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

Ngoc Khanh Truong*, Christian Pape*, Sebastian Rieger*, Sven Reißmann†
*Department of Applied Computer Science
Fulda University of Applied Sciences, Fulda, Germany
Email: {ngoc.k.truong, christian.pape, sebastian.rieger}@cs.hs-fulda.de
†Datacenter
Fulda University of Applied Sciences, Fulda, Germany
Email: sven.reissmann@rz.hs-fulda.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, or the ability to geographically move cloud services based on lowest available energy prices and renewable energy. This paper 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. The results can be used to evaluate whether network services and virtual resources can be migrated to distant sites to reduce energy costs. Correspondingly, the paper illustrates the impact of high memory and input/output (I/O) load on live migrations of network services and evaluates possible optimization techniques.**

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

I. INTRODUCTION

Energy costs are an important factor for data centers and IT infrastructures as a whole. Drivers for the increasing costs over the last years 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% [1][2] 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 [3]. Since virtualization is used for compute, storage and network resources in modern data centers, these infrastructures support automatic provisioning and management of virtual resources, that 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 [4], as well as adaptive scheduling, placement and migration of virtual resources. The scheduler can consider electricity prices and the availability of renewable energy resources across multiple data centers [5]. Hence, an energy- and cost-efficient adaptive placement of virtual resources can be attained. Nevertheless, network services impose special requirements for live migrations. The network load, e.g., on virtual network functions (VNF), 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 enhanced by using special acceleration functions of the underlying hardware, i.e., TCP offloading or single-root I/O virtualization (SR-IOV), that also hold specific constraints for live migrations.

In this paper, an analysis of the impact of these implications for the migration of virtual network services across distributed cloud environments is presented. Our 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 rest of this paper is laid out as follows. Section II presents related work and defines the research questions of this paper. 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. Section V characterizes the testbed that we 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. Finally, Section VI draws a conclusion, discusses the findings of the evaluation compared to the related work, and gives an outlook on further research that we will carry out 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 virtual machines in an OpenStack-based environment is discussed in [6][7]. 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 virtual machine placement considering the availability of renewable energy is discussed in [8]. Furthermore, more general evaluations are given in [9] and [10]. 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 [11]. In [12] an estimation of the energy consumption of physical servers running VMs and an algorithm for energy-efficient VM placement are described. The well-known ElasticTree project [13] focusses on energy-efficient computer networks by throttling network components using OpenFlow. Other projects like ECODANE [14] 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 [15][16][17]. An evaluation of the power consumption during VM migration tasks is presented in [5]. This publication also includes a breakdown on different data center components like storage, network and compute resources. Furthermore, [18] discusses an energy-aware virtual data center architecture using software defined networks (SDN). Finally, [19] introduces benchmarking test metrics for performance and reliability monitoring and discusses related issues. A study comparing different hypervisors concerning migration time and efficiency is presented in [20]. The interference effects of simultaneously running migrations and the efficiency of different permutations of migrations are reviewed in [21].

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. Furthermore, in such highly virtualized environments, the virtual resources providing the services are the 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. 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 renewable energy 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 renewable energy 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 [22].

IV. ENERGY-EFFICIENT PLACEMENT OF VIRTUAL NETWORK SERVICES IN PRIVATE CLOUDS

The migration of virtual network services to regions where renewable energy sources are currently available or where energy prices are lower, can substantially improve the overall energy efficiency. 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 the consequences of migration costs.

A. Scheduling

A common OpenStack environment is based on multiple services handling different aspects of the cloud environment. 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. 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 paper 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 [23]. 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 [24] or post-copy [25] approach. In the former, VM memory pages are iteratively copied to the target without stopping the services running on the migrated VM. Every change on 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 [26], 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 change exceeds the transferred 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 [27]. 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 paper: 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.

V. EVALUATION

This experimental study concentrates on the impact of migration on memory- and I/O-intensive services. For this purpose, we set up an experiment 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 the predefined amount of memory. We also activated swapping 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.

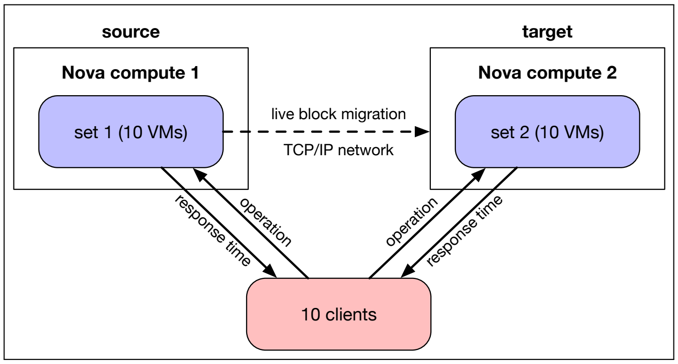

Figure 1. Overview of the methodology of the experiment.

The performance of each VM will be measured by 10 clients, each sending HTTP requests to the VMs in a fixed time interval. Additionally, we produce extra load 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 (stands for R. Rivest, A. Shamir and L. Adleman, see [28]) 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. We chose block live migration 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, we used the 1 GBit/s NICs to better reinforce small effects of different migration parameters and changes. Furthermore, we varied the number of concurrent migrations 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 1 shows an overview of the methodology.

Besides several configurations that were necessary to implement a true live migration in OpenStack [23], 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. The experiment was performed using a script and was repeated 10 times. After changing a parameter in the experiment (e.g., the memory workload shown in Figure 2) it was run 10 times again. All runs led to reproducible results.

B. *Research Results and Discussion*

Figure 2 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 we reduce the number of VMs migrated concurrently. The source of this impact is the overhead of nova-scheduler handling the migration requests.

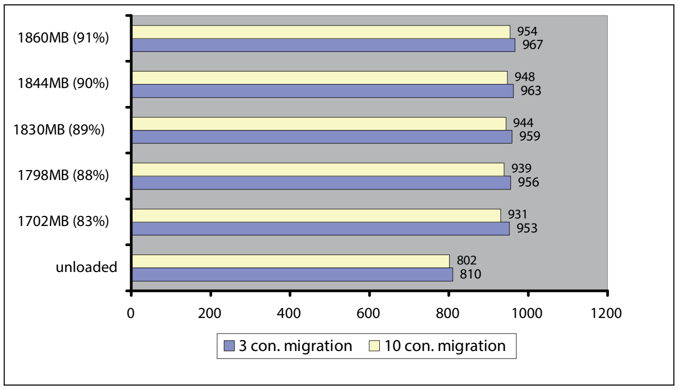

Figure 2. Total migration time (in seconds).

During the migration process, we observed that the performance for search operations within the VMs degrades significantly starting from 1830 MB loaded-memory (89% of total memory size). This degradation is shown in Figure 3, 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

3 during the migration denotes the switch process that was explained in Section IV-B.

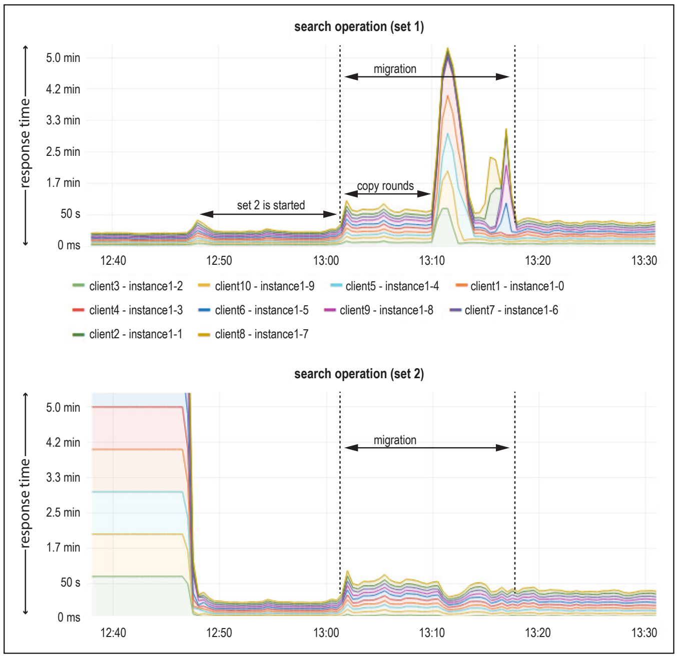

Figure 3. Performance of search operations with a memory workload of 1830MB on Set 1 and Set 2 before, during and after the migration.

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 3, we can recognize that the performance of Set 1 for 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.

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

VM set	*Increased response time during migration (s)*			*Increased response time after migration (s)*		
	1830M	1844M	1860M	1830M	1844M	1860M
Set 1	3.307	4.389	6.241	1.655	2.527	3.431
Set 2	2.712	3.678	3.422	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 we perform block live migration with separate operations, 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. CONCLUSIONS 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 years. This includes adaptive migration, e.g., to consolidate or enhance the utilization of physical resources, as well as long-distance migration, that is not only covered by the related work and research presented in this paper, 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 renewable energy at distant data center sites.

Based on our previous research projects in this area, in this paper we present 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 (i.e., virtual switch or kernel enhancements, data plane development kit - DPDK, SR-IOV). 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, we used additional tools (i.e., *stress*, *openssl*, *dd*, *find*) 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 paper, 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 utilizes 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 small 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 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 [5] 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 paper, 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 [22]) 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.

REFERENCES

[1] Greenberg, A, Hamilton, J, Maltz, D A, and Patel, P, “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.

[2] 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.

[3] A. Andreyev, “Introducing data center fabric, the next-generation Facebook data center network,” 2014, URL: https://code.facebook.com/posts/, 2017.06.07.

[4] 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, 2017.07.31.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

[12] D. Versick, D and D. Tavangarian, “CAESARA-combined architecture for energy saving by auto-adaptive resource allocation.” in DFN-Forum Kommunikationstechnologien, 2013, pp. 31–40.

[13] B. Heller et al., “ElasticTree - Saving Energy in Data Center Networks.” NSDI, 2010.

[14] T. Huong et al., “Ecodanereducing 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] W. Hu et el., “A quantitative study of virtual machine live migration.” CAC, 2013.

[21] K. Rybina, A. Patni, and A. Schill, “Analysing the Migration Time of Live Migration of Multiple Virtual Machines.” CLOSER, 2014.

[22] 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, 2017.06.07.

[23] O. Found., “Openstack administration guide,” 2017, URL: http://docs.openstack.org/admin-guide-cloud/index.html, 2017.06.07.

[24] C. Clark et al., “Live Migration of Virtual Machines.” NSDI, 2005.

[25] 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.

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

[27] Microsoft, “Virtual machine live migration overview,” 2015, URL: https://technet.microsoft.com/en-US/library/hh831435.aspx, 2017.06.07.

[28] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.” Commun. ACM, vol. 21, no. 2, 1978, pp. 120–126.