Enhancing the Energy Efficiency in Enterprise Clouds Using Compute and Network Power Management Functions

Kai Spindler, Sven Reissmann, Sebastian Rieger
Department of Applied Computer Science
University of Applied Sciences Fulda
Fulda, Germany
{kai.spindler, sven.reissmann, sebastian.rieger}@cs.hs-fulda.de

***Abstract*—Enterprise cloud infrastructures and virtualization technologies constitute a growing proportion in today's data centers. For these data centers the ongoing operational costs are not negligible, especially for electricity which is also increased by cooling. Solutions that raise the energy efficiency allow to reduce these operational costs and to optimize the utilization of the data center infrastructure. The following paper presents a solution to optimize the energy efficiency by observing the current utilization parameters of compute resources and network devices and by taking appropriate actions based on this data. This optimization will be carried out by an automated instance with a comprehensive view on the data center assets, which is relocating virtual machines and optimizing the network structure. The paper presents a lightweight prototype that can be integrated in enterprise cloud environments using standard OpenStack components and application programming interfaces. By monitoring the energy consumption of resources in the environment and combining state of the art in energy-efficient cloud computing with upcoming power management techniques for compute, storage and especially network resources, new possibilities to increase the energy efficiency in enterprise clouds are introduced.**

Keywords–Enterprise Clouds; OpenStack; Energy Efficiency; Computer Networks; Power Management.

I. INTRODUCTION

Enterprise or private 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 usage 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 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 a 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 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 enterprise 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.

This paper presents a solution to enhance the energy efficiency in OpenStack-based enterprise cloud environments. 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]. The paper presents 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 and 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 enterprise 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. Requirements for our prototype, to enhance the energy efficiency by combining the state of the art techniques and extending them, are defined in Section III. The implementation of our prototype and mechanisms to optimize the energy efficiency in enterprise clouds are presented in Section IV. Finally, Section V 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. Additionally, the state of related research projects is discussed.

A. OpenStack-based Enterprise 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 (e.g., 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 is offering 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 or to start or stop VMs depending on the current load is possible during the lifecycle of a service. This flexibility is providing 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 Enterprise Clouds

In todays rapidly growing IT infrastructures, energy efficiency is no longer a secondary requirement, but rather has become one of the main objectives when planning and operating new data centers. A reason for this development is the common sensitization for an ecologically sustainable use of global resources. Furthermore, large-scale data centers are consuming 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 supply [3]. 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 todays data center operators, which have to be economical while facing increasing energy costs [4]. It can be said that cloud computing by definition leads to energy efficiency through its operation 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 [2], therefore instant availability of additional resources is required.

The energy consumption of a VM running in OpenStack mainly depends 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., router, switch), but also on the distance of the involved components (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 operation costs are lower. In OpenStack, the placement of VMs on a specific cloud computing fabric controller (Nova) is mainly determined by nova-scheduler [5]. While offering several techniques 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 to secondary resources, such as the availability of a specific storage or network capabilities. Moreover, the Filter Scheduler addresses the operational requirements for resilience or consolidation of VMs by explicitly allowing its 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 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 is providing 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. According to [6], 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 high bandwidth, which leads to increasing link speeds and therefore rises the power consumption of each switch port. Additionally, redundant links are required to assure resilience of the network, again increasing the power consumption. Concepts like Equal Cost Multipathing or Multipath TCP are available to utilize the equipment up to its capacity. However, variable bandwidth requirements (e.g., decreased usage during nighttime) makes it economically reasonable to scale down the network as well. For wired local area networks (LAN), which we primarily focus on, there are already some power management techniques being offered by the vendors of networking components. First and foremost, the LAN standard 802.3 was extended to include 802.3az, also called energy-efficient ethernet (EEE) [7]. Since this extension is part of the regular 802.3-2012 standard, it is likely that in the near future all equipment will support EEE.

While EEE manufacturers claim that 802.3az allows a reduction of the energy consumed by a single port by up to 81% [8], this benefit comes with the price of increased latency during the low power idle (LPI) phase [9]. Regarding the fact that currently data center network infrastructures are moving to 10 Gbit/s ethernet and beyond, where power consumptions per port are usually over 5 Watts [8], the power savings for the entire data center infrastructure are even higher. Furthermore, there are other vendor-specific power management functions of networking components (i.e., Cisco EnergyWise [4]) that are not covered by EEE. Compared to power management functions of compute and storage resources (e.g., APM, ACPI), that have constantly evolved over the last decades, power management functions for network components are relatively new and will supposedly be improved due to energy efficiency requirements in the near future.

All of the existing solutions are able to reduce the local power consumption on individual network components and ports, but they are unaware of the current global requirements in the entire network. Therefore, their scope is rather limited and the energy efficiency optimization is rather isolated. Some research projects, notably Stanford's ElasticTree have identified this problem, but did not integrate it with an appropriate placement of VMs and especially did not discuss the requirements of enterprise clouds [10]. By using a network controller that is aware of the entire topology, such links could be disabled or throttled during off-peak times while still maintaining fault tolerance requirements. Moreover, such a controller could also activate and deactivate entire networking components based on the current requirements to enhance the energy efficiency. These assumptions and 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 [11][12][13]. However, these approaches do not consider an optimal placement of the VMs with respect to temperature, cooling and network connectivity requirements. Additionally, the extensions presented in these papers cannot be used with the current Havana Release of OpenStack. Furthermore, an integration of additional custom criteria for scheduling decisions regarding the optimal placement of VMs is not supported. A more generalized and detailed evaluation of an energy-efficient placement of VMs in cloud environments and relevant parameters is given in [14][15]. However, these contributions do not offer testbeds for OpenStack environments. Common factors and algorithms to estimate the energy demand of VMs and their migration are discussed in [16].

Concerning energy-efficient computer networks, especially the ElasticTree project [10] 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 [17] to include traffic engineering. Also, theoretical energy-aware optimizations of data center networks were presented in [18][6]. Requirements and constraints for energy-efficient placement of VMs regarding the network connectivity were explored in [19][20][21]. However, these solutions do not include existing power management techniques like we described for networking resources (e.g., [6][8][9]) in the previous sections. Furthermore, these approaches do not include power management functions like the Advanced Configuration and Power Interface (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. 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 enterprise clouds.

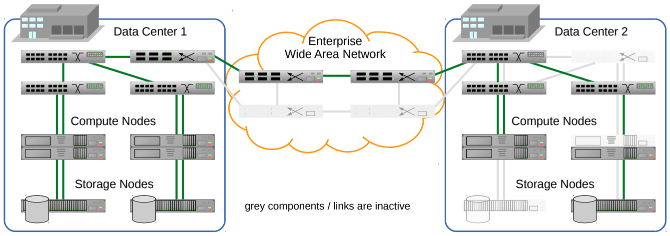

Figure 1. Power management for energy-efficient compute, storage and networking resources in enterprise clouds.

Figure 1 shows an example of an enterprise 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 enterprise 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 1.

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

As introduced in Section II-A, OpenStack is not by itself able to manage resources with respect to the energy efficiency. Therefore, we present concepts supporting the decision about when and how resources like VMs can be relocated to increase the energy efficiency while respecting 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 are general resource informations, 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 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 global metric we identified to be interesting to evaluate whether it makes sense to move VMs from one data center to another would be the current local electricity price at a specific site.

Having 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. More important, however, it is possible to shutdown the servers switchport to reduce the energy consumed by the network as mentioned in Section II-C. 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 respectively. 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 networking equipment could be powered on and off in a controlled way, if an appropriate mechanism exists to optimize the energy consumptions based on the strategy discussed in this section.

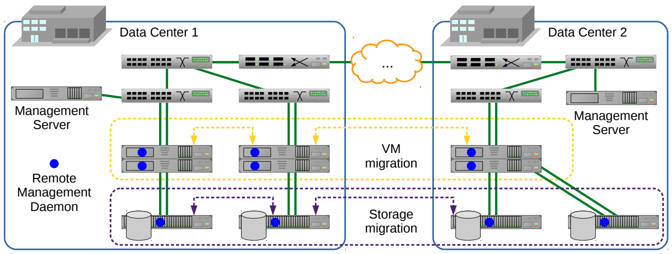

Figure 2. Integration of power management components to enable energy-efficient compute, storage and networking.

As shown in Figure 2, we introduce dedicated management servers in each data center, which have a global view over all servers in the data center. Additionally, management data from other data centers can be synchronized to have the same global knowledge. Each of the management servers is collecting data from the compute, storage and networking nodes in the data center using remote management daemons. Based on this data, they decide when to move the VMs by instructing the involved hypervisors to start a live migration process.

B. Energy-efficient Network Connectivity in OpenStack Environments

The complexity of computer networks with respect to energy consumption can be reduced to consist of nodes and links (Figure 1). Regarding the energy efficiency of a network, two factors driving the energy consumption can be identified. First and foremost, the energy requirements are defined by the number of nodes and links. This especially includes power dissipation at each component. Second, the utilization of each node and each link influences its individual energy consumption. The higher the utilization, the more energy is needed for each component. Nonetheless, a sufficient utilization of all links and components leads to increased efficiency. From a theoretical point of view, the network builds a graph, with each edge representing a link. 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, it is possible to model the topology of a network and apply the metrics described above. The network connection of a VM is 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 that can be turned on or off depending on the current utilization of the active links or the requirements to resilience. Hence, graph databases can be used to support the decision for energy-efficient placement and 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., in off-peak times, or reactivated based on network utilization.

IV. ENHANCING THE ENERGY EFFICIENCY OF VIRTUAL MACHINES IN ENTERPRISE 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 Section III.

A. AEQUO Testbed Based on OpenStack

For our proof of concept we used Rackspace Private Cloud [22], which provides a fast way to deploy an OpenStack environment with all its components. The installation and configuration of the components is done by Chef, which uses so called cookbooks to deploy the OpenStack services. Our proof of concept uses three virtual machines. Two of them are used as dedicated compute nodes while the other one is used as a hypervisor hosting the rest of the infrastructure. The hypervisor has two VMs running, one serves as the Nova Controller and includes block storage (Cinder), networking (Neutron), dashboard (Horizon), image service (Glance) and orchestration (Heat) components. The other machine serves as the Chef server, which is used for deploying all services during the installation process and later on for adding new components, which makes the system easily expandable. All virtual machines are set up using Ubuntu 12.04 LTS operating system. To be able to move the VMs from one compute node to another during operation, it is necessary to install a shared storage on the nova controller and all nova compute nodes. The shared storage, which is realized using the distributed scale-out file system Gluster [23], is also used by AEQUO to exchange management information regarding the individual node.

B. 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 three components. First is the Collector Daemon that runs on each of the compute nodes and is responsible for accumulating performance data, like temperature or energy consumption. In our current implementation we are primarily evaluating the temperature because it is easy to collect for this early approach. The two other components are running on the Nova controller, whereby the Aggregator Daemon is collecting the data received from the Collector Daemon. Additionally, the Aggregator Daemon is writing its data into an SQLite database. Finally, our third component is the Balancing Daemon, which is querying the data from the SQLite database to evaluate it. This historical data is included in the process of making decisions whether or not to move a VM. Figure 3 illustrates AEQUO's components.

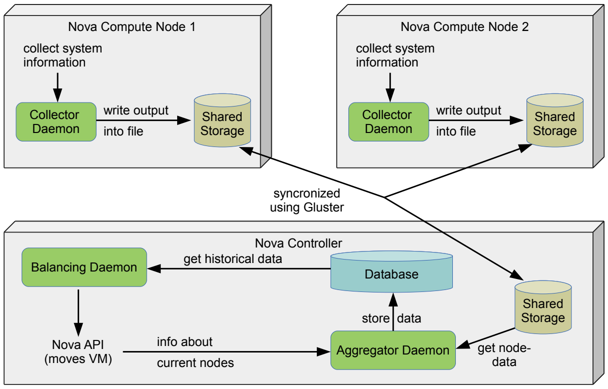

Figure 3. Components and architecture of AEQUO.

In the current version of our prototype, the database consists of three tables, which we illustrate in Table I. The measure table contains historical data from the monitored compute nodes. The table nodedata contains meta information about the compute nodes. Finally, the table vmmove keeps a log providing information about VM movements.

The Collector Daemon running on the compute nodes does not need any configuration, due to the fact that it is just collecting data and writes it to the shared storage. Aggregator Daemon and Balancing Daemon are implemented in a single Python script, as they run simultaneously on the Nova Controller. Using different arguments, the Python script can either be started to run the daemons, test whether the system is running properly or to show the latest data collected. Additionally, an option is offered to insert meta information regarding newly added compute nodes. When started, the script first checks whether the database already exists or needs to be created from scratch. Considering that a significant part of electrical energy consumed by computing resources is transformed into heat [11], our current implementation uses simple thermal thresholds over a certain time to decide whether VMs should be moved. However, the prototype can easily be extended to include sophisticated algorithms, e.g., as presented in Section II-D.

TABLE I
DATABASE STRUCTURE USED BY AEQUO.

Table: nodedata

Field	**Type**	**Description**
hostname	TEXT	hostname of the compute node
ip	TEXT	IP address of the compute node
status	TEXT	status of the node

Table: measure

Field	**Type**	**Description**
hostname	TEXT	hostname of the compute node
time	INTEGER	timestamp of the data
temp1	REAL	temperature of the compute node

Table: vmmove

Field	**Type**	**Description**
hostname	TEXT	hostname of the compute node
time	INTEGER	timestamp
moved_from	TEXT	source
moved_to	TEXT	destination

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

As we already mentioned in Sections II-C and III-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. One 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 is moved from a rack and it is subsequently shut down. At this point, we are evaluating to include asset or 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) like mentioned in chapter III-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 III-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 usage of 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 1.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a lightweight prototype to enhance the energy efficiency in enterprise cloud environments that uses standard OpenStack APIs and components. As a starting point, the prototype monitors the temperature and cooling of the components in our cloud testbed, allowing thermal-aware scheduling and migration of virtual machines. Compared to the related work described in Section II-D, our prototype can easily be integrated in OpenStack environments based on the current Havana release. Hence, it serves as a testbed in our research project to evaluate different strategies and state of the art research findings dealing with the energy efficiency in private cloud environments like [14][15]. These techniques can easily be integrated in our prototype thanks to its modularity as described in the implementation section of this paper. On the one hand, we are optimizing the placement and usage of compute and storage resources in OpenStack environments. On the other hand, we focused on network paths including links and devices connecting the virtual machines to the network. While energy-efficient networks were also discussed in [10][6][20], we built our prototype to leverage existing and upcoming local power management techniques of compute and networking components (e.g., [6][8]). 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. We will implement a correspondent scenario in our testbed using our prototype. As a next step of our research, we will measure and evaluate the power savings using the compute, storage and networking equipment in our OpenStack testbed, including models to calculate the cost for virtual machine live-migration [16]. Furthermore, our future work includes the evaluation of benefiting from different energy prices and lower temperature at multiple sites, e.g., to reduce energy costs for cooling. Additionally, we will evaluate the integration of the mechanisms we developed in OpenStack's orchestration framework Heat and the monitoring of energy efficiency metrics in OpenStack's Ceilometer.

ACKNOWLEDGMENT

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

REFERENCES

[1] 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.

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

[3] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, 2008, pp. 68–73.

[4] 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.

[5] OpenStack, "OpenStack Configuration Reference - Scheduling," 2014, URL: http://docs.openstack.org/trunk/config-reference/content/section_compute-scheduler.html, 2014.05.26.

[6] 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.

[7] 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.

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

[9] Intel, "Energy efficient ethernet: Technology, application," 2011, URL: https://communities.intel.com/community/wired/blog/2011/05/05/energy-efficient-ethernet-technology-application-and-why-you-should-care, 2014.05.26.

[10] B. Heller et al., "ElasticTree: Saving Energy in Data Center Networks." in NSDI, vol. 3, 2010, pp. 19–21.

[11] A. Beloglazov, "Energy-efficient management of virtual machines in data centers for cloud computing," Dissertation, Feb. 2013, URL: http://repository.unimelb.edu.au/10187/17701, 2014.05.26.

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

[13] ——, "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.

[14] 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.

[15] 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.

[16] D. Versick and D. Tavangarian, "CAESARA - Combined Architecture for Energy Saving by Auto-Adaptive Resource Allocation," in 6. DFN-Forum Kommunikationstechnologien, 2013, p. 31.

[17] T. Huong et al., "ECODANEreducing energy consumption in data center networks based on traffic engineering," in 11th Würzburg Workshop on IP (EuroView2011), 2011.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Rackspace, "OpenStack Private Cloud Software," 2014, URL: http://www.rackspace.com/cloud/private/openstack_software/, 2014.05.26.

[23] RedHat, "GlusterFS," 2014, URL: http://gluster.org/community/documentation/index.php/OSConnect, 2014.05.26.