. Table II shows the maximum possible hours and the percentage of total time (26,280h) when a displacement would be possible (best case scenario). This scenario always occurs if the difference between the current powers at the two sides is greater than or equal to 30%. A great potential becomes visible. The next step is now to show if and when a displacement is possible and reasonable. A Mathlab/Simulink simulation, which is under construction, should provide information about the potential of shifting DCs. These simulations will consider models of complete DCs and use official weather data to include renewable energy sources in the evaluation.

IV. ENERGY-EFFICIENT PLACEMENT AND NETWORK CONNECTIVITY OF VIRTUAL MACHINES

In the following sections, we describe various capabilities of OpenStack regarding the placement of VMs and identify requirements for adding energy efficiency criteria to this process. A special focus is laid on the energy efficiency of the network connection between VMs in distributed private clouds.

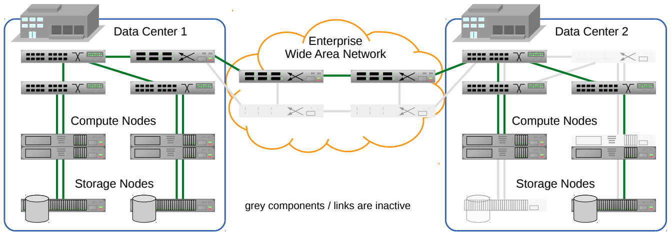

Figure 3. Power management for energy-efficient compute, storage and networking resources in private clouds.

Figure 3 shows an example of a private cloud IT infrastructure that is distributed over two data centers at different sites. Each data center provides compute, storage and network resources as described in Section II-A. Regarding the power management, each of these components consumes energy based on its utilization. Furthermore, as the components are connected to each other over the network, by deactivating or throttling individual components or links, the energy consumption of the private cloud can be reduced, e.g., during off-peak times. Also, redundant components or links can be deactivated completely in favor of increased energy efficiency when active fault tolerance is not needed, e.g., due to low utilization. The deactivation or throttling is symbolized by the grayed out links and components shown in Figure 3.

A. Energy-efficient Placement of Virtual Machines in OpenStack Environments

As described in Section II-A, OpenStack is not, by itself, able to manage resources with respect to energy efficiency. Therefore, we present concepts to support the decision-making process about when and how resources like VMs can be relocated to increase the energy efficiency with respect to the required dependencies (i.e., storage, network). To decide whether or not to move a VM from one host to another, it is necessary to know various metrics about the system that runs the hypervisor. Basically, two kinds of metrics are needed to support these decisions. The first is general resource information, like free RAM, disk space or system load. Using this data, it is possible to determine whether the system still has enough free resources, so that additional VMs can be moved to this host. A second metric of importance is defined by the temperature and energy consumption of the system, which is closely related to the PUE. Since the current load and the temperature of a system are closely related, it is possible to correlate these metrics, and to draw conclusions about the energy consumption of the system. Another general metric we identified to be interesting is the current electricity price at each site. Comparing price differences and migration costs, it is possible to evaluate whether energy costs can be reduced by moving VMs from one data center to another. Also, currently available or stored renewable energy can be taken into account for the evaluation.

Given all these data, it is necessary to select the desired strategy regarding the optimization of the energy efficiency. First, it is a good idea to shutdown a server completely if other servers can provide enough free resources to take over its load. Additionally, it is possible to shutdown the servers network switch ports to reduce the energy consumed by the network as mentioned in Section II-C. Besides the network, also a shutdown of other dependencies (e.g., storage resources) can be considered. Basically, there are two options to turn servers on and off. The first option is to control the server using Wake-on-LAN (WOL) if the system was put into ACPI status S3 (Suspend to RAM), S4 (Suspend to disk) or S5 (soft off). Another option is to use IP-based switchable power distribution units (PDU) to switch sockets and attached devices on and off. Using this technique, the BIOS should be configured to automatically boot the system after AC power is restored. Also, entire racks with multiple compute, storage and network equipment could be powered on and off in a controlled way, if an appropriate mechanism exists (and the contained components tolerate the shutdown, e.g., network equipment) to optimize the energy consumptions based on the strategy discussed in this section.

As shown in Figure 4, we introduce a new management component, which has a global view over all servers in the data center. Furthermore, management data from other data centers is collected to get a global knowledge about the resources at every site. The management component collects data from the computes nodes in the data center using REST requests to communicate with the OpenStack API. Based on the collected data, the management component decides when to move VMs by instructing the involved compute nodes to start a live migration process. When this process is able to free enough resources, so that one compute node becomes idle, the management component should take actions to shutdown or hibernate the corresponding compute node to save energy.

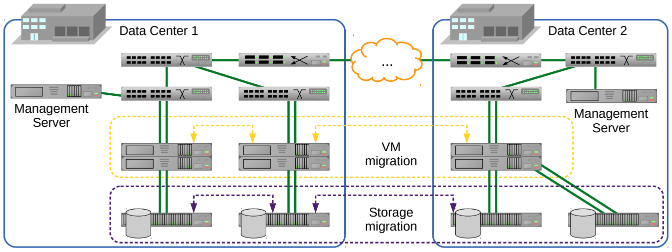

Figure 4. Integration of power management components to enable energy-efficient compute, storage and networking.

B. Energy-efficient Network Connectivity in OpenStack Environments

The complexity of computer networks with respect to energy consumption can be reduced to nodes and links of the network as described in Section II-C. Regarding the energy efficiency of a network in an OpenStack environment, two factors driving the energy consumption can be identified. First and foremost, the energy requirements are defined by the amount of nodes and links. This especially includes power dissipation at each component. Second, the utilization of each node influences its individual energy consumption. The higher the utilization, the more energy is needed for each component. However, as described in Section II-C, current networks are not energy-proportional, so the power consumption of the nodes is not proportional to the utilization of the links. Nonetheless, a sufficient utilization of all links and components leads to increased efficiency. From a theoretical point of view, the network in OpenStack environments builds a graph, with each edge representing a link. By calculating the minimum spanning tree, it is possible to identify the minimum number of links needed to connect all active components. Each link, which is not part of the minimum spanning tree represents a possible candidate to shutdown. However, the problem remains to calculate the preferable spanning tree, considering the energy efficiency and current load of each link, as well as the preferred minimum redundancy, which may differ depending on the specific network segment (i.e., core links connecting the data centers). A promising solution, which considers these requirements has been presented in [21].

To include the metrics of each link in the network a weighted graph can be defined, where the weights of the edges represent the load or utilization of the link, its performance (latency, bandwidth, jitter, failure rate) or in our specific example the energy consumption. By using a graph database (e.g., as part of a network management system), it is possible to model the topology of a network and apply energy consumption metrics to contained nodes and links. Besides classical spanning tree algorithms, as described above, algorithms that allow redundant paths and hence enable fault tolerance and load-balancing, like multiple shortest path trees, can also be used to detect the energy-efficient network topology based on the weights of the links. Network connections of a VM are given by one or multiple paths in the graph. Querying the database, the energy requirements of the network can be evaluated. Also, constraints like fault tolerant links can be defined in the database, as already described in Section II-C. Furthermore, this way the management servers are able to identify redundant links and nodes that can be turned on or off depending on the current utilization of the active links or resilience requirements. Hence, graph databases can be used to support the decision for energy-efficient network connectivity of VMs. Given the dependencies and metrics represented by weights in the graph, components and links can be deactivated or throttled, e.g., during off-peak times, or reactivated based on network utilization.

The management servers can also use the OpenStack network node (using OpenStack Neutron) or an external network management system to support the decisions regarding energy-efficient network connectivity. For example, the network management system or OpenStack Neutron could imply specific topology or performance constraints in the OpenStack environment that need to be considered despite the possibility to minimize the power consumption by deactivating, throttling or suspending network links and nodes.

V. Enhancing the Energy Efficiency of Virtual Machines in Private Clouds using AEQUO

Based on the latin word for equal, we named our prototype AEQUO, as it implements a management component to balance the power requirements in OpenStack environments. The prototype is part of a research project at the University of Applied Sciences Fulda with the purpose of creating a proof of concept to enhance the energy efficiency of cloud environments. In this section, we describe the implementation of our prototype based on the requirements that we defined earlier in Sections III and IV.

A. *Implementation of AEQUO*

AEQUO is implemented in Python, which integrates well into the testbed, as most of OpenStack's components are written in the same language and offer a Python API. The current implementation consists of a central management component which resides on the controller node. In the current prototype the CPU utilization is the only metric used to determine whether a VM should be migrated or not. Other metrics, such as the current electricity price at a site as mentioned in Section II-B, are also feasible options. AEQUO is getting the data and metrics about the compute nodes and the VMs using REST requests. The REST API is provided by the OpenStack Python API. Before it is possible to retrieve such data, it is required to authenticate against OpenStack. Figure 5 gives a brief overview of the architecture of AEQUO and the REST communication with the Openstack API.

After successful authentication against the OpenStack API, AEQUO starts to get basic data about the compute nodes, such as their name, details about the CPU (i.e, number of cores) and the current status (active or shutdown). The process of acquiring this data happens without any user intervention, since the data is provided by OpenStack Nova. At this point, AEQUO has an overview over all the active compute nodes. The next step is to acquire basic data about the virtual machines running on these compute nodes. This includes the current state of the machine, the name and the ID which identifies the VM. The prototype only includes VMs in its calculations that are in an active state, hence, VMs that are not shutdown, migrating or in a failed state. This data is also acquired by querying Nova via REST.

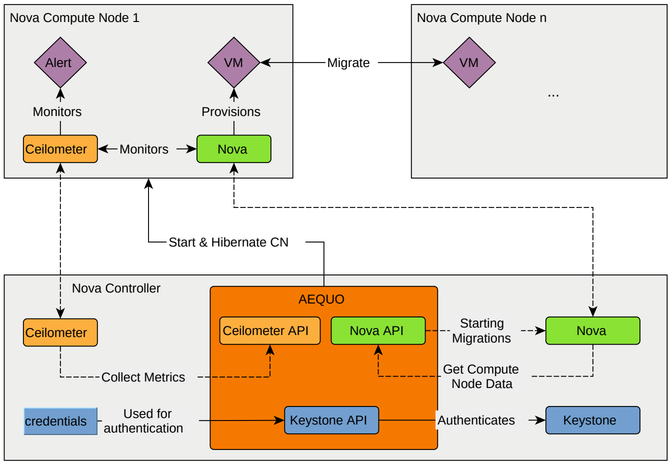

Figure 5. Architecture of AEQUO and integration into OpenStack.

After the acquisition of the metadata, the prototype queries the Ceilometer API about the metrics (e.g., CPU utilization) of the compute nodes and the VMs running on them. This process does not include VMs that are already shutdown, migrating or not in an operating state. The prototype also requires a list of compute nodes it should manage. This list contains the nodes' MAC addresses, to start them again if needed via Wake On LAN. It would be possible to automatically collect the MAC addresses of the nodes (though they are not provided by OpenStack Nova), but this would require that all compute nodes are running when AEQUO is started, which is not desirable. For this reason, the prototype requires a list (in form of a file) that is read by AEQUO upon startup containing the compute node names and their corresponding MAC addresses. This also enables the user to define compute nodes which should excluded from the management by AEQUO.

When all the data is acquired, the prototype starts to process it. First and foremost, the prototype detects two states of each compute node's utilization in which the energy consumption can be improved. The first state is the "overloaded state", in which a compute node is considered to be operating near to its maximum amount of physical resources. This critical level is defined by a hard limit. In the current prototype a compute node is considered to be overloaded, if the average CPU utilization was over 90% in the last 2 minutes. If a compute node is in the overloaded state, AEQUO evaluates which of its VMs could be moved. If there is more than one compute node being overloaded, the one with the highest CPU utilization will be selected. Two minutes after a VM was moved from an overloaded compute node, the situation is reevaluated. If the compute node is still overloaded, the next virtual machine will be moved and so on.

In each step, AEQUO individually evaluates where the VM can be migrated, to accomplish an optimal placement with respect to the total power consumption. It is a bad idea to select the compute node with the lowest CPU utilization as the target for the migration, because this compute node might have been prone to be underloaded. Therefore, the compute node with the highest utilization that is still able to handle the additional load of the moved VM without getting overloaded after the migration should be chosen. This is accomplished by looking at the current load of the compute nodes and adding the load of the VM that is to be moved. If the resulting value is lower than 90%, the destination compute node still has enough resources free to accommodate the VM. In case that there are no free resources available and a compute node is prone to be overloaded, AEQUO just prints a warning message. Since the prototype also supports hibernating unused compute nodes, there might be some nodes that could be started to get the necessary resources for the migration.

Despite of overloaded compute nodes, hosts can also be in an ”underloaded state”. In contrast to the overload state, underload detection is not done by simply looking at the lower limit. Instead, the prototype picks the compute node with the lowest current load and evaluates if the load of this node can be distributed onto the remaining nodes without causing them to be overloaded afterwards. Since there are now probably several VMs to be moved at once, an appropriate result needs to be calculated, before AEQUO takes any actions because there might not be an optimal solution. The problem that needs to be solved is a bin packing problem [49]. Since this is a combinatorial NP-hard problem, the prototype uses a simplified approach based on the first-fit decreasing algorithm. This approach may not provide an optimal solution, but requires $O(n \log n)$ time instead of $O(n^2)$. If a solution was found, it will be applied to the cloud environment, meaning that all the VMs of a underloaded compute node will be moved to a new location. After each migration that the prototype instructs Nova to do, using the corresponding REST API, AEQUO will pause for about 2 seconds. This break is used to give Nova and libvirt enough time to process all the messages being sent to carry out the migration. In other environments, the required timespan for this pause might be different depending on how many VMs a compute node might have on an average and how long it takes to migrate them to another location. After all VMs have been migrated away from the selected compute node, the node can be hibernated, which is currently done by AEQUO using SSH.

All these calculations are scheduled periodically at fixed times. At the beginning all required data will be collected and analyzed. Depending on the results, appropriate actions will be executed and the process starts over again.

B. Optimizing the Energy Efficiency of Virtual Machine Placement and Network Connectivity in OpenStack Environments

As we already mentioned in Sections II-C and IV-B, there are also opportunities to reduce the energy consumption of the network components. Using AEQUO with its capability to monitor and control compute nodes, we currently prepare the infrastructure and graph database to extend our prototype to manage network devices. A possible scenario would be to completely power off a 19-inch rack, including all contained networking equipment like the ToR-Switch (top of rack) as well as the cooling for the rack. Therefore, it is necessary to make AEQUO aware of the components in each rack, and the energy consumption of these parts. This is necessary to support decisions, in which the entire load can be moved from a rack that could be subsequently shut down. At this point, we are evaluating to include asset/facility management or monitoring tools serving as an additional data source for AEQUO.

Another possibility to save energy is to shutdown redundant paths and network devices or links that are only needed at peak times. The devices could be powered off completely by using power distribution units (PDU) as mentioned in Section IV-A. Alternatively, some network devices (e.g., Cisco IOS routers or CatOS switches) have CLI support to power modules or ports up or down. To use these functions, AEQUO needs to be aware of the network structure, to decide what parts of the network can be powered off. As mentioned above, we are currently implementing a graph database as defined in Section IV-B. Instead of shutting down the links completely, network components that support Energy Efficient Ethernet (EEE), as described in Section II-C or techniques that control the power used by individual ports of the switch, could also be integrated, e.g., to throttle the link speed or enter EEE's low power idle mode. As described in Section II-C, the power reduction in this case comes with the drawback of increased latency, which has a negative impact especially on real-time applications. Hence, AEQUO can be used to temporarily turn on EEE and related mechanisms in the networking components when no real-time applications are used (e.g., less VoIP applications or video conferencing traffic during the night). Furthermore, the activation and deactivation of power management mechanisms can also be configured on redundant network paths, as illustrated in Figure 3.

C. Network Device Standby and Power Management

As described in Section II-C, existing network power management solutions could be enhanced to include mechanisms to power off network functions, links and components. The ElasticTree project [21] already referenced the possibility to introduce standby or sleep functions for networking hardware. As such standby power management functions were (and are still) missing in network equipment, [21] suggested to power down idle or underutilized switches completely. From our observations with network equipment from Cisco, HP and Arista, this approach has several drawbacks. For example, the Arista 7050S-52 and 7150S-24 switches (running Arista's Extensible Operating System (EOS) [50]) we used in our testbed in Section VI, take about 5 minutes to boot after being powered on again. Furthermore, powering down the entire switch is currently only possible using external power distribution units (PDUs) as described in Section IV-A. Since it is designed to run continuously, common professional networking equipment does not even have a power switch. As we used the two switches in a multi-chassis link aggregation (MLAG) setup, with each server being connected to both switches, we were able to power down one of the switches, e.g., during off-peak periods, without communication loss in the entire network.

Using external PDUs, the switch is unaware of being powered off, hence we observed minor communication disruptions (i.e., dropping some frames, causing TCP congestion control to reduce the bit rate, spanning tree topology changes due to suspected link flapping). Also, frequent power disruptions might impose risks to the software and hardware state of the switch, though we did not observe such effects in our tests. Furthermore, we simulated the switch power management by using virtualized network operating systems (namely, Arista's Virtual Extensible Operating System (vEOS) [50]) in a VMware vSphere environment. Using this environment, we were able to suspend the VMs running the virtual switches and resume them, to simulate a suspend-to-disk function in the switch. Again, only minor communication disruptions (as described above) were noticeable. Compared to the boot time of our Arista Hardware using EOS, the resume of virtualized EOS (vEOS) took only up to 5 seconds in our tests.

Since Arista is using a Linux Kernel in their products, we investigated further to evaluate the possibilities of a ”suspend to ram” solution. The EOS firmware is formed by a package of a patched Linux Kernel (3.4.43 for EOS 4.14.2F), an initial ramdisk, a root file system (squashfs) and a boot configuration. Arista offers the sources for the firmware and the contained packages on their website [50]. After modifying the sources and adjusting the parameters of our cross-compilation environment, we were able to compile a new kernel containing power management support (i.e., ACPI) and build a custom EOS 4.14.2F firmware. The kernel module for the Application Specific Integrated Circuits (ASICs) used in the 7150S (Intel FM6000 [51]) could not be compiled due to missing sources, hence some modules from the stock firmware were used. Using Arista's API, currently unused ports and their transceivers were successfully switched off (using 10GBase-SRL SFP+ this procedure saves ˜2 Watt/port). Supend-to-disk (ACPI S4) and ”suspend to ram” (ACPI S3) were not unusable, as the BIOS provided with the switch (Arista's Aboot-norcal2-2.0.9) does not fill the necessary ACPI tables. Also, watchdogs and custom patches introduced by Arista limited the power management and shutdown abilities of the switches. However, Arista uses the open source BIOS coreboot in its switches that does support filling the ACPI tables with S3 and S4 capabilities.

The hardware layout used in the 7150S switch consists of a general purpose motherboard (based on AMD Tilapia Fam10 reference design [52]) and a network switch ASIC (Intel FM6000) being connected via PCIe. Patches developed by Arista removed the ACPI capabilities of the coreboot firmware, though the AMD Tilapia Fam10 board supports them [53]. After modifying the supplied sources and adjusting our cross-compilation environment, we successfully compiled a new coreboot firmware for the switch. As the Tilapia board also includes the management network port, we were able to unload the ASIC (Intel FM6000) fpdma kernel module and use PCIe power management. However, significant power savings would require ”suspend to ram” support, which requires further modifications to the firmware (e.g., saving registers in non-volatile memory using AmdS3Save [53]). Another challenge is waking up the switch after a successful suspend. Wake-on-lan on the management network port of the general purpose motherboard is theoretically possible, e.g., combined with separate low-power proxy devices as introduced in [54].

Several papers presented theoretical approaches to implement power management functions in networking hardware (i.e., routers and switches), e.g., also using sleep, standby and rate-adaption techniques [55][56]. Some of them also focused on power management in LAN switches [57][58][59] combined with Energy Efficient Ethernet (EEE) in [60][61]. An interesting combination with the suspend functions we discussed above, might also be the migration of virtualized routers and switches as presented, e.g., in [62]. Energy-aware deployment of routers might also offer advantages to ensure connectivity across distant data centers, as shown in Figure 4.

Other recent related work shows the possibilities of employing power scaling mechanisms in custom built routers, e.g., based on NetFPGA cards [63]. We are currently evaluating to include such sleep and power scaling techniques in our testbed. An option would be to develop custom physical or virtual network switches (e.g., as described in [63]). The other option would be to test the integration of power management mechanisms in existing typical data center networking devices. Upcoming white box switches and open network operating systems (e.g., [64][65]) are offering new possibilities and less restrictions compared to not entirely open platforms like Arista's EOS. White box switches (e.g., [66][67]) also use merchant silicon ASICs for the data plane and typically common x86 architecture for general purpose CPUs and the control plane that include ACPI power management in their reference design. However, as described for the Arista devices above, these features are typically disabled or unused today.

Some network hardware providers already offer network power management frameworks (e.g., HP Adaptive Power Architecture [68], Cisco EnergyWise [5]). For example, Cisco EnergyWise can control EEE and hibernation functions of Cisco Catalyst 2960-X switches [69]. However, these solutions suffer the same problems as described above for our testbed. The switch can be put into hibernation using EnergyWise API or the CLI at the switch, but wake-up can only occur at a specific previously scheduled time or by manually pressing a button on the switch. While this is applicable for small offices or shops, e.g., at night or during non-office hours, traffic patterns of networks in private clouds are hard to predict and hence hibernation cannot easily be scheduled on a regular basis. This also holds true for multipath environments and redundant network devices, as described in Section II-C, as the performance requirements of private cloud networks typically cannot be foreseen in all cases. Therefore, besides the power management functions that are integrated in current networking devices (e.g., CPU frequency scaling, EEE), holistic power management frameworks that are able to toggle the power on temporarily unused or redundant links, custom ASICs or enable sleep or rate-adaption for an entire switch, are an upcoming challenge for continuously increasing bandwidths and the power consumption of today's network infrastructures as described in Section II-C.

Using our AEQUO prototype, as described in Section V-A, we can control the power of temporarily underutilized ports in the multipath network infrastructure shown in Figure 6. Additionally, AEQUO can issue CLI commands to the network switches that deactivate ASIC modules or use ACPI and PCIe power management (Active State Power Management (ASPM)) as presented in this section. Moreover, our AEQUO prototype can be combined with external network management and monitoring platforms (e.g., OpenNMS [70]) or data center infrastructure management (DCIM) solutions. For example, the network management system could inform AEQUO about a planned outage to ensure that all necessary redundant links and components are up, or AEQUO in turn could send information about current placement (e.g., across multiple sites, WAN links) of virtualized networking functions (e.g., routers, firewalls) to the network management system to reflect and monitor the changes in the network topology. Arista EOS and custom virtual network switches also support issuing CLI commands using OpenFlow. Therefore, AEQUO could also inform a central software-defined networking (SDN) controller instead of the network management system about its power management decisions. This way, the network could be dynamically adapted to ensure performance and fault-tolerance requirements within or across multiple data centers while also enhancing the energy efficiency of its links and devices.

VI. EXPERIMENTAL TESTBED

In the previous sections, we introduced procedures to automatically migrate virtual machines between data centers. This enables us to consolidate the virtual machines of an organization onto a minimum number of physical servers and to shutdown unneeded components. As a result, we are able to increase the energy efficiency of an organization's actively running IT infrastructure. However, the procedure of optimizing the virtual machine placement introduces load on the infrastructure and causes additional energy consumption. To get an idea of the energy impact of virtual machine migration and energy savings from shutting down physical components, a testbed has been set up at two sites in Germany. We have improved the initial test environment described in [1] with modern components, which are widely used in regular data centers, allowing us to simulate a typical cloud environment with independent compute, storage and networking components.

A simplified overview of the Fulda University site test setup is depicted in Figure 6. At the compute layer, the testbed consists of four identical Dell PowerEdge R620 servers (for reasons of clarity, only two compute nodes are shown in Figure 6), each equipped with two Intel Xeon E5-2650 processors and 256 GB of memory. A unified storage backend was built by utilizing two NetApp E2700 systems with a total of 48 SAS drives, which each compute node is connected to using an independent 16 Gbit/s fibre-channel link. The networking layer was built using the Arista datacenter switches 7050S-52 and 7150S-24, which provide 52 and 24 10-Gigabit-Ethernet ports, respectively. Each compute node is connected to both of the switches using a dedicated 10 Gbit/s fiber link. To setup a cloud environment on the described hardware the OpenStack Icehouse 2014.1.3 release on an Ubuntu 14.04.1 LTS platform was chosen. Two physical servers were used as dedicated compute nodes with Openstack Nova, whereas all other OpenStack components including block storage (Cinder), networking (Neutron), dashboard (Horizon), image (Glance), orchestration (Heat) and telemetry (Ceilometer) services are running on the two remaining servers, which are not shown in Figure 6 as described above. By this, we are able to perform measurements of VM migrations without side-effects introduced by the OpenStack infrastructure.

A primary requirement of our testbed is the feasibility to log detailed measurements of the individual components' power consumption. To meet this requirement, two Raritan PX2-5260R power distribution units (PDU) have been installed, each using an independent electric circuit and providing 12 separately measured power outlets to connect our equipment to. The measurement accuracy of our PDUs was verified by using a professional digital power meter of the type Yokogawa WT333. The discrepancies found were minimal and can be neglected. All components at the compute, storage and networking layers are equipped with redundant power supply units (PSU), which are connected to each of the PDUs. Further, we installed the network management system OpenNMS to continuously poll both PDUs to get the current power consumption of each power outlet. The resulting data gets stored in a round robin database (RRDTool) for further processing and graph generation.

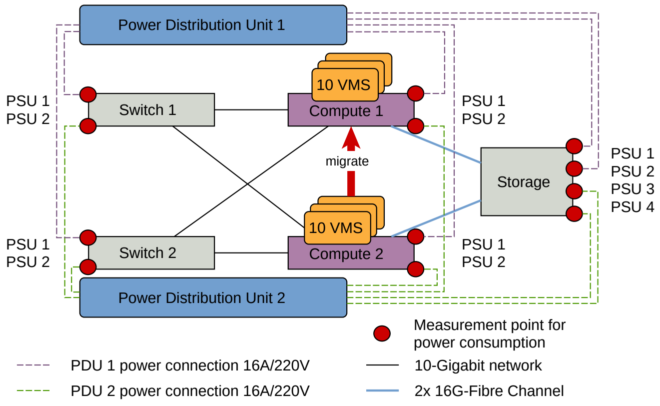

Figure 6. Test environment and network cabling.

The described testbed enables us to perform various test cases (i.e., VM migration, suspending physical components) under reproducible circumstances, as well as detailed measurement of the energy consumption of each component. In addition, another full-featured cloud environment has been set up at the Clausthal University of Technology, providing compute, storage and networking layers in a similar manner. The testbed is running OpenStack Icehouse as well and was connected to University Fulda as an OpenStack region for testing purposes. In the future, this setup will allow us to perform more realistic measurements of virtual machine migration, by taking much more metrics (e.g., WAN latency) into account.

VII. EVALUATION

Using the setup described in Section VI, we constructed a test case to get an idea of the impact of virtual machine (VM) migration and network interface suspension on the overall power consumption. The test case consists of multiple phases, which are executed automatically one after another, measuring specific characteristics of the overall power consumption.

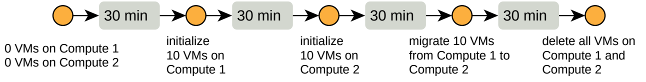

Figure 7. Testplan for measuring different load scenarios.

Figure 7 depicts the procedure of the test case, starting with both compute nodes in idle mode (no VMs running). At an interval of thirty minutes, first ten VMs on compute node 1 and another ten VMs on compute node 2 are spawned. Afterwards, all running VMs are migrated from compute node 2 to compute

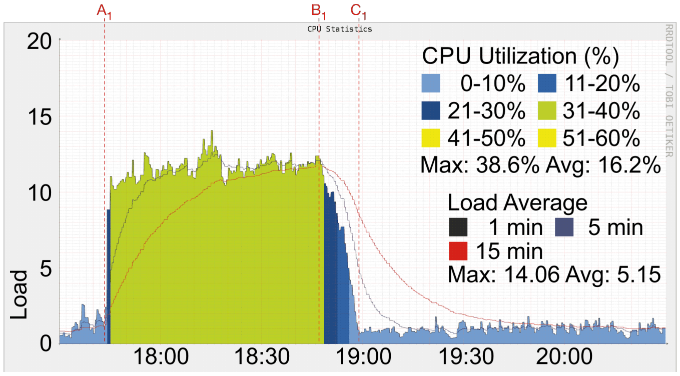

Figure 8. CPU utilization of compute node 1.

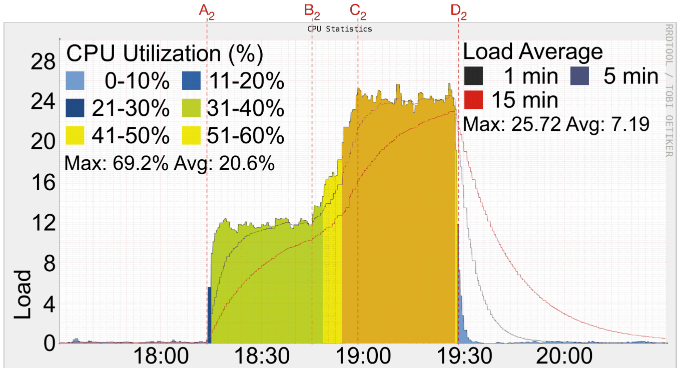

Figure 10. CPU utilization of compute node 2.

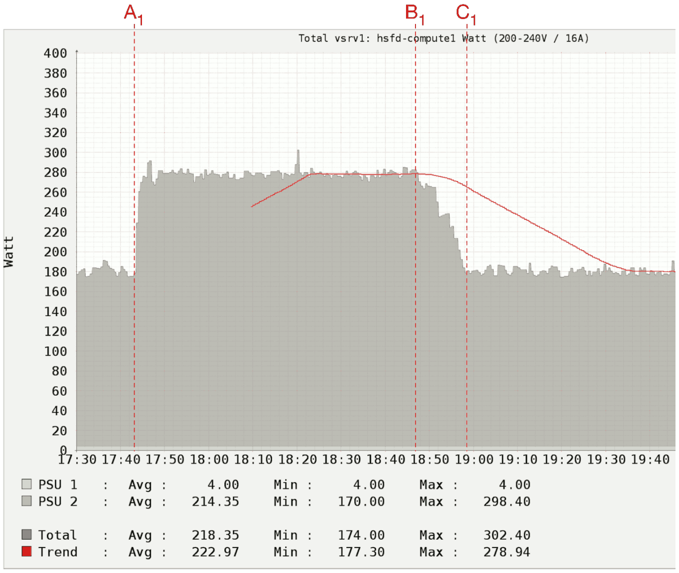

Figure 9. Power consumption of compute node 1.

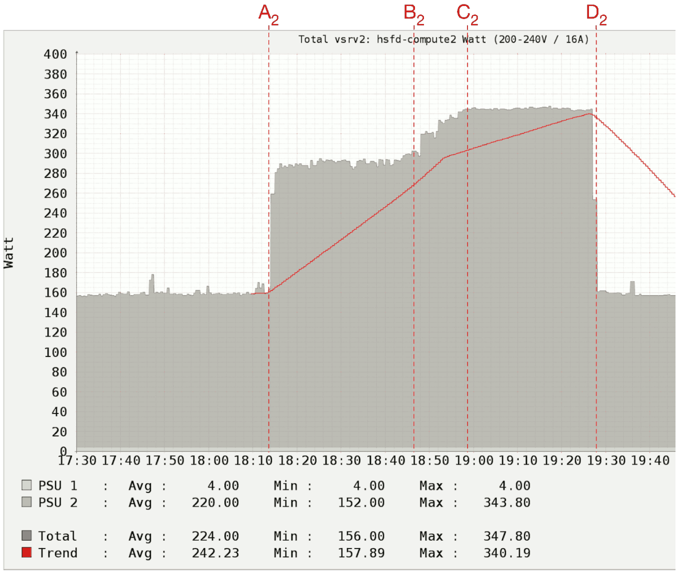

Figure 11. Power consumption of compute node 2.

node 1. Finally, all VMs are removed from both compute nodes. While the test was running, the CPU utilization of the compute nodes, as well as the power consumption of all of the involved components was continuously measured. Figure 6 shows the measurement points for power consumption in our testbed. Further, Figures 8 and 10 reveal the CPU utilization while Figures 9 and 11 show the power consumption for each of the compute nodes. Beside the CPU utilization the metric *Load* is measured to get an idea of the overall system capacity. Basically the *Load* metric is the moving average value of the number of queued processes waiting for execution. A *Load* value of 1.00 states that a system equipped with one CPU is running exactly at its capacity. A single CPU-system running at a *Load* of 2.00 has exactly the same number of processes queued as processes currently running. Of course, the *Load* metric also depends on the number of usable CPUs. Thus, a server equipped with four CPUs will reach the limits of its performance capability at a load value of 4.00.

The different phases of the test are denoted by red dotted lines in Figures 8 to 12. The first step of creating 10 VMs on compute node 1 is visualized in Figures 8 and 9 starting at mark A_1. The second step of creating the same number of VMs on compute node 2 is visible in Figures 10 and 11, starting at mark A_2. All VM instances created on the compute nodes are set up to use the *stress* utility to generate a consistent CPU load. To prevent running into system limits, the *stress* command is configured to generate a maximum of 40% of CPU utilization while keeping the overall system load below a value of 16 (the number of cores available) on each compute node. This leads to a CPU utilization of about 35% at the physical compute nodes, which is represented by different colors of the area in Figures 8 and 10. The effect of adding load to the compute nodes is also clearly visible in the corresponding power consumption graphs (Figures 9 and 11). While each of the compute nodes consumes about 155 Watt at the beginning of our test, the power consumption increases to about 290 Watt after initializing ten VMs (Figures 9 A_1 and 11 A_2). Our next step was the migration of all VMs from compute node 1 to compute node 2. The beginning of the migration is indicated by markers B_1 and B_2 and the completion is depicted by markers C_1 and C_2 in Figures 8 and 10, respectively. As all VMs are now running on compute node 2, the CPU of the node is now utilized by about 70%. After the last interval of thirty minutes, all VMs were shutdown, which is recognizable at mark D_2 in Figures 10 and 11.

By collecting power consumption metrics from all of the components, we are able to trace the weight distribution in the whole setup. This is needed to include the actual cost of the migration (e.g., additional CPU and networking load) in the overall power economization estimation. Figure 12 depicts the overall energy consumption of our testbed, divided into compute, storage and networking components. The total power consumption with ten VMs running on each compute node amounts to about 555 Watt and is shown starting at mark B. The migration of all VMs to compute node 2, which took around 10 minutes is recognizable between mark C and D. It led to an increased CPU utilization of about 75% and a system load of about 25 on compute node 2. There was no measurable impact on the energy consumption of the storage and networking components while the migration was ongoing. However, an interesting effect is visible regarding the total power consumption of the compute components after the migration phase. With 0 VMs on compute node 1 and 20 VMs on compute node 2, the total power consumption is around 525 Watt, which is about 30 Watt less, than in the case where the VMs were spread over both compute nodes. The effect is clearly visible in Figure 12, when comparing the power consumption of the compute components in phase B to C (ten VMs on each compute node) and phases D to E (twenty VMs on compute node 2). Of course, this effect is a result of components being more efficient when operating on higher load, which supports our idea of consolidating VMs on a minimum possible number of physical hosts.

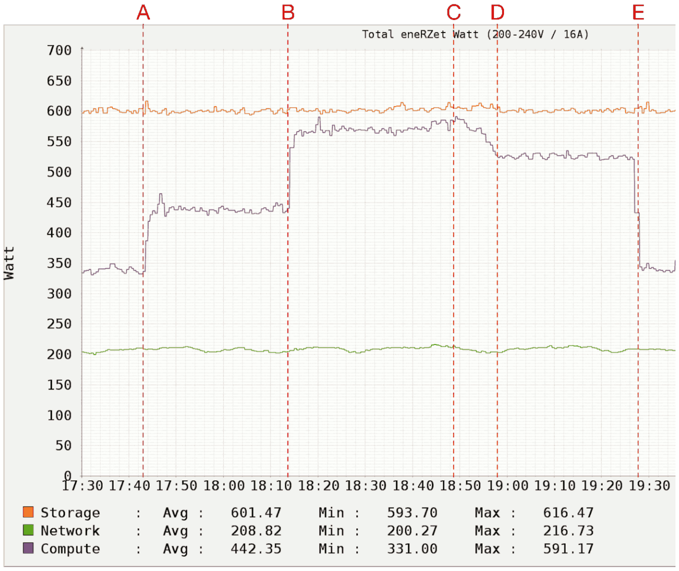

Figure 12. Total power consumption of the testbed setup.

Finally, the possibility of suspending one of the compute nodes was evaluated. For instance, in phase C_1 this would allow us to economize the power consumed by the compute node itself, as well as the power drawn by the networking interfaces it was using to connect to the switches and storage systems. The result of this is not visualized in our figures. However, one can see the effect by subtracting a compute nodes idle power consumption from the total value.

Our measurement of the testbeds energy consumption shed light on possible further enhancements of our prototype and identified non energy-proportional entities in our infrastructure that have a huge potential for energy savings. The AEQUO prototype provides the capabilities to address these promising components and builds the foundation for controlling the OpenStack services based thereupon. Compared the the related work presented in Section II-D, AEQUO offers a novel approach that is able to take different parameters (e.g., utilization, network connectivity requirements, temperature) into account for an energy-efficient placement of virtual resources in existing OpenStack environments. This way, new and upcoming compute and network power management techniques can be leveraged to implement distributed energy-aware private cloud infrastructures. Furthermore, currently available renewable energy resources at different data center sites or different energy prices, as described in Section III, can be used as a parameter for the optimization.

VIII. CONCLUSION AND FUTURE WORK

In this article, we presented an overview of our efforts to enhance the energy efficiency in OpenStack-based private cloud environments. Our prototype, called AEQUO, uses OpenStack's standard APIs to monitor the components used in our enterprise cloud testbed. As introduced, AEQUO uses OpenStack Nova and Ceilometer, to acquire the virtual machines (VMs) and hosts running in a private cloud environment together with their current CPU utilization. Furthermore, our prototype is able to control and schedule VM migrations to allow energy efficiency enhancements, i.e., by powering off unused or underloaded recourses. Using wattmeters in external power distribution units (PDU) we measured the power consumption while starting, migrating and shutting down VMs (instances) in our private cloud testbed.

The power measurements revealed as expected, that the power consumption of network and storage components that were used did not change during our tests, in contrast to the compute nodes (servers). This is interesting, because the process of migrating a VM from one host to another requires its virtual hard disk and RAM state to be transfered over the network. Additionally, the hard disk data needs to be persisted on the destination storage system. Therefore, network and storage components we used in our testbed must be classified as non energy-proportional components. Thus, there is potential for further research and optimizations. For example, the network switches in our testbed use merchant silicon instead of proprietary ASICs for the data plane, and standard x86 hardware for the control plane, which supports standard power management mechanisms (e.g., ACPI). Hence, we analyzed, whether we could modify the switch operating system to leverage standard x86 power management functionality. This way, we were able to deactivate unused links and also developed experimental extensions that support to suspend parts of the switch to improve the power management.

Another finding of our test runs was the fact, that there is no linear correlation between the number of VMs running on a compute node and its power consumption. In fact, it seems that the energy efficiency can be enhanced, if the VMs can be executed on a single host, rather than running the same number of VMs on several compute nodes. Of course, this only holds if the the CPU utilization has not already reached the physical maximum. However, in our tests the service quality offered by the virtual machines throughout the tests, which should be impacted by the corresponding consolidation ratio, was not monitored. This leads to the question whether a number of

VMs running on a single compute node can provide the same service quality (besides reliability concerns) compared to the same number of VMs running on several distributed compute nodes. An approach to investigate this aspect would be to use a web server benchmark utility to produce load and gather metrics for the service quality of web servers running in the VMs. Another possible topic for further research that we will focus on is the use of container virtualization instead of type-1 virtualization. In this case, containers are spawned or destroyed on demand without the need to transmit any hard disk or RAM contents. Typically, for example underlying type-1 VMs can provide persistent storage and database services to these containers. Also, the scheduling process of OpenStack could be improved to increase the energy efficiency. For instance, placing machines or affinity groups of dependent machines near their users can reduce network latency and improve user experience. As the prototype that we present in this article contains an extension to the OpenStack scheduler, it could be enhanced to support such an energy-efficient scheduling of new virtual resources (e.g., containers, VMs).

Compared to the related work described in Section II-D, our prototype can be easily integrated in OpenStack environments based on the current Juno release. Further techniques and scheduling algorithms or resource pooling mechanisms can be included in our prototype, thanks to its modularity as described in the implementation section of this article. Currently, our prototype is able to optimize the VM placement and utilization of compute resources in OpenStack environments. Furthermore, we focused on network paths including links and devices connecting the VMs to the network. While energy-efficient networks were also discussed in [21][9][39], our prototype is able to leverage existing and upcoming local power management techniques of compute and networking components (e.g., [9][19]). This way, for example, redundant links in the network can be throttled or even entire devices disabled when network and storage dependencies are integrated into the optimization.

Correspondent scenarios that enhance network and storage power management will be implemented in future testbed scenarios using our prototype. In this article, we already presented several concepts as a starting point to improve network power management. Regarding the storage components, apart from the included standby and power management of the hard disk drives, further investigation is needed. Additionally, our future work includes the evaluation of benefits from different energy prices and lower temperature at multiple sites, e.g., to reduce energy costs for cooling.

As described in this article, we evaluate the use of renewable energy resources for distributed data centers running private/ enterprise or hybrid cloud environments. By measuring the costs for migrations or scheduling of resources in cloud environments, we can evaluate the use of renewable energy at different data center sites or leverage varying electricity prices. As shown in this article, scheduling or migrating virtual resources between data centers in northern, central or southern Germany already has potential for energy efficiency enhancements on the basis of days. We are currently working on a simulation model to evaluate power consumption, migration costs and the use of available renewable energy at distant data centers. Upcoming frameworks for the acquisition of energy consumption metrics in OpenStack (e.g., KWAPI and IPMI extensions) that are currently under development also offer promising possibilities for further extensions to our prototype.

ACKNOWLEDGMENT

The authors would like to thank the Hessen State Ministry of Higher Education, Research and the Arts (HMWK) for partially funding the research presented in this paper within the ”Putting Research into Practice” program.

REFERENCES

[1] K. Spindler, S. Reissmann, and S. Rieger, “Enhancing the energy efficiency in enterprise clouds using compute and network power management functions,” in ICIW 2014, The Ninth International Conference on Internet and Web Applications and Services, 2014, pp. 134–139.

[2] P. Mell and T. Grance, “The NIST definition of cloud computing,” NIST special publication, vol. 800, no. 145, 2011, p. 7.

[3] C. Pape, S. Reissmann, and S. Rieger, “RESTful correlation and consolidation of distributed logging data in cloud environments,” in ICIW 2013, The Eighth International Conference on Internet and Web Applications and Services, 2013, pp. 194–199.

[4] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, Dec. 2008, pp. 68–73.

[5] S. S. Sandhu, A. Rawal, P. Kaur, and N. Gupta, “Major components associated with green networking in information communication technology systems,” in International Conference on Computing, Communication and Applications (ICCCA). IEEE, 2012, pp. 1–6.

[6] OpenStack, “OpenStack configuration reference - scheduling,” 2014, URL: http://docs.openstack.org/trunk/config-reference/content/section_compute-scheduler.html, 2015.05.23.

[7] A. Andreyev, “Introducing data center fabric, the next-generation Facebook data center network,” 2014, URL: https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/, 2015.05.23.

[8] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, “Energy consumption in optical IP networks,” Journal of Lightwave Technology, vol. 27, no. 13, 2009, pp. 2391–2403.

[9] T. Cheocherngngarn, J. H. Andrian, D. Pan, and K. Kengskool, “Power efficiency in energy-aware data center network,” in Proceedings of the Mid-South Annual Engineering and Sciences Conference, May 2012.

[10] K. Bilal et al., “A taxonomy and survey on green data center networks,” Future Generation Computer Systems, vol. 36, 2014, pp. 189–208.

[11] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, “Energy proportional datacenter networks,” in Proceedings of the 37th Annual International Symposium on Computer Architecture, ser. ISCA ’10. ACM, 2010, pp. 338–347.

[12] K. Bilal, S. U. Khan, and A. Y. Zomaya, “Green data center networks: Challenges and opportunities,” in Proceedings of the 2013 11th International Conference on Frontiers of Information Technology, ser. FIT ’13. IEEE Computer Society, 2013, pp. 229–234.

[13] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, “A power benchmarking framework for network devices,” in Proceedings of the 8th International IFIP-TC 6 Networking Conference, ser. NETWORKING ’09. Springer-Verlag, 2009, pp. 795–808.

[14] K. Bilal et al., “A survey on green communications using adaptive link rate,” Cluster Computing, vol. 16, no. 3, Sep. 2013, pp. 575–589.

[15] D. Valencic, V. Lebinac, and A. Skendzic, “Developments and current trends in ethernet technology,” in 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO). IEEE, 2013, pp. 431–436.

[16] P. Reviriego et al., “Energy efficiency in 10Gbps Ethernet transceivers: Copper versus fiber,” in Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC). IEEE, 2010, pp. 1–3.

[17] D. Larrabeiti, P. Reviriego, J. A. Hernández, J. A. Maestro, and M. Urueña, “Towards an energy efficient 10 Gb/s optical Ethernet: Performance analysis and viability,” Opt. Switch. Netw., vol. 8, no. 3, Jul. 2011, pp. 131–138.

[18] O. Haran, “Applicability of EEE to fiber PHYs,” 2007, URL: http://www.ieee802.org/3/eee_study/public/sep07/haran_1_0907.pdf, 2015.05.23.

[19] K. Christensen et al., “IEEE 802.3az: the road to energy efficient Ethernet,” Communications Magazine, IEEE, vol. 48, no. 11, 2010, pp. 50–56.

[20] Intel, “Energy efficient Ethernet: Technology, application and why you should care,” 2011, URL: https://communities.intel.com/community/wired/blog/2011/05/05/energy-efficient-ethernet-technology-application-and-why-you-should-care, 2015.05.23.

[21] B. Heller et al., “Elastictree: Saving energy in data center networks,” in Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI’10. USENIX Association, 2010, p. 17.

[22] A. Beloglazov, “Energy-efficient management of virtual machines in data centers for cloud computing,” Dissertation, Feb. 2013.

[23] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers,” in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, ser. CCGRID ’10. IEEE Computer Society, 2010, pp. 826–831.

[24] ——, “Openstack neat: A framework for dynamic consolidation of virtual machines in OpenStack clouds - A blueprint,” Technical Report CLOUDS-TR-2012-4, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, Tech. Rep., 2012.

[25] A. Song, W. Fan, W. Wang, J. Luo, and Y. Mo, “Multi-objective virtual machine selection for migrating in virtualized data centers,” in Pervasive Computing and the Networked World. Springer, 2013, pp. 426–438.

[26] N. A. Singh and M. Hemalatha, “Reduce energy consumption through virtual machine placement in cloud data centre,” in Mining Intelligence and Knowledge Exploration. Springer, 2013, pp. 466–474.

[27] A. Verma, P. Ahuja, and A. Neogi, “pMapper: Power and migration cost aware application placement in virtualized systems,” in Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, ser. Middleware ’08. Springer-Verlag New York, Inc., 2008, pp. 243–264.

[28] D. Versick and D. Tavangarian, “CAESARA - combined architecture for energy saving by auto-adaptive resource allocation,” in 6. DFN-Forum Kommunikationstechnologien, 2013, p. 31.

[29] X. Wang, X. Liu, L. Fan, and X. Jia, “A decentralized virtual machine migration approach of data centers for cloud computing,” Mathematical Problems in Engineering, vol. 2013, Aug. 2013.

[30] A. Beloglazov and R. Buyya, “Energy efficient allocation of virtual machines in cloud data centers,” in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, ser. CCGRID ’10. IEEE Computer Society, 2010, pp. 577–578.

[31] T. Carli, S. Henriot, J. Cohen, and J. Tomasik, “A packing problem approach to energy-aware load distribution in clouds,” CoRR, Mar. 2014.

[32] L. Shi, J. Furlong, and R. Wang, “Empirical evaluation of vector bin packing algorithms for energy efficient data centers,” Computers and Communications (ISCC), 2013 IEEE Symposium on, 2013, pp. 9–15.

[33] K. Craig-Wood and P. J. Krause, “Towards the estimation of the energy cost of Internet mediated transactions,” Energy Efficient Computing SIG Technical Report, 2013.

[34] V. C. Coroama, L. M. Hilty, E. Heiri, and F. M. Horn, “The direct energy demand of Internet data flows,” Journal of Industrial Ecology, vol. 17, no. 5, Oct. 2013, pp. 680–688.

[35] D. Hoyer and D. Schröder, “Untersuchung des Potentials von rechenzentrenübergreifendem Lastmanagement zur Reduzierung des Energieverbrauchs in der IKT,” 2009, English: “Study of the potential of using cross-data center load management to reduce energy consumption of ICT”.

[36] T. Huong et al., “ECODANE - reducing energy consumption in data center networks based on traffic engineering,” in 11th Würzburg Workshop on IP (EuroView2011), 2011.

[37] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, “CARPO: Correlation-aware power optimization in data center networks,” in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 1125–1133.

[38] V. Mann, A. Kumar, P. Dutta, and S. Kalyanaraman, “VMFlow: leveraging VM mobility to reduce network power costs in data centers,” in NETWORKING 2011. Springer, 2011, pp. 198–211.

[39] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, “VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers,” Computer Networks, vol. 57, no. 1, 2013, pp. 179–196.

[40] M. A. Adnan and R. Gupta, “Path consolidation for dynamic right-sizing of data center networks,” in Sixth International Conference on Cloud Computing (CLOUD). IEEE, 2013, pp. 581–588.

[41] The Green Grid, “Carbon usage effectiveness (CUE): A green grid data center sustainability metric,” 2010, URL: http://www.thegreengrid.org/∼/media/WhitePapers/CarbonUsageEffectivenessWhitePaper20101202.ashx, 2015.05.23.

[42] The Umweltbundesamt, “New ”blue angel” for data centers,” 2015, URL: http://www.umweltbundesamt.de/en/press/pressinformation/new-blue-angel-for-data-centers, 2015.05.23.

[43] B. Valov, “Outlook in the future of german north sea 2020 - power system for offshore wind power,” Wind-Kraft Journal, German Offshore, vol. 5. Auflage, 2008, pp. 38–39.

[44] M. Sterner et al., “Renewable (power to) methane: Storing renewables by linking power and gas grids,” Electricity Storage Association 20th Anniversary Conference, Charlotte, North Carolina, USA, 2010, 2010.05.06.

[45] J. Ringelstein and D. Nestle, “Application of bidirectional energy management interfaces for distribution grid services,” in Electricity Distribution-Part 1, 2009. CIRED 2009. 20th International Conference and Exhibition on. IET, 2009, pp. 1–4.

[46] L.-A. Brischke, M. Hoppe-Kilpper, and A. Tiedemann. Grid integration in Germany. URL: http://www.wwindea.org/technology/ch04/en/4_1_2_2.html, 2015.05.23.

[47] Agentur für Erneuerbare Energien, “www.foederal-erneuerbar.de,” 2015, URL: http://www.foederal-erneuerbar.de/uebersicht/bundeslaender/BW|BY|B|BB|HB|HH|HE|MV|NI|NRW|RLP|SL|SN|ST|SH|TH|D/kategorie/solar/auswahl/183-installierte_leistun/#goto_183, 2015.05.23.

[48] ——, “www.foederal-erneuerbar.de,” 2015, URL: http://www.foederal-erneuerbar.de/uebersicht/bundeslaender/BW|BY|B|BB|HB|HH|HE|MV|NI|NRW|RLP|SL|SN|ST|SH|TH|D/kategorie/wind/auswahl/180-installierte_leistun/#goto_180, 2015.05.23.

[49] M. Sindelar, R. Sitaraman, and P. Shenoy, “Sharing-aware algorithms for virtual machine colocation,” in Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures, 2011, pp. 367–378.

[50] Arista, “Arista EOS central,” 2015, URL: https://eos.arista.com/, 2015.05.23.

[51] Intel, “Intel ethernet switch fm5000/fm6000 series,” 2015, URL: http://www.intel.com/content/www/us/en/switch-silicon/ethernet-switch-fm5000-fm6000-series.html, 2015.05.23.

[52] Coreboot, “Coreboot tilapia fam10,” 2015, URL: https://github.com/coreboot/coreboot/tree/master/src/mainboard/amd/tilapia_fam10, 2015.05.23.

[53] ——, “Coreboot AMD S3,” 2015, URL: https://raw.githubusercontent.com/coreboot/coreboot/master/Documentation/AMD-S3.txt, 2015.05.23.

[54] S. Nedevschi et al., “Skilled in the art of being idle: Reducing energy waste in networked systems,” in NSDI, vol. 9, 2009, pp. 381–394.

[55] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, “Reducing network energy consumption via sleeping and rate-adaptation,” in NSDI, vol. 8, 2008, pp. 323–336.

[56] M. Gupta and S. Singh, “Greening of the Internet,” in Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ser. SIGCOMM ’03. ACM, 2003, pp. 19–26.

[57] G. Ananthanarayanan and R. H. Katz, “Greening the switch,” in Proceedings of the 2008 Conference on Power Aware Computing and Systems, ser. HotPower’08. USENIX Association, 2008, p. 7.

[58] M. Gupta and S. Singh, “Using low-power modes for energy conservation in Ethernet LANs,” in INFOCOM 2007. 26th IEEE Interna-

tional Conference on Computer Communications. IEEE, May 2007, pp. 2451–2455.

[59] M. Gupta, S. Grover, and S. Singh, “A feasibility study for power management in LAN switches,” in Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on, Oct 2004, pp. 361–371.

[60] W. Yang, J.-H. Jung, and Y.-C. Kim, “Performance evaluation of energy saving in core router architecture with low power idle for OBS networks,” in Information Networking (ICOIN), 2012 International Conference on, Feb 2012, pp. 318–323.

[61] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, “Reducing the energy consumption of Ethernet with adaptive link rate (ALR),” IEEE Trans. Comput., vol. 57, no. 4, Apr. 2008, pp. 448–461.

[62] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, “Virtual routers on the move: Live router migration as a network-management primitive,” in Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, ser. SIGCOMM ’08. ACM, 2008, pp. 231–242.

[63] N. P. Ngoc et al., “A new power profiling method and power scaling mechanism for energy-aware NetFPGA gigabit router,” Computer Networks, 2014.

[64] Open Networking Lab (ON.LAB), “ONOS - open networking operating system,” 2014, URL: http://onosproject.org/, 2015.05.23.

[65] Cumulus Networks, “Cumulus Linux,” 2015, URL: http://cumulusnetworks.com/, 2015.05.23.

[66] Facebook, “Introducing ”Wedge” and ”FBOSS,” the next steps toward a disaggregated network,” 2014, URL: https://code.facebook.com/posts/681382905244727/introducing-wedge-and-fboss-the-next-steps-toward-a-disaggregated-network/, 2015.05.23.

[67] Open Compute Project, “ONIE hardware status,” 2015, URL: http://www.opencompute.org/wiki/Networking/ONIE#Testing_and_Certification, 2015.05.23.

[68] Hewlett-Packard Development, “Energy efficient networking,” 2011, URL: http://h17007.www1.hp.com/docs/mark/4AA3-3866ENW.pdf, 2015.05.23.

[69] Cisco, “Cisco Catalyst 2960-X platform: The greenest Catalyst switches ever,” 2013, URL: http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-x-series-switches/white_paper_c11-728857.html, 2015.05.23.

[70] OpenNMS Group, Inc., “OpenNMS management application platform,” 2015, URL: http://www.opennms.org/, 2015.05.23.