Energy-efficient Live Migration of I/O-intensive Virtual Network Services Across Distributed Cloud Infrastructures Leveraging Renewable Energies

Ngoc Khanh Truong*, Christian Pape*, Sven Reißmann†, Thomas Glotzbach‡ Sebastian Rieger*
*Department of Applied Computer Science
Fulda University of Applied Sciences, Fulda, Germany
{ngoc.k.truong, christian.pape, sebastian.rieger}@cs.hs-fulda.de
†Datacenter
Fulda University of Applied Sciences, Fulda, Germany
sven.reissmann@rz.hs-fulda.de
‡Department of Electrical Engineering and Information Technology
Hochschule Darmstadt University of Applied Sciences, Darmstadt, Germany
thomas.glotzbach@h-da.de

Abstract—**Virtual infrastructures and cloud services became more and more important over the past years. The abstraction from physical hardware offered by virtualization supports an increased energy efficiency, for example, due to higher utilization of underlying hardware through consolidation. Also, the abstraction enables the ability to geographically move cloud services, e.g., to be able to benefit from lowest available energy prices and renewable energy. This article gives an overview on such migration techniques in distributed private cloud environments. The presented OpenStack-based testbed is used to measure migration costs along with the service quality of virtualized network services. Correspondingly, the article illustrates the impact of high memory and input/output (I/O) load on live migrations of network services and evaluates possible optimization techniques. The results gained from the experiments presented in this article, can be used to evaluate whether network services and virtual resources can be migrated to distant sites to reduce energy costs. A potential benefit of such migrations can be to leverage from fluctuating renewable energies across multiple data center sites. Possible improvements as well as side effects of this use case are presented in the evaluation regarding the live migration of virtual network services. Regarding virtual network services, potential drawbacks can result from additional latency when maintaining and using the virtual services across distant locations. To mitigate these effects, the article describes a way to identify dependencies and affinities between virtual and physical resources based on network flow data. The evaluation used a data set of characteristic networks flows from around one hundred virtual machines of the production environment at Fulda University of Applied Sciences. While respecting these requirements and dependencies, the optimization described in this article used weather data of multiple years of three different distant locations in Germany. Possible improvements of the utilization of renewable energies due adaptive placement and migration of virtual resources were evaluated using this data. Together with the detailed evaluation of the costs of these migrations, which especially rise for I/O-intensive migrations, e.g., for virtual network services, the results of this article can be used to increase the overall energy efficiency of data centers in distributed cloud infrastructures.**

Keywords*–*Cloud Computing; Network Services; Live Migration; Energy Efficiency; Renewable Energy.

I. INTRODUCTION

A solution for energy-efficient live migrations of I/O-intensive virtual network services across distributed cloud infrastructures was presented in [1]. In this article, these findings will be elaborated and their benefit for the use of renewable energy (RE) sources between distant data centers while limiting possible drawbacks of distributed virtual resources, e.g., due to resource dependencies and associated affinity groups, will be explained. Energy costs are an important factor for data centers and IT infrastructures as a whole. Drivers for the increasing costs over the last decade have been electricity prices, but also the growing energy demand of data centers and IT infrastructures. Regarding the electricity price, the changes in national energy policies to move from low-priced conventional, e.g., nuclear, power to renewable energies (e.g., in the European Union and especially in Germany), augur that energy costs will increase even further. While the percentage of the costs for network equipment and services have been negligible for data centers in the past, this is likely to change due to increased bandwidth and the steadily increasing number of network devices, amplified by the evolving "Internet of Things" and cloud-based services. Recent papers even state that the network power consumption could grow beyond 25% [2][3] of the total data center energy demand. This is especially likely for large data centers (i.e., Google, Amazon, Facebook), whose inner data center traffic is quickly increasing [4]. Since virtualization is used for compute, storage and network resources in modern data centers, these infrastructures support automatic provisioning and management of virtual resources, which can be used to optimize the energy efficiency. For example, virtual resources can be consolidated to reduce the required hardware based on the current load. During off-peak hours, resources and links can be powered down or use power management, while being quickly and automatically reactivated on demand. This also allows for elastic scalability [5], as well as adaptive scheduling, placement and migration of virtual resources. The scheduler can consider electricity prices and the availability of RE resources across multiple data centers [6]. Hence, an energy- and cost-efficient adaptive placement of virtual resources can be attained.

Regarding the network, cloud environments typically employ network virtualization to implement networking functions for their delivered services. These virtualized network services offer transparent use and flexibility regarding the underlying resources. Such services, e.g., in the form of virtual network functions (VNF), are not only getting more and more momentum in service provider networks (as defined, e.g., in the network functions virtualization (NFV) reference model of the ETSI [7]), but also in virtualized network infrastructures as a whole. While the "on-demand self-service" and "rapid elasticity" paradigms [5] of cloud and software-defined infrastructures imply that virtual resources, including virtualized network services, can quickly be spawned and destroyed, not all use cases of virtual network services fulfill such a "pets versus cattle" approach [8] for cloud resources. Spawning and stopping new VMs or containers behind a load balancer for example, is easy to implement, while the migration of entire clusters of services and load balancers across cloud infrastructures, without losing network connectivity (e.g., sessions, traffic flows) requires special techniques. Virtualization and cloud environments typically allow for a transparent migration of virtual resources across different underlying hardware components (e.g., implementing a "live migration" technique). Hence, network services impose special requirements for live migrations. The network load, e.g., on VNFs, is typically higher than on back end servers, due to their function as a front end for multiple services or servers. This leads to a high I/O rate of the virtual machines (VMs) and containers offering such virtual network services (e.g., VNF). Sometimes, these I/O-intensive memory and network operations are therefore enhanced by using special acceleration functions of the underlying hardware, i.e., TCP offloading or single-root I/O virtualization (SR-IOV), which also hold specific constraints for live migrations, due to the fact that they are depending on local physical hardware (i.e., network interface cards).

In this article, an analysis of the impact of these implications for the migration of virtual network services across distributed cloud environments is presented. Our experimental approach uses an OpenStack-based testbed migrating virtual network services under load and evaluating the results. Additionally, techniques to improve the energy efficiency of the migration are discussed. By using a live migration, the services can be transferred seamlessly during operation instead of interrupting existing connections leading to additional energy being required to reestablish lost connections. However, the energy consumption of the migration itself needs to be optimized (e.g., limiting resources and time needed for the migration).

The migrated virtual network services can include typical NFV or SDN components (e.g., virtual switches, controllers). Regarding the energy efficiency, again the stated "pets versus cattle" paradigm cannot be applied to every migration of virtual network services. Besides the negative effect of the downtime while respawning, e.g., VNFs, also more energy is consumed, if components like switches, firewalls or load balancers are simply destroyed and lost connections or flows need to be reestablished, increasing the load on the underlying cloud infrastructure. Furthermore, more energy is consumed if constraints like placing the network functions close to back end servers etc. are not satisfied (e.g., due to resource dependencies or "affinity groups" of virtual resources). Hence, the energy consumption of the migration itself needs to be optimized (e.g., limiting the resources and time needed for the migration). This article also includes an outlook on further improvement by using containers and microservices, reducing the effort for live migrations regarding the state and data that needs to be transferred. The migration techniques for virtual network services can also be combined with upcoming network and server power management, to increase energy efficiency and power savings even further.

A potential beneficial use case for energy-efficient live migrations is the combination with RE sources for data centers, as already discussed, e.g., in [9]. This includes placing newly started virtual resources in sites, which are temporarily offering a high amount of RE resources, as well as migrating running virtual resources to such sites. This might require long distance migration and placement across geographically distributed data centers, e.g., in cloud infrastructures, to leverage from fluctuating RE sources (e.g., solar or wind power). However, especially in the case of highly distributed data centers, the already stated relevance of dependencies for the virtual resources (e.g., underlying compute, storage, network resources), needs to be considered. For example, this means that affinity groups for virtual resources (e.g., virtual or physical resources that need to be combined to form a service being offered to distant users) must be respected during the optimization of energy-efficient placement and migration of the virtualized resources. This way, the location independent placement of virtualized resources, due to the abstraction of the virtualization infrastructure from physical hardware, can be used to enhance the use of RE sources within data centers accounting for a large portion of national energy consumption [9].

The rest of this article is laid out as follows. Section II presents related work and defines the research questions of this article. In Section III, the state of the art in energy-efficient private clouds, as well as the usage of virtual network services and live migration of virtual resources in such infrastructures are described. The model for our approach is introduced in Section IV, describing the requirements for scheduling and migrations of virtual network services in private clouds, to support an energy-efficient placement while respecting corresponding requirements like dependencies and affinities of virtual and physical resources. Section V characterizes the testbed created to measure the impact of virtual network service migrations on the energy efficiency of private cloud infrastructures and presents the results of the evaluation. Additionally, Section VI discusses the potential to use energy-efficient placement and migration of virtual resources to leverage fluctuating renewable energies while respecting the earlier defined requirements regarding dependencies and affinities between virtual and physical resources. Finally, Section VII draws a conclusion, discusses the findings of the evaluation compared to the related work, and gives an outlook on further research in this area.

II. RELATED WORK

Migration of virtual resources and its impact on application performance is subject of current research. The energy-efficient placement of VMs in an OpenStack-based environment is discussed in [10][11]. Indeed, these approaches target on the algorithms used for placing VMs based on temperature and cooling demands, but also focus on network requirements for the VMs. A vector-based algorithm for VM placement considering the availability of renewable energies is discussed in [12]. Furthermore, more general evaluations are given in [13] and [14]. These publications examine the relevant parameters for an energy-efficient placement of VMs in a data center. A basic analysis of VM migration costs and the impact of migration on application performances is discussed in [15]. In [16], an estimation of the energy consumption of physical servers running VMs and an algorithm for energy-efficient VM placement are described. The ElasticTree project [17] focuses on energy-efficient computer networks by throttling network components using OpenFlow. Other projects like ECODANE [18] extend these ideas to also provide traffic engineering techniques. Constraints and requirements for energy-efficient placement of VMs related to their network connectivity were introduced in [19][20][21]. An evaluation of the power consumption during VM migration tasks is presented in [6]. This publication also includes a breakdown on different data center components like storage, network and compute resources. Furthermore, [22] discusses an energy-aware virtual data center architecture using software-defined networking (SDN). Finally, [23] introduces benchmark test metrics for performance and reliability monitoring and discusses related issues. A study comparing different hypervisors concerning migration time and efficiency is presented in [24]. The interference effects of simultaneously running migrations and the efficiency of different permutations of migrations are reviewed in [25].

Several publications also address the identification of affinity groups, e.g., between virtual resources and services. Migrations of complete groups including their underlying SDN network structures are introduced in [26]. The base method called LIME is formalized and its correctness is proofed. A deduplication strategy for transmitting identical memory pages only once during the migration of VM groups is outlined in [27]. In [28], the grouping of VMs with the focal point on performance of highly parallelized applications is described. Another algorithm for optimizing the migration of related VMs by reserving bandwidth along traffic paths is outlined in [29]. Statistical techniques are used in [30] to create representative groups of similar VMs in order to simplify monitoring. Management tasks identified for this subset of machines can be applied on all relevant virtual resources.

III. State Of The Art

The evolution of cloud services in IT infrastructures enables companies to speed up business processes and scale their services on demand. Physical servers, storage and network devices are consuming energy, but today these components are typically just the foundation for virtualized workload running on top of them. Moreover, in such highly virtualized environments, the virtual resources providing the services are the decisive consumers of power and bandwidth. Orchestration and automation techniques like SDN can help to optimize the power consumption in cloud infrastructures. To ensure the service quality and scalability along with the energy efficiency, it is necessary to investigate the behavior of these virtual resources, e.g., regarding available migration techniques.

A. Energy-efficient Private Clouds

Today, energy efficiency and power management is a foundation pillar in modern data centers. This is mainly driven by increasingly high energy costs and energy consumption in large-scale IT infrastructures. The sensitization for the sustainable use of resources like renewable energies, e.g., as a result of the Fukushima nuclear disaster and the consequential renunciation of nuclear energy for example in Germany and Europe, additionally supports this process of rethinking data center designs. Data centers are using a large amount of power not only for running the IT components and equipment, but also for cooling them. The ratio between energy consumed by IT equipment and the overall power consumption including cooling and energy loss in power supplies is known as the power usage effectiveness (PUE). This value describes the operational overhead of data centers and is an eligible candidate for optimization approaches.

The concept of cloud computing enables companies to better utilize their physical IT resources and empowers them to dynamically scale their services in a location-independent manner. To take advantage of these benefits, a consequent resource management must be deployed. Ideally, this means that currently not required compute resources, as well as their dependencies like upstream or downstream storage or network devices are partially or fully suspended or shut down. The consumption of energy in a common cloud environment depends on its directly associated physical infrastructure components like compute resources (i.e., central processing unit - CPU, random access memory - RAM), storage devices (i.e., storage area networks - SAN, network attached storage - NAS, local or direct-attached storage) and network components (i.e., routers, switches, firewalls). Thus, the power consumption of a service depends on the physical IT resources that are needed to provide it. However, VMs providing cloud services are not picky concerning their location of execution, as long as required dependencies are met at either site.

By migrating virtual resources across distant data centers in different regions, it is possible to optimize energy efficiency and cost. Such "follow-the-sun" data center services move their workload to different geographic regions to more efficiently balance computing demand while taking into account the latency for the end-users to access the service. Usually, the output of RE sources is fluctuating, which means that the energy is not always available when needed and also not necessarily produced near the point where it is consumed. Further, energy storage at industrial scale is not available yet. Related to that, this also leads to seasonal and regional energy price fluctuations. The cloud paradigm enables companies to move their workload nearby the currently available RE sources and to take advantage of the economic benefits by consuming energy at lower prices.

B. Migration of Virtual Resources in Private Clouds

Today's cloud software is providing a layer for scalable and elastic cloud applications that allows to deploy virtual network services (e.g., VNFs) like routers, load balancers or firewalls. Also, private cloud platforms like OpenStack already added a lot of these functions to their service portfolio. As a result, many industry-leading service providers are starting to use OpenStack as a platform to deliver reliable and scalable services and applications. This includes VMs running customer-facing applications, as well as virtualized storage and networking components needed for the service delivery. Of course, containers as a very thrifty and scalable building block for cloud services can also be provisioned and deployed in these infrastructures. However, to offer reliable, elastic and energy-efficient services, these resources have to be movable across the infrastructure components. This movability of virtual resources is mostly provided by VM migration from one node to another. The migration can be implemented live or online by transferring block storage of the VM or using a shared storage back end, and finally transmitting the main memory and CPU state. Furthermore, a VM can also be migrated offline by suspending, transmitting its state and resuming the machine consecutively. These approaches are described in detail in Section IV-B. When a VM does not contain any essential data and the configuration can be realized by an automated provisioning mechanism, it is also possible to just destroy a VM or container on the source node and recreate or respawn it on the destination node. It is obvious that this technique minimizes network transfer costs and requirements for shared storage hardware but also implies that the cloud application or service is well-designed related to elasticity. Moreover, live migration techniques for containers are currently developed and discussed. While the small size of containers compared to VMs reduces the network traffic for the migration, saving the state of containers holds much more dependencies and hence is more difficult to implement [31].

IV. Energy-efficient Placement of Virtual Network Services in Private Clouds

The migration of virtual network services to regions where RE sources are currently available or where energy prices are lower, can substantially improve the overall energy efficiency of distributed data centers. However, if the costs for the migration are too high, e.g., due to a reduced performance of the migrated resources, the migration will be inefficient. For these reasons, when designing services, it is important to understand how the migration process is performed in the underlying infrastructure to restrict possible negative consequences of migration costs.

A. Scheduling

A common OpenStack-based cloud environment is based on multiple services. First of all, Nova, the compute fabric controller, encapsulates the hypervisor and is responsible for the execution of VMs. Block-level storage is provided by the Cinder service. It manages the complete life cycle of block devices for the virtual servers. The image service Glance stores disk and server images and their metadata and assures that they are available to the compute nodes. The networking component Neutron manages multi-tenant virtual networks supporting different network architectures. For example, traffic can be managed using SDN technologies like OpenFlow. Also, OpenStack Neutron already offers some virtual network services (i.e., VNF) like firewalls and load balancers as a service. While OpenStack contains additional components, this article is based on the OpenStack core services described above. Scheduling and placement of virtual resources in OpenStack environments is carried out by schedulers of the services given above. For example, the nova-scheduler checks which compute nodes can provide the requested resources. The decision is based on filters (i.e., based on capacity, consolidation ratio, affinity groups) that can be modified by an administrator.

B. Migration Techniques

One of the crucial points when performing the migration is to ensure that services should not be disrupted during the migration process, otherwise possible service-level agreements (SLAs) will be violated. OpenStack, which typically uses libvirt and the kernel-based virtual machine (KVM) hypervisor, provides three different migration types to move VMs from the source host to a destination host with almost no downtime: shared storage-based live migration, block live migration and volume-backed live migration [32]. Shared storage-based live migration, as the name states, requires a shared storage that is accessible from source and destination hypervisors. During the migration only the memory content and system state (e.g., CPU state, registers) of the VM are transferred to the destination host. This migration type in OpenStack can be performed using a pre-copy [33] or post-copy [34] approach. In the former, VM memory pages are iteratively copied to the target without stopping the services running on the migrated VM. Every change in memory state (i.e., dirtied memory) during the copy phase will trigger another transfer of modified memory pages. If predefined thresholds have been reached, e.g., the number of iterative copy rounds or the total amount of transmitted memory, or the amount of modified memory pages in the preceding copy round is small enough [35], the copy process is terminated, whereby the source VM is suspended, the source hypervisor copies the remaining modified memory pages and system state and resumes the VM on the destination. Depending on the dirtied page rate this switching can cause a downtime. A big issue of pre-copy migration arises at the iterative copy rounds. If the rate of memory changes exceeds the transfer rate over the network, then the copy process will run infinitely. This limit can be eliminated by post-copy migration, in which at the beginning of the migration the migrating VM is stopped on the original node, then the non-memory VM state is copied to the destination, after which the VM will be resumed on the target. In parallel, a prepaging will be performed. At this stage, the memory pages are proactively pushed by the source to the destination VM. Any access to the memory pages on the target VM that have not yet been copied, result in the generation of page faults, requiring to transfer the accessed memory pages over the network. This process is known as demand paging. Obviously, this behavior can solve the indefinitely migration problem, but can cause a huge degradation of VM performance because of the large amount of page faults transferred over the high-latency medium in comparison to pre-copy migration. Moreover, post-copy cannot recover the memory state of the migrated VM in the case of network failure during the transfer of the page faults.

As the requirement of a shared storage increases the financial burden, block live migration is considered more cost effective. No shared storage is required when the migration takes place. Hence, this migration type is especially useful when moving the VMs between two sites over long distances without having to expose their storage to one another. This type is very similar to Microsoft Hyper-V Shared-Nothing Live Migration feature [36]. Initially, not only a VM on the remote host is created, but also the virtual hard disk on the remote storage. During the migration, at first the virtual hard disk contents of the running VM must be copied to the target host. Changes of disk contents as a result of write operations will be synchronized to the destination hard disk over the network. After the migration of the VMs storage is complete, the copy rounds of memory pages are executed, which perform the same processes used for shared storage-based live migration. Once this stage is successfully finished, the target hypervisor will resume the VM, while the source hypervisor deletes the VM and its associated storage. Volume-backed live migration behaves like shared storage-based live migration since VMs are booted from volumes provisioned by Cinder instead of ephemeral disk, i.e., VM disks on shared storage. To achieve energy-efficient placement of VMs, the migration costs must be taken into account. These costs play an important role for the scheduling process to decide when and how often services should be migrated to remote hosts.

Two categories of parameters to calculate migration costs will be analyzed in this article: total migration time, which denotes how long the migration lasts from the start of copy rounds until the VM is resumed on the remote host, and performance loss, which focuses on the degradation of the services' performance during the migration process. Apparently, these costs are strongly impacted by the iterative copy rounds due to any modification on memory pages or disk contents. They should be thoroughly calculated to allow the scheduler to efficiently place services not only in terms of energy, but also their quality of service.

C. Communication Flows

In most cases a VM cannot be considered on its own. Also, several VMs are part of the provisioning of a service or business process. End-users access services from their devices via front end systems. These facades handle the communication with the end-user devices, the service itself is provided in cooperation with several other systems. This includes systems responsible for directory services, domain name resolution and identity management but also components of the business process itself like database management systems and application servers. For each exchange of data, a connection needs to be established between the involved systems, e.g., via the internet and transport layer.

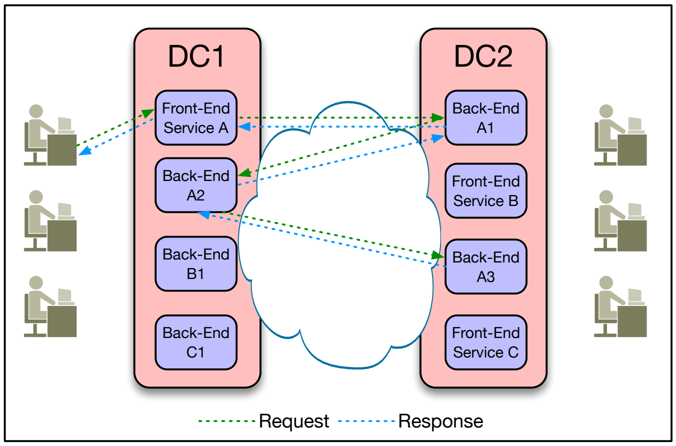


Figure 1. Suboptimal distribution of virtual resources across two data centers.

This clearly illustrates that latencies in communication flows between the involved systems have to be accumulated. The resulting overall latency is usually small when the systems reside at the same site or ideally inside the same hypervisor. But these latencies rise when the involved systems are distributed geographically. In most cases, the geographical positioning of the end-users can be neglected due to their uniform distribution, but a poor distribution strategy of the back end services can lead to negative effects on end-users latencies and an overall degradation of service quality. Figure 1 depicts a suboptimal distribution of virtual resources in two data centers. An end-user's request to the front end server results in additional requests to back end servers, which due to their poor placement across different data centers lead to a high end-user latency.

A favorable distribution is shown in Figure 2. Although the distance between end-user and the front end system is greater, a smaller overall processing time is achieved by grouping the involved server systems in one data center. These sets of associated VMs are known as affinity groups. Typically they comprise the services and their dependencies, e.g., in the form of virtual resources.

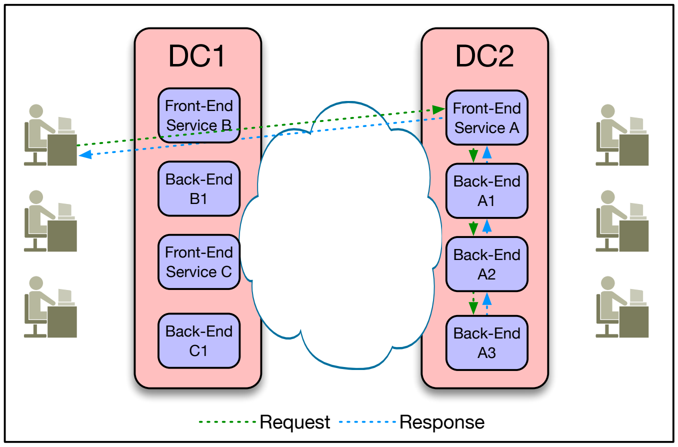


Figure 2. Favorable distribution of virtual resources across two data centers.

D. Identification of Affinity Groups using Network Flows

There are different motivations for the collection of network traffic data, e.g., capacity planning, accounting or security monitoring (see for example [37]). The foundation builds the analysis of aggregated network flows. As stated in RFC 3917 [38], a so-called network flow can be seen as a set of internet layer packets that pass an observation point during a given time interval and share a common set of properties. The observable properties and the usable sampling mechanisms differ for the various technologies like NetFlow v5, NetFlow v9, IPFix and sFlow. For the sake of identifying affinity groups, all of these protocols provide source and destination addresses and basic counters. However, NetFlow v5 is only usable for monitoring IPv4 traffic. Today, the collection and aggregation of network flow data is not limited to physical switching or routing hardware. Especially, networks with redundant links and devices allow traffic to follow different paths from source to destination. A viable solution is to collect the network flow data as near as possible to the source or destination and to assure that all packets pass this observation point. In today's data centers and their high degree of virtualization, these data can also be collected using virtual network devices.

Listing 1. Configuration of NetFlow, sFlow and IPFIX using an Open vSwitch (see [39])

```
# ovs-vsctl -- set bridge vswitch \
  netflow=@netflow -- --id=@netflow create \
  netflow target=\"10.1.1.42:2055\" \
  active-timeout=30

# ovs-vsctl -- set bridge vswitch \
  sflow=@sflow -- --id=@sflow create \
  sflow agent=eth0 target=\"10.1.1.42:6343\" \
  header=128 sampling=64 polling=10

# ovs-vsctl -- set bridge vswitch \
  ipfix=@ipfix -- --id=@ipfix create \
  ipfix targets=\"10.1.1.42:4739\" \
  obs_domain_id=123 obs_point_id=456 \
  cache_active_timeout=60 cache_max_flows=12
```

For instance, VMware's Distributed Switch allows the collection of network flows and their export via NetFlow or

IPFix to an external collector. Furthermore, an Open vSwitch – used esp. in KVM- and Xen-based virtualization environments – supports the export of aggregated data using IPFix, NetFlow and sFlow. These virtual switches also form the basis for networking in OpenStack-based environments, as being used in this article. Listing 1 shows the commands for enabling flow collection and export using NetFlow, sFlow and IPFix.

Network flows and the associated communication endpoints can be used to build up affinity groups based on an aggregated metric and a given threshold. An affinity group can be seen as a partitioning for the set of VMs M. Thus, an equivalence relation R_P can be defined as follows:

a) Reflexivity: Each virtual resource $m \in M$ is part of the relation, so $\forall m \in M : (m, m) \in R_P$.

b) Symmetry: The related VMs whose communication volume exceed a given threshold s should also be included in the relation. So, let $\lambda(m_i \in M, m_j \in M)$ be a function that returns an aggregated metric (e.g., packet count or octet count) for two given VMs and it holds that $\lambda(m_i, m_j) = \lambda(m_j, m_i)$. Based on this constraint the symmetry is also given for the relation for machines that exceed a given threshold s, so $\forall m_i, m_j \in M, \lambda(m_i, m_j) \geq s : (m_i, m_j) \in R_P$.

c) Transitivity: The missing tuples for fulfilling the transitivity should also be added by $\forall (m_i, m_j) \in R_P, (m_j, m_k) \in R_P : (m_i, m_k) \in R_P$.

Based on this definition, a partitioning for the set M, i.e., the affinity groups, can be defined as $P_M = \{[m]_{R_P} | m \in M\}$. An example of such a breakdown can be seen in Figure 3.

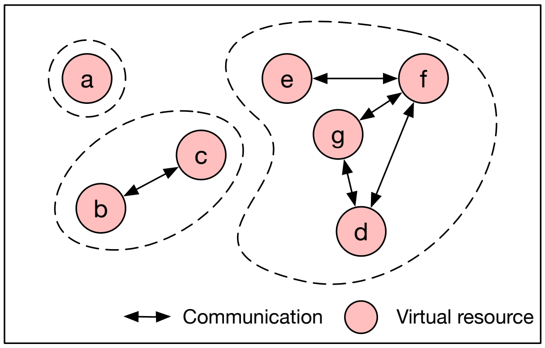


Figure 3. Partitioning of virtual resources in affinity groups based on their communication volume.

Beside the evaluation of the performance impact of VM migrations, the results also show that it is profitable to migrate VMs in order to group VMs geographically. In [40], an algorithm to optimize the usage level of renewable energies of distributed data centers was introduced. In addition to this goal, this algorithm uses network flow data to build affinity groups of VMs to avoid negative side effects. For example, these side effects could be high latency or bitrate capacity exhaustion. As a result, the consideration of the affinity groups allows for the minimization of the average distance across virtual resources that packets are traveling between. Thereby, a decrease of the overall service latency is enabled.

V. EVALUATION

This experimental study concentrates on the impact of migration on memory- and I/O-intensive services. For this purpose, an experiment was set up in an OpenStack environment that is presented in the following sections.

A. Testbed Environment and Methodology

Our testbed environment consists of two physical servers that act as compute nodes and two NetApp E2700 providing block storage over 16 Gbit/s FibreChannel. Each of the compute nodes running Ubuntu 14.04 is equipped with two 8-core Intel(R) Xeon(R) E5-2650v2 2.60 GHz CPUs and 256GB of main memory. The nodes are connected using two 1 Gbit/s Ethernet interfaces over a Cisco C3750 switch. All migrated VMs run Ubuntu 14.04 with 1 vCPU, 2 GB of memory and 10 GB of disk space. In our study, migration costs of a web proxy as a virtual network service is analyzed. 10 VMs (Set 1) representing web proxy servers are initially launched on Nova-Compute 1 with a defined memory workload using the tool *stress*, which keeps dirtying a predefined amount of memory. Swapping was also activated to simulate additional I/O load on the service. If all memory for user space (1702 MB, 83% of memory size) is already allocated, inactive memory pages will be swapped out to disk.

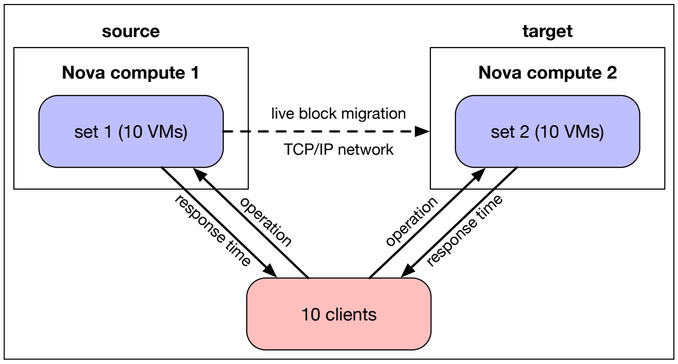


Figure 4. Overview of the methodology of the experiment.

The performance of each VM will be measured by 10 clients, each sending HTTP requests to the VMs in a fixed time interval. Additionally, extra load was produced on those VMs by sending other requests for various operations from the clients, such as searching a directory, writing a 20 MB file (disk I/O load) and generating 4096 bit RSA keys (CPU load). The response times for those requests are then used as a performance metric. After 15 minutes of measurement the same process is performed on 10 VMs of Set 2 on Nova-Compute 2. All source VMs are then concurrently migrated from Nova-Compute 1 to Nova-Compute 2 using block live migration. Block live migration were chosen due to its advantage in the case of moving the VMs located on two sites with large distance. While also 10 Gbit/s Ethernet is available in our servers and switches, the 1 Gbit/s NICs were used to better reinforce small effects of different migration parameters and changes. Furthermore, the number of concurrent migrations were varied to better understand the impact of the bandwidth on the migration. The performance of VMs on Nova-Compute 2 was also investigated to observe the influence of the migration on instances running on the target host. Figure 4 shows an overview of the methodology.

Besides several configurations that were necessary to implement a true live migration in OpenStack [32], the *max_requests* and *max_client_requests* parameters in libvirt had to be increased to 40, to support the large number of 10 concurrent migrations in the experiment. This parameter was changed in the libvirt configuration file and followed by a restart of the libvirt service. The experiment was performed using a script and was repeated 10 times to ensure significant and reproducible results. All runs led to reproducible results. After changing a parameter in the experiment (e.g., the memory workload shown in Figure 5) it was run 10 times again.

B. Research Results and Discussion

Figure 5 demonstrates the experimental results for different memory workloads. The results show that the total migration downtime increases proportionally with stressed memory size caused by the iterative transfer of dirtied memory pages generated by the command-line tool *stress*. Another reason for this effect is the more intensive swapping of memory pages leading to a repeated modification of disk contents and thus more additional transfers over the network. In addition, the block live migration process in OpenStack will last longer, if the number of VMs migrated concurrently was reduced. The source of this impact is the overhead of nova-scheduler handling the migration requests.

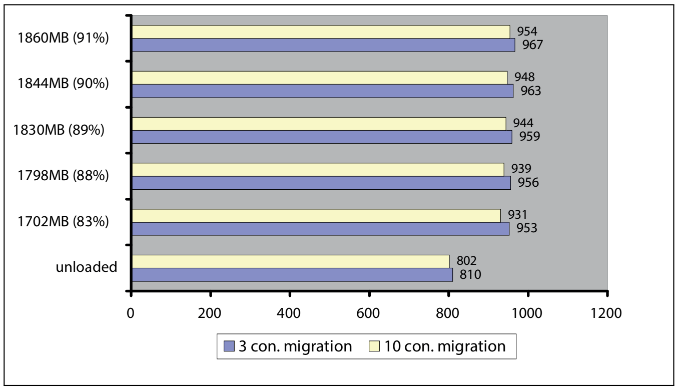


Figure 5. Total migration time (in seconds).

During the migration process, the performance degradation for search operations within the VMs significantly starting from 1830 MB loaded-memory (89% of total memory size) were observed. This degradation is shown in Figure 6, which demonstrates the response time for search operations on both sets before, during and after concurrently migrating 10 VMs of Set 1 to Nova-Compute 2. Response times were capped to a maximum of 60 seconds as seen in the figure for the second set before its creation. The average response time on Set 1 during the copy rounds rises from 2.299s to 5.606s, approximately 144%. Moreover, the migration of Set 1 to Nova-Compute 2 influences the VMs performance for search operations on this node. Particularly, the average search response time of Set 2 increases around 110% from 2.45s to 5.164s. After the VMs are moved to Nova-Compute 2, the performance of both sets is also decreased, by approximately 72% on Set 1 and 61% on Set 2, since Set 1 produces more I/O workload on the disk of the target host. The peak in Figure 6 during the migration denotes the switch process that was explained in Section IV-B.

Another conspicuous point is that the performance loss during the migration strongly depends on the amount of stressed memory as shown in Table I. The performance loss increases linear with the size of the memory workload. This could be due to the fact that the available amount of memory for buffer/cache used for I/O operations is too low so that more intensive I/O flush processes occur. Consequently, more disk synchronization must be performed over the network during the migration, causing a slowdown in the response times. In Figure 6, one can recognize that the performance of Set 1 for the search operation slightly degrades when the VMs of Set 2 on Nova-Compute 2 are started, although they do not use a shared storage. For instance, the average response time of Set 1 increases from 2.067s to 2.532s (22.5%) in the case of 1830 MB loaded-memory, from 2.229s to 3.083s (39.7%) in the case of 1844 MB loaded-memory and from 2.856s to 6.858s (140%) in the case of 1860 MB loaded-memory. This result shows that many simultaneous intensive I/O operations on an extremely memory-intensive VM have an immense impact on the I/O performance of the underlying system in OpenStack and on the performance of I/O operations in hosted VMs, respectively. Nevertheless, this effect does not emerge if the stressed memory falls below 1830 MB, as well as for other non-I/O-related operations.

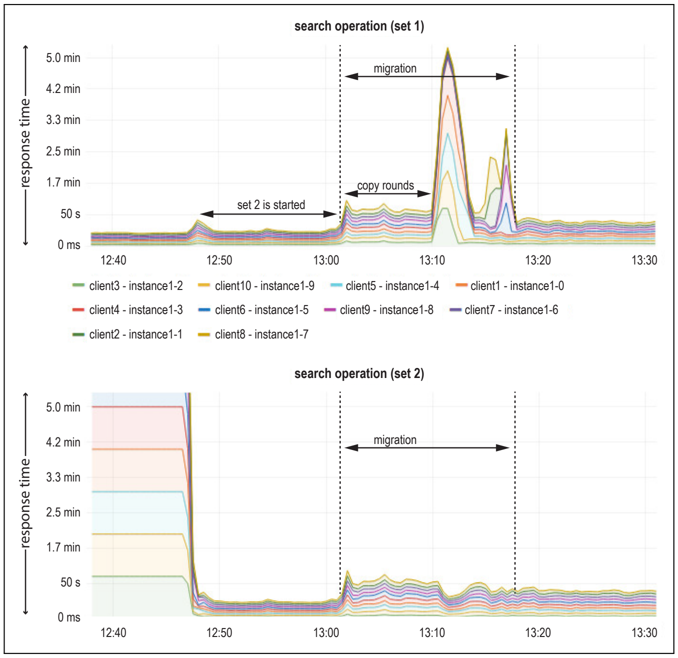


Figure 6. Performance of search operations with a memory workload of 1830 MB on Set 1 and Set 2 before, during and after the migration.

TABLE I. PERFORMANCE LOSS OF SEARCH OPERATION WITH DIFFERENT MEMORY WORKLOADS.

VM set	*Increased response time during migration (s)*		
	1830 MB	1844 MB	1860 MB
Set 1	3.307	4.389	6.241
Set 2	2.712	3.678	3.422

VM set	*Increased response time after migration (s)*		
	1830 MB	1844 MB	1860 MB
Set 1	1.655	2.527	3.431
Set 2	1.498	2.044	1.265

Last but not least, the performance of the main operation of the web proxy, serving HTTP requests, as well as the performance of the CPU-related operation, generating a 4096 bit RSA key, are only significantly impacted as the amount of stressed memory rises above 1860 MB. The average response time for HTTP requests to the migrating set grows from 0.166s to 0.785s during the migration process, whereas the one for the operation of generating an RSA key rises from 3.759s to

6.3s. This degradation effect arises only if those operations are carried out while other I/O-intensive operations such as a search for a file are running. When block live migration with separate operations were performed, the performance deviation did not occur. Therefore, it could be stated that not only I/O operations are strongly impacted by the migration process, but also have direct influence on the other operation types.

VI. USING RENEWABLE ENERGIES FOR VIRTUAL NETWORK SERVICES IN PRIVATE CLOUDS

As stated in the introduction in Section I, efficient placement and migration of virtual resources can be used to leverage RE sources. However, the energy output of RE sources is fluctuating. Hence, the energy is not always available when needed or vice versa. In addition, the energy is not always produced near its point of use (e.g., offshore wind energy). First of all, this leads to the necessity to store the energy in between or shift the consumption in time. Until now, the storage of energy is just conditionally feasible, as it is expensive and not available in industrial scale. In contrast, the possibility to shift the energy consumption of data centers with the help of an intelligent energy management is viable, as stated in the introduction of this article and in further detail, e.g., presented in [9]. Migrating VMs can help counteract the issue of fluctuating energy sources. However, as described in Section IV-B, each shift is associated with migration costs. VMs should only be moved if it is certain that a shift is worthwhile. Furthermore, it must be ensured that the VMs do not oscillate between different locations in short time. This could happen, for example, if the RE fluctuates sharply in short periods at different locations. Various optimization options have already been presented in this paper. To further optimize the number of shifts, this article proposes an additional idea besides the concept of migrating VMs based on available RE. To avoid additional shifts, VMs should be started directly where high RE power is available, or where high performance is expected over time. The adaptive placement of VMs can use the same underlying virtualization technology (i.e., virtual resource scheduling), as already described in Section IV-A. To reduce the possibility of required migrations, a weather forecast of the following day should be included in the placement decision. In order to ensure the energy supply, energy providers have been using weather forecasts for a long time for the prognosis of energy from wind turbines. Recently, such systems have also been used for photovoltaics (PV). The weather forecast can estimate how much energy will be available in certain locations. With this information, the VMs can now be started exactly where energy from RE is expected. It also should be accepted that energy from RE is not available immediately, or not consistently. For example, if there is a lot of energy from PV at location A according to the weather forecast for the following day, VMs should be started there. In this case, it may be necessary to accept that the VMs are started, e.g., at 8 a.m. in the morning, but the power from PV is not available until 10 a.m. in an acceptable size. A software fed with relevant information can create a schedule for such scenarios to circumvent these deficiencies and possible additional costs. Virtual resources (like VMs) that recur regularly and are present for a longer period of time per day, can be started at locations with high RE. Weather forecasts can be requested from various weather services. Artificial neural networks (ANNs) or machine learning algorithms can also be used to create and improve the schedule. They can learn from the interaction with the virtualized resources and the user behavior, e.g., based on incoming and outgoing communication flows, as discussed in Section IV-C.

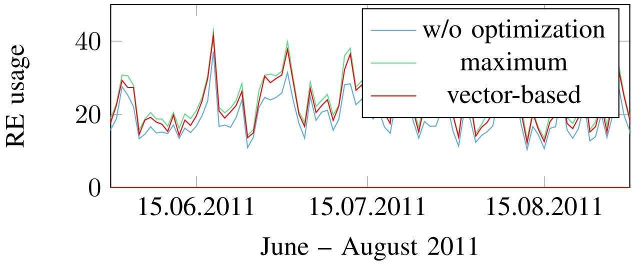
A. Considering Renewable Energies for Virtual Machine Placement Optimization

To evaluate potential benefits from leveraging renewable energies for the placement of VMs in cloud environments, this section presents an optimization of the placement and possible migration of virtual resources across geographically distributed data centers. Primarily, the optimization focuses on the utilization of fluctuating renewable energies at three locations in Germany. The optimization uses weather data from these distant locations, as already presented in [9]. Furthermore, as explained in detail in Section IV-D, affinities and dependencies between virtual resources have to be taken into account for the optimization. Therefore, traffic patterns from VMs running in the virtualization environment of the data center at Fulda University of Applied Sciences and their anonymized incoming and outgoing flows where used as the important secondary criterion for the optimization. The optimization was based on the algorithm introduced in [40]. It uses vector-based approach to iteratively search for virtual resources to migrate while taking the geographical distance and corresponding affinities as well as available renewable energy sources into account. This algorithm tries to maximize the level of utilization of renewable energies and also limits and prevents negative side effects like increased end-user latencies. As mentioned, the algorithm's primary goal of maximizing the usage level of renewable energies, used weather data for three data center locations in northern, central and southern Germany. This data included measurements of global solar radiation and wind speed in ten minute resolution over multiple years. The general feasibility of such an optimization approach was introduced in [9]. Furthermore, network flow data produced by our university data center were used. The data set contains flows between approximately hundred virtual resources, as well as flows identified to either come from or go to physical machines from these virtual resources.

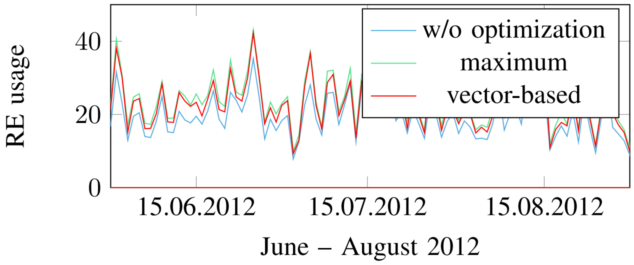
Based on this data, a RE usage optimization was conducted, which moved VMs between data centers in order to move energy consumers near to the energy producers, as discussed for the live migrations in this article in Section IV. Results for the years 2011, 2012 and 2013 are shown in Figure 7a, 7b and 7c. For better legibility and clarification the consecutive summer months June, July and August are chosen for the displayed period in the charts.

B. Optimized Communication Distances for Affinity Groups

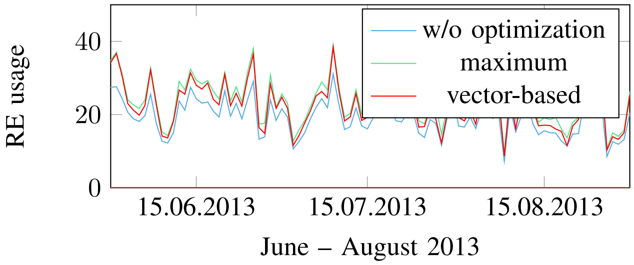
Beside the goal of raising the utilization of RE sources, the algorithm mentioned in the previous section also optimized the communication distances for affinity groups, so that the average distance each packet needs to travel across the network was reduced. This was accomplished by determining the geographical location for flow communication endpoints and by computing the average distance per packet. This not only assures traffic locality for affinity groups, but also reduces the average distance per packet to end-users. This results in a reduced end-user latency. The optimization used real-world network flow and affinity characteristics from the mentioned



(a) Optimized RE usage for the summer period 2011.

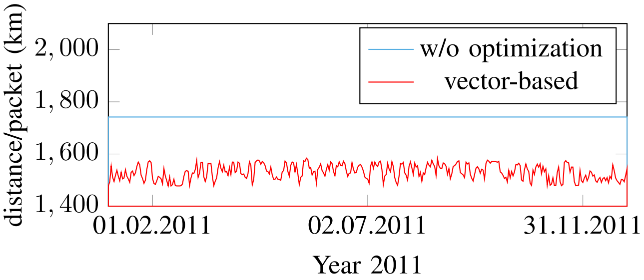


(b) Optimized RE usage for the summer period 2012.

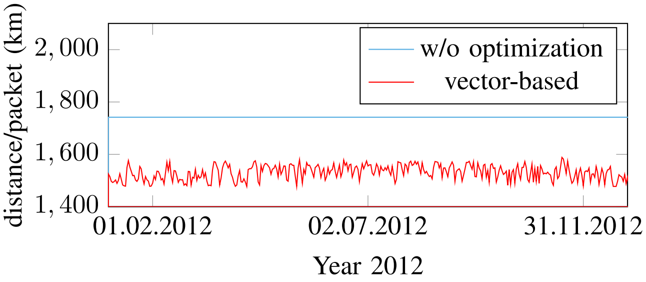


(c) Optimized RE usage for the summer period 2013.

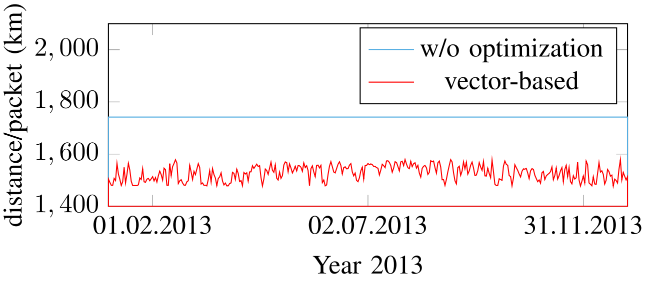
Figure 7. Optimized usage of renewable energy (RE) for the years 2011-2013.



(a) Optimized average distance per packet for the year 2011.



(b) Optimized average distance per packet for the year 2012.



(c) Optimized average distance per packet for the year 2013.

Figure 8. Optimization for the three years 2011-2013.

data set from VMs within a virtualization environment at Fulda University of Applied Sciences.

Figure 8a, 8b, and 8c show the results for the three years 2011, 2012 and 2013. The figures clearly illustrate that the average distance of communication endpoints was minimized. The results are summarized in Table II.

TABLE II. VECTOR-BASED ALGORITHM OPTIMIZED DISTANCE.

Year	Minimum [km]	Maximum [km]	Average [km]
2011	-265.55	-0.26	-211.21
2012	-265.59	-0.30	-211.34
2013	-265.59	-0.26	-214.39

The average communication distance was reduced between 211.21 km and 265.59 km per packet. This could lead to an approximate latency improvement of, e.g., one or two milliseconds for fiber-based networks.

VII. CONCLUSION AND FUTURE WORK

Energy costs are an important factor for today's IT infrastructures, due to rising energy prices and increasing power consumption. The virtualization offered for compute, storage and network resources, e.g., in private clouds, allows for a seamless and transparent migration of virtual resources due to the abstraction from the underlying hardware. These migration techniques can be used to enhance the energy efficiency in data centers and have been constantly evolving over the last decade. This includes adaptive migration, e.g., to consolidate or enhance the utilization of physical resources, as well as long-distance migration, which is not only covered by the related work and research presented in this article, but also by current virtualization and hypervisor products (e.g., the introduction of long-distance vMotion in VMware vSphere 6 that was previously already available in Microsofts Hyper-V). Regarding the energy efficiency, however, additional costs of the migration itself have to be taken into account. These costs can either directly (i.e., higher load on the physical compute, storage and network resources) or indirectly gain energy costs, e.g., if the migrated services and applications cannot provide the same service quality during the migration. Hence, to improve the energy efficiency by using live migration techniques offered in cloud infrastructures, the migration costs need to be minimized. This especially holds true, if the migration is used to benefit from lower energy prices or the availability of RE at distant data center sites.

Based on our previous research projects in this area, this article introduce an evaluation of the migration costs for I/O-intensive VMs in an OpenStack environment. Due to the incoming and outgoing network traffic, especially virtual network services operated in VMs typically have a large I/O footprint in the infrastructure that is typically compensated by using hardware acceleration (e.g., virtual switch or kernel enhancements, DPDK, SR-IOV, FD.io, XDP etc.). To be able to measure the additional load caused by a live migration of such services, and to quantify the impact on the service quality, additional tools (i.e., *stress*, *openssl*, *dd*, *find*) are used to add artificial I/O load on the machines while migrating them to another physical host in the OpenStack infrastructure. Based on the findings presented in this article, the migration time increases proportionally to the added artificial I/O load. Furthermore, the load on storage and network resources grows accordingly as expected. The burden of the ongoing live migration can especially be measured if more than 80% of the total memory of the VM are continuously utilized and changed. Interestingly, the migration time can be reduced by increasing the number of concurrent live migrations. This is due to the impact of the scheduler and message bus, handling the migrations in OpenStack together with libvirt and KVM. Similar effects can be observed with other hypervisors like vSphere or Hyper-V, though these products typically limit the number of parallel live migrations to smaller values.

The results of the experiments show a significant performance decrease for I/O read operations on the VMs during the migration. This conspicuous effect is likely due to limited available buffer/cache and extensive flush operations during the migration. The impact on the underlying OpenStack infrastructure leveraging libvirt and KVM, can also be observed in a performance decrease during start of VMs with high I/O and memory load, even if the VMs are running on separate hosts using different block storage. Several I/O operations (i.e., using *dd*, *find*, *stress*) were used to evaluate this decrease while constantly monitoring the service quality of the main operation. During the migration, a *find* process across the files on the VMs experienced a significant performance decrease. Also, VMs running on the target machine for the migration, experience a significantly reduced performance during this period. Moreover, for high additional artificial I/O loads, the main operation of the virtual network service was also impacted accordingly. Response times on the migrated service (offering the function of a web proxy) increased from 0.166s to 0.785s during the migration. The high I/O load on the VMs leads expectedly to higher overall response times as more and more VMs are consolidated on a single physical host. However, a previous paper [6] presented an expected increase of the overall energy efficiency due to the higher utilization of the physical host, made possible by this consolidation.

Building on the results presented in this article, we are currently focusing our research on live migration techniques for containers as a lightweight virtualization alternative compared to full-size VMs. Some types of services allow migration and scaling by simply destroying the containers at one site and respawning them at another. The required live migration techniques for containers are still being developed (e.g., in CRIU [31]) and are also within the focus of some related research projects. Initial results of our experiments show that the transferred amount of data during container migrations is expectedly less compared to VMs. Conversely, the migration process itself is more difficult, as the entire state of a process stack in the operating system needs to be stored and transferred. Existing checkpoint and restore techniques need to be extended to support live migration of container-based virtual network services. As virtualization techniques like containers are evolving, the requirement to seamlessly migrate virtual resources is likely to grow. Additionally, virtualization techniques themselves make heavy use of virtual network functions, for example to form overlay networks for distributed container networks, e.g., using virtual routers, switches, load balancers or firewalls in distributed clouds, offering potential for energy-efficient migrations of virtual network services and resources in upcoming cloud infrastructures.

In this article, affinity groups and dependencies between migrated and adaptively placed virtual resources have been identified and considered for the optimization. This way, possible negative side effects of migrations, e.g., leading to high access latencies or higher communication overhead after the migration or separation of highly dependent resources were addressed. Concerning the identification of affinity groups of VMs and virtual network services, an additional classification of communication endpoints could be viable to prioritize and weight traffic for different virtual resources (e.g., staff/customer machines or central services). Furthermore, a comparison of the raised migration costs in relation to the benefits of the optimization would be helpful to rate the quality and effectiveness of the developed algorithms. Moreover, an energetic estimation of communication efforts based on the distance packets need to be sent across, can shed light on energy savings in computer networks based on real world scenarios.

As a potential beneficial use case of the optimization presented in this article, the use of renewable energies was discussed. Migrating as well as consolidating virtual resources at sites with currently high renewable energy power or low energy prices was presented. Besides the migration of virtual resources, this could also include an energy-aware placement of virtual resources in the first place. If affinity groups and dependencies of virtual and physical resources are also considered for the optimization, this allows for a better overall utilization of renewable energies for data centers owing the location-independent placement and movability of virtual resources. The importance of affinity groups and dependencies is especially important for the virtual network services. The network not only enables distributed services, but also introduces costs either directly from operating the communication systems and links or indirectly, e.g., resulting from additional latencies when using these services. While these dependencies where addressed in the optimization algorithm used in this article, a further optimization regarding involved costs could be to use forecasts of available renewable energies at each site in distributed cloud infrastructures. Further research needs to be carried out in this area, to profit from the fluctuation of renewable energies. This includes the investigation of the use of machine learning techniques to improve forecasts for available renewable energies as well as dependencies and affinities (e.g., based on requirements of network flows between the virtual services and end-users). The data sets used for weather data at three different locations in Germany and the network flow data of virtual machines at Fulda University of Applied Sciences will be used as a starting point for further research projects in this area.

REFERENCES

[1] N. K. Truong, C. Pape, S. Rieger, and S. Reißmann, “Energy-efficient live migration of I/O-intensive virtual network services across distributed cloud infrastructures,” in ICSNC 2017, The Twelfth Interna-

tional Conference on Systems and Networks Communications, 2017, pp. 6–11.

[2] 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, Dec. 2008, pp. 68–73.

[3] 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, 2012.

[4] 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/, 2018-05-26.

[5] P. Mell and T. Grance, The NIST definition of cloud computing. Washington DC: National Institute of Standards and Technology, 2011, URL: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf, 2018-05-26.

[6] 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.

[7] ETSI, “Network Functions Virtualisation (NFV); Infrastructure Overview,” 2015, URL: {http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/001/01.01.01_60/gs_NFV-INF001v010101p.pdf}, 2018-05-26.

[8] S. Tilkov, “The Modern Cloud-Based Platform.” IEEE Software, 2015.

[9] K. Spindler, S. Reissmann, R. Trommer, C. Pape, S. Rieger, and T. Glotzbach, “AEQUO: Enhancing the energy efficiency in private clouds using compute and network power management functions,” International Journal On Advances in Internet Technology, vol. 8, no. 1 & 2, 2015, pp. 13–28.

[10] 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. IEEE Computer Society, 2010, pp. 826–831.

[11] ——, “OpenStack neat: A framework for dynamic consolidation of virtual machines in OpenStack clouds–a blueprint,” Cloud Computing and Distributed Systems (CLOUDS) Laboratory, 2012.

[12] C. Pape, S. Rieger, and H. Richter, “Leveraging Renewable Energies in Distributed Private Clouds,” MATEC Web of Conferences, vol. 68, Aug. 2016, p. 14008.

[13] 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.

[14] 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.

[15] 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. Springer-Verlag New York, Inc., 2008, pp. 243–264.

[16] D. Versick and D. Tavangarian, “CAESARA-Combined Architecture for Energy Saving by Auto-adaptive Resource Allocation.” in DFN-Forum Kommunikationstechnologien, 2013, pp. 31–40.

[17] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “ElasticTree - Saving Energy in Data Center Networks.” NSDI, 2010.

[18] T. Huong, D. Schlosser, P. Nam, M. Jarschel, N. Thanh, and R. Pries, “ECODANE-reducing energy consumption in data center networks based on traffic engineering,” in 11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop Visions of Future Generation Networks (EuroView2011), 2011.

[19] V. Mann, K. Avinash, P. Dutta, and S. Kalyanaraman, “VMFlow: Leveraging VM Mobility to Reduce Network Power Costs in Data Centers.” Networking, vol. 6640, no. Chapter 16, 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] 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.

[22] Y. Han, J. Li, J. Y. Chung, J.-H. Yoo,, and J. W.-K. Hong, “SAVE: Energy-aware Virtual Data Center embedding and Traffic Engineering using SDN.” NetSoft, 2015, pp. 1–9.

[23] T. Kim, T. Koo, and E. Paik, “SDN and NFV benchmarking for performance and reliability.” APNOMS, 2015, pp. 600–603.

[24] W. Hu, A. Hicks, L. Zhang, E. M. Dow, V. Soni, H. Jiang, R. Bull, and J. N. Matthews, “A quantitative study of virtual machine live migration,” in Proceedings of the 2013 ACM cloud and autonomic computing conference. ACM, 2013, p. 11.

[25] K. Rybina, A. Patni, and A. Schill, “Analysing the migration time of live migration of multiple virtual machines.” in CLOSER, 2014, pp. 590–597.

[26] S. Ghorbani, C. Schlesinger, M. Monaco, E. Keller, M. Caesar, J. Rexford, and D. Walker, “Transparent, live migration of a software-defined network,” in Proceedings of the ACM Symposium on Cloud Computing, ser. SOCC ’14, 2014, pp. 3:1–3:14. [Online]. Available: http://doi.acm.org/10.1145/2670979.2670982

[27] U. Deshpande, X. Wang, and K. Gopalan, “Live gang migration of virtual machines,” in Proceedings of the 20th international symposium on High performance distributed computing. ACM, 2011, pp. 135–146.

[28] T. Lu, M. Stuart, K. Tang, and X. He, “Clique migration: Affinity grouping of virtual machines for inter-cloud live migration,” in Networking, Architecture, and Storage (NAS), 2014 9th IEEE International Conference on. IEEE, 2014, pp. 216–225.

[29] G. Sun, D. Liao, D. Zhao, Z. Xu, and H. Yu, “Live Migration for Multiple Correlated Virtual Machines in Cloud-based Data Centers,” IEEE Transactions on Services Computing, vol. PP, no. 99, 2015, pp. 1–1.

[30] C. Canali and R. Lancellotti, “Improving scalability of cloud monitoring through pca-based clustering of virtual machines,” Journal of Computer Science and Technology, vol. 29, no. 1, 2014, pp. 38–52.

[31] K. Kolyshkin, “CRIU: Time and space travel for linux containers,” 2015, URL: http://de.slideshare.net/kolyshkin/criu-time-and-space-travel-for-linux-containers, 2018-05-26.

[32] OpenStack Foundation, “OpenStack administration guide,” 2017, URL: http://docs.openstack.org/admin-guide-cloud/index.html, 2018-05-26.

[33] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, “Live migration of virtual machines,” in Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2. USENIX Association, 2005, pp. 273–286.

[34] M. R. Hines, U. Deshpande, and K. Gopalan, “Post-copy live migration of virtual machines,” ACM SIGOPS Operating Systems Review, vol. 43, no. 3, 2009, p. 14.

[35] A. Strunk, “Costs of virtual machine live migration: A survey,” 2012 IEEE Eighth World Congress on Services, 2012, pp. 323–329.

[36] Microsoft, “Virtual machine live migration overview,” 2015, URL: https://technet.microsoft.com/en-US/library/hh831435.aspx, 2018-05-26.

[37] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, “Flow Monitoring Explained - From Packet Capture to Data Analysis With NetFlow and IPFIX.” IEEE Communications Surveys and Tutorials, 2014.

[38] J. Quittek, T. Zseby, B. Claise, and S. Zander, “Requirements for IP Flow Information Export (IPFIX),” Internet Requests for Comments, RFC Editor, RFC 3917, October 2004, 2018-05-26. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3917.txt

[39] ovs-vsctl, Open vswitch manual ed., The Linux Foundation, 1 Letterman Drive, Building D, Suite D4700, San Francisco CA 94129, 2017, 2018-05-26. [Online]. Available: http://openvswitch.org/support/dist-docs/ovs-vsctl.8.pdf

[40] C. Pape, S. Rieger, and H. Richter, “Leveraging Renewable Energies in Distributed Private Clouds,” in Proc. International Conference on Advances on Clean Energy Research (ICACER 2016), 16-18 April 2016, pp. 26–30. [Online]. Available: http://www.icacer.com