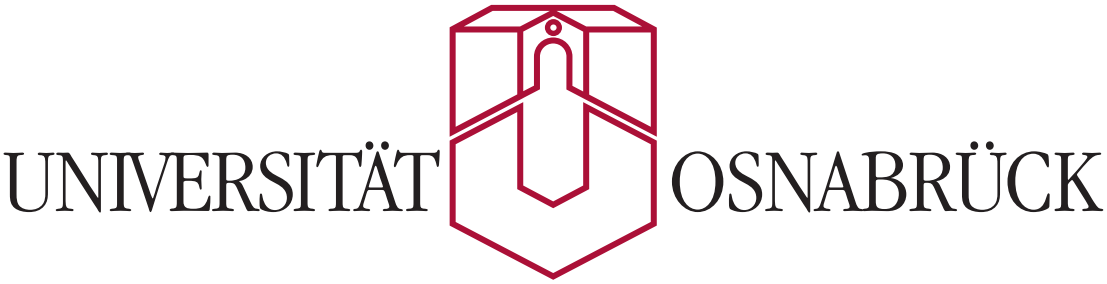

Institut für Informatik
Arbeitsgruppe Verteilte Systeme

Seminar Green Networking

Server Consolidation Techniques in Virtualized Data Centers

Leon Richardt
Matrikelnummer

Wintersemester 2019/2020

19. Dezember 2019

Contents

1 Introduction

In recent years, companies such as Google [Goo19], Amazon [Ama19], and Microsoft [Mic19] have continually expanded their cloud computing platforms in order to keep up with an increasing demand for faster and more reliable online services. They advertise scalability and stability for services running on their platforms. Since energy usage by data centers kept rising during the early 2000s [BMN+08, pp. 25–32] (roughly doubling over the course of five years), cloud computing providers were incentivized to develop solutions that keep costs low but reliability and performance high. These two goals seem contradictory at first: An obvious way to achieve reliability is the introduction of redundancy which in turn leads to an increase in power consumption. Indeed, studies have found that average server CPU utilization is between 10% and 50% for about 80% of the time [BH07]. Server virtualization and consolidation techniques provide possibilities to address these issues and find a solution that is energy-efficient as well as financially feasible [VG17].

In this paper, we give an overview on how these methods can be applied to large-scale systems. To start off, Section 2 gives a brief introduction to the basic model underlying virtualized data centers. Section 3 presents a number of different system parameters consolidation strategies could optimize for. Finally, Section 4 summarizes the obtained results and gives an outlook on possible future developments.

2 Principles

In this section, we lay out some of the fundamentals required to understand the idea of saving power through server virtualization. Furthermore, we introduce the notion of virtual machines (VMs) and how they can be used to efficiently run a large number of workloads on a smaller number of physical machines (PMs).

A key idea for saving power through server consolidation lies in reducing the number of idle machines in the data center. A naive approach might consider running all tasks on a single system. However, this is not a viable practice in every scenario; often due to security sensitive applications which require stronger isolation. VMs provide a possible solution to this problem.

2.1 Power Consumption in Data Centers

According to [IEA19], the global power consumption in data centers amounted to 198 TWh in 2018 which is about 1% of total global electricity demand. Although data center workloads are expected to triple by 2021, power consumption is projected to stay constant over the same period of time. This development can be attributed to advances in software and hardware components. One software solution, virtualization, will be further discussed in this paper.

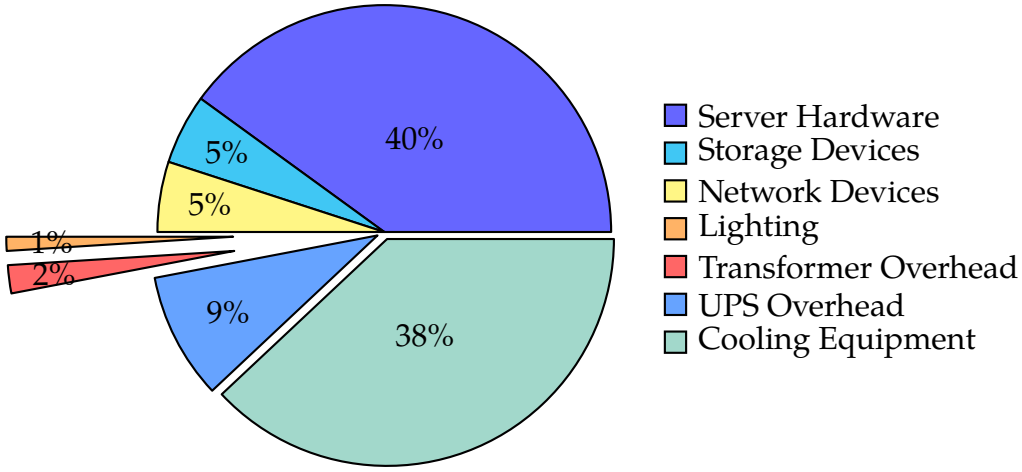

Figure 1: Average distribution of power consumption by different data center components, see [MBS+11].

In data centers, server hardware is not the only factor requiring power. For example, a significant percentage of power usage is accounted for by the cooling system. Figure 1 presents a breakdown of the most relevant elements.

Server Hardware Closely attached ("internal") server components, such as the CPU and memory.

Storage Devices External storage devices, generally consisting of hard disk drives (HDDs) and solid-state drives (SSDs).

Network Devices Devices supporting the local network layout, like routers or switches.

Lighting Although a comparably small point, lighting equipment in data centers must also be considered.

Transformer Overhead To generate the required alternating voltages, transformers may need to be used. In doing so, however, power loss will occur. Masanet *et al.* [MBS+11] assume an efficiency rate of 95%.

UPS Overhead Uninterruptible power supplies (UPS) are designed to protect the data center equipment from sudden power outages. They provide power from battery until the main circuit, or a backup system, can be restored. In [MBS+11], this redundancy is assumed to incur a power loss of about 20%.

Cooling Equipment The largest auxiliary consumer of energy is the cooling infrastructure. It includes equipment such as air conditioners, coolant pumps, fans, and water chillers.

Power consumption can therefore be modelled as a function of the energy usage of server hardware, storage devices, network devices, and infrastructure equipment.

2.2 Virtual Machines

Popek and Goldberg [PG74] define a VM to be "an efficient, isolated duplicate of the real machine". In practice, this property concerns the ability to execute any program that can also be executed by a suitable unvirtualized machine. VMs are created and managed by a virtual machine monitor (VMM): a piece of software that is responsible for distributing the available physical resources (e.g. CPU time, memory, or network bandwidth) to the hosted VMs. Some authors also use the term *hypervisor* instead of VMM.

A distinction must be made between Type-1 (*native, classic system*) and Type-2 (*hosted*) VMMs. While Type-1 VMMs run directly on the host's hardware, Type-2 VMMs run on top of the host operating system. It follows that Type-1 VMMs must implement their own hardware-specific drivers whereas Type-2 VMMs can rely on the host operating system to provide such functionality. An example for a native VMM is Xen [Xen19]; an example for a hosted VMM is VirtualBox [Vir19]. Some software, such as KVM [KVM19], cannot easily be classified into one of these two categories.

2.2.1 VM Migration

A key feature of VMs is the possibility to migrate an existing VM to another physical machine. In order to perform a successful migration, different aspects of the VM image need to be considered:

1. The *internal state*. This includes the current state of the virtual CPU and memory as well as networking and storage adapters.

2. The *external state*. This item concerns the VM's use of external devices, such as USB devices, removable media or networking equipment.

One possible procedure to migrate a VMs is described by Clark *et al.* [CFH+05]. It is designed to incur minimal overhead on the overall system performance:

1. Select a source VM and a destination PM (*target*) to migrate to. (Potentially aided by certain optimization criteria, described in section 3.)

2. Iteratively copy memory to the target. Since transferring the memory state usually has the biggest impact on performance, this is done while the VM is still running. Memory pages that are modified after they have been copied are marked to be sent again.

3. Stop the VM and transfer all non-memory state. Send any pages that are still marked for retransmission.

4. Resume the VM image on the target.

The technique used in Step 2 is known as *iterative pre-copying*. At the beginning of the migration process, many pages need to be copied to the target. Memory pages that

have been modified during an iteration are sent to the target again. Assuming typical workloads, increasingly less pages will have been modified in each iteration (since each iteration completes quicker than the previous one). This also means that the amount of differing memory pages between source and destination is reduced in each iteration. Pre-copying is stopped once a sufficiently low threshold of deviation has been achieved. Alternatively, the process may also be stopped when no significant reduction in the number of pages to transfer can be achieved.

By migrating VMs from a large number of moderately busy PMs to a smaller number of heavily loaded PMs, idle machines can be put into low-power modes or shut down entirely. This can provide significant power savings.

3 Optimization Parameters

As mentioned above, VM migration provides an useful mechanism to balance workloads across multiple PMs. One can imagine numerous ways to measure workloads: CPU utilization, IO utilization, or service reliability could come to mind. This section presents a number of parameters to consider when selecting source VMs and target PMs for migration.

3.1 Hardware Utilization

In [GRCK09], Gmach *et al.* introduce a load balancing model and policies that optimize CPU and memory utilization in server pools. A PM is defined to be overloaded if CPU or memory load exceed a previously specified maximum threshold. In contrast, the *system* is said to be underloaded if the average CPU and memory usage stay below a certain minimum threshold.

The authors describe a number of rebalancing policies a migration manager can employ:

Reactive A reactive policy responds when a violation of either threshold is detected.

When an overload situation occurs, the migration manager determines a migration candidate on the overloaded server. That VM is then transferred to the least loaded server with adequate resources for the workload. If no suitable PM for the workload exists at the time of migration, a new server is started.

In an underload scenario, the migration manager determines the least loaded PM and tries to transfer all running VMs in such a way that no server will be overloaded after the migration.

Trace-Based With this policy, the migration manager will periodically use historical data to predict the expected workload for the next interval. At the start of every interval, traces of past workloads are considered to construct a workload distribution fitting the historical data. Current system load is not taken into account. In [GRCK09], the authors decided to select a 4 h rebalancing interval for their case study.

Reactive/Trace-Based This policy combines the previous strategies: Traces are used to rebalance the workload at the start of every interval. In the case of a threshold violation, the migration manager is invoked to resolve the situation as described in the Reactive policy.

Reactive/On-Demand Balancing This policy works like the previous approach. However, the trace-based rebalancing is *not only* applied periodically but also whenever servers are being under-utilized.

The authors measure the effectiveness of these policies by counting the CPU/memory threshold violations per hour. Clearly, the "success" of these policies is dependent on the threshold values selected by the data center operators. When operating on strict thresholds (high underload and low overload limits), the pure Reactive and pure Trace-Based policies produce the most violations. Especially the Trace-Based migration policy is prone to violations since deviations from the traces will not lead to a change in allocation behavior. Therefore, employing a purely Trace-Based policy only makes sense when the workload adheres to strict periodic patterns.

The combined approaches seem to be much more robust: They can leverage historical information to overprovision PMs for the next interval. In case the trace data still turns out not to be applicable to the current workload scenario, the migration manager may resolve violations by performing further migrations.

3.2 Network Traffic

Meng, Pappas, and Zhang [MPZ10] assert that modern data center applications are increasingly communication-intensive whereas existing VM placement strategies solely base their decisions on factors such as CPU and memory utilization.

In order to study the problem, they assume the existence of n VMs and n slots (one slot represents the ability to place a VM on a PM). They further assume that the number of slots any given server can host has been decided by "traditional" planning tools, i.e. tools that only consider CPU/memory utilization. Let D_{ij} be the traffic rate between two VMs v_i and v_j, and C_{ij} the communication cost between the slots s_i and s_j. [1] Additionally, an external traffic rate e_i is defined for every VM, as well as an external communication cost g_i between slot s_i and the network gateway.

The problem is then to find a bijective VM-to-slot mapping $\pi\colon [1,\dots,n] \to [1,\dots,n]$ such that the expression

$$\underbrace{\sum_{i,j=1,\dots,n} D_{ij} C_{\pi(i)\pi(j)}}_{\text{Total internal cost}} + \underbrace{\sum_{i=1}^{n} e_i g_{\pi(i)}}_{\text{Total external cost}}$$

is minimal. The authors prove that finding an optimal mapping is NP-hard but also propose the approximation algorithm *Cluster-and-Cut* with time complexity $O(n^4)$.

[1] In [MPZ10], the number of switches between two slots is chosen as a metric for the communication cost.

The general idea of the algorithm is described as follows: Based on their mutual communication rate, VMs are classified into clusters with regard to their traffic rate. Likewise, slots with low intercommunication cost are clustered with the same size. Each VM cluster is then recursively mapped to a slot cluster. The algorithm aims to place VM pairs with high traffic between them onto slot pairs with low connection costs.

The authors find that the effectiveness of the algorithm is dependent on the network topology used in the data center. They claim that topologies implementing load balancing, such as VL2 [GHJ+11], do not profit as much as topologies that are load-ignorant. According to an experimental case study conducted in [MPZ10], the placement results of Cluster-and-Cut roughly lead to a 10% improvement over two reference strategies, while also running faster. This could be a major advantage when short migration intervals are desired.

3.3 Thermal Efficiency

As described in Section 2.1, cooling systems are major power consumers in modern data centers. Hence, some placement strategies are concerned with minimizing the required cooling effort (and thus, power consumption). A naive approach might decide to place newly created VMs on those servers currently observing the minimal temperature. However, Tang, Gupta, and Varsamopoulos [TGV07] argue that this course of action is not ideal. Instead, they propose to minimize the total *heat recirculation* inside the system. Heat recirculation occurs when exhausted (hot) air from a server's ventilation outlet mixes with the newly supplied cold air from the ventilation system. As a result, hotter air than originally supplied may enter the server inlets. Since the temperature of this mixture is hard to predict, the ventilation system needs to cool down the supplied air further in order to provide a "buffer" for local hotspots.

In [TGV07], the authors show that minimizing heat recirculation is equivalent to minimizing the maximum inlet temperature of all servers. In particular, this relationship is modelled by the expression

$$T_{\text{in}} = T_{\text{sup}} + \underbrace{D(a, a, \dots, a)^{\top}}_{\text{Idle temperature}} + \underbrace{b \cdot D\,(c_1, c_2, \dots, c_n)^{\top}}_{\text{Workload temperature}},$$

where

- $T_{\text{in}} = \left(t_{\text{in}_1}, \dots, t_{\text{in}_n}\right)$ is the vector of inlet air temperatures for each server,
- $T_{\text{sup}} = \left(t_{\text{sup}_1}, \dots, t_{\text{sup}_n}\right)$ is the vector of air temperatures supplied by the ventilation for each server,
- D is a matrix describing the data-center-specific temperature circumstances,
- $(c_1, \dots, c_n)$ is the vector describing the load of each server i,
- and b is the "energy consumption per load" of a server (assumed to be constant for all servers).

Therefore, the aim is to find a load distribution $(c_1, \ldots, c_n)$ such that

$$\min_{(c_1,\ldots,c_n)} \left\{ \max_{i=1,\ldots,n} \left\{ t_{\text{in}_i} \right\} \right\}$$

(the minimal peak temperature occurring in any load distribution) is reached.

They describe a genetic algorithm that approximates an optimal load distribution. As a fitness function, the peak inlet temperature of all VMs is used. Mutation is done by rebalancing the load distribution of a solution; crossover is done by exchanging a subset of VM assignments. Based on simulations conducted with this algorithm, the authors find a 20% to 30% reduction of energy cost compared to other algorithms at moderate load profiles.

3.4 Evaluation

The above results seem reasonably promising for data center operators. However, there are some problems and open questions that still need to be answered.

Applicability It might not be trivial to decide whether a given optimization criterion is applicable to the expected workload of a data center. The Cluster-and-Cut algorithm, for example, can only be sensibly used when VMs in the same data center communicate with each other. While this may often be the case with high-performance computing applications, cloud computing providers like Google and Amazon will likely not profit from the metric.

Experimental Data A common flaw in the studies and papers presenting new load balancing algorithms is the setting in which they were tested. Most of the time, simulations were run against traces of past workloads. Thus, the behavior of the policies in real-world data centers cannot truly be evaluated.

This is understandable from a business standpoint: Putting production-critical applications at the risk of being compromised by untested load balancing algorithms is a tough sell. Minutes of downtime or service degradation can already lead to considerable financial damage.

Balancing Parameters Different optimization parameters may be contradictory. For example, a hardware utilization-based and a network traffic-based algorithm can produce completely incoherent VM placements. There is no "one-fits-all" model that could be applied to arbitrary workload profiles. Especially large data centers have lots of local parameters and conditions to consider. Development of a unified model is still an open problem; if it can be done at all [MPZ10].

Moreover, data center operators must balance the energy saving potential with existing service-level agreements (SLAs) among their customers. This aspect is examined more closely in [GRCK09].

4 Conclusion & Outlook

In this paper, we gave an overview of the power requirements in contemporary data centers and introduced the notion of a virtual machine. We found that many factors contribute to high energy consumption. In response to these observations, load balancing strategies concerning different optimization parameters were evaluated. We assess that VM migration is a useful tool to effectively distribute tasks among a number of servers. In any case, quality of service must be maintained during VM consolidation.

However, attention must be paid to the specific conditions present at a data center. As demonstrated in Section 3, many of the introduced balancing techniques heavily rely on special characteristics featured in a data center, e.g. the network topology, or the workloads running on the servers. Nevertheless, some strategies may be applied more broadly. For example, cooling considerations are likely relevant in almost every data center.

Comparing the performance of different migration strategies against each other is difficult since authors benchmark their implementations on different workloads and different hardware. Hence, judging the presented techniques solely by the results attained in their original papers will likely not produce a representative ranking. An open problem is the development of a joint system model considering many optimization parameters in parallel.

A recent development regarding scalable data center operation are containerized applications, managed by tools like Kubernetes. These systems also allow for task isolation and load scaling, while incurring less of an overhead than VMs.

Since cloud computing is projected to continue growing over the near future [IEA19], one can expect load balancing and task placement technologies to stay relevant and be improved on over the next few years.

Appendix

List of Abbreviations

HDD hard disk drive

SLA service-level agreement

SSD solid-state drive

UPS uninterruptible power supply

VM virtual machine

VMM virtual machine monitor

PM physical machine

References

[Ama19] Amazon. (Dec. 2019). Amazon elastic compute cloud documentation, Amazon EC2, [Online]. Available: https://docs.aws.amazon.com/ec2/index.html.

[BH07] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007, ISSN: 1558-0814. DOI: 10.1109/MC.2007.443.

[BMN+08] R. E. Brown, E. R. Masanet, B. Nordman, W. F. Tschudi, A. Shehabi, J. Stanley, J. G. Koomey, D. A. Sartor, and P. T. Chan, "Report to congress on server and data center energy efficiency: Public law 109-431," Berkeley, CA, Jun. 2008.

[CFH+05] 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*, ser. NSDI'05, Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286.

[GHJ+11] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," *Commun. ACM*, vol. 54, no. 3, pp. 95–104, Mar. 2011, ISSN: 0001-0782. DOI: 10.1145/1897852.1897877.

[Goo19] Google. (Dec. 2019). Cloud computing services, Google Cloud, [Online]. Available: https://cloud.google.com/.

[GRCK09] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource pool management: Reactive versus proactive or let's be friends," *Computer Networks*, vol. 53, pp. 2905–2922, 2009. DOI: 10.1016/j.comnet.2009.08.011.

[IEA19] IEA. (2019). Data centres and data transmission networks – tracking buildings – analysis, [Online]. Available: https://www.iea.org/reports/tracking-buildings/data-centres-and-data-transmission-networks (visited on 12/15/2019).

[KVM19] KVM Project. (Dec. 2019). KVM, [Online]. Available: https://www.linux-kvm.org (visited on 12/05/2019).

[MBS+11] E. R. Masanet, R. E. Brown, A. Shehabi, J. G. Koomey, and B. Nordman, "Estimating the energy use and efficiency potential of u.s. data centers," *Proceedings of the IEEE*, vol. 99, no. 8, pp. 1440–1453, Aug. 2011, ISSN: 1558-2256. DOI: 10.1109/JPROC.2011.2155610.

[Mic19] Microsoft. (Dec. 2019). Microsoft azure cloud computing platform & services, [Online]. Available: https://azure.microsoft.com/en-us/.

[MPZ10] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *2010 Proceedings IEEE INFOCOM*, ISSN: 0743-166X, Mar. 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5461930.

[PG74] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," *Commun. ACM*, vol. 17, no. 7, pp. 412–421, Jul. 1974, ISSN: 0001-0782. DOI: 10.1145/361011.361073.

[TGV07] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Thermal-aware task scheduling for data centers through minimizing heat recirculation," in *2007 IEEE International Conference on Cluster Computing*, ISSN: 2168-9253, Sep. 2007, pp. 129–138. DOI: 10.1109/CLUSTR.2007.4629225.

[VG17] A. Varasteh and M. Goudarzi, "Server consolidation techniques in virtualized data centers: A survey," *IEEE Systems Journal*, vol. 11, no. 2, pp. 772–783, Jun. 2017. DOI: 10.1109/JSYST.2015.2458273.

[Vir19] VirtualBox. (Dec. 2019). Oracle VM VirtualBox, [Online]. Available: https://www.virtualbox.org/ (visited on 12/10/2019).

[Xen19] Xen Project. (Dec. 2019). Xen, Xen Project, [Online]. Available: https://xenproject.org/ (visited on 12/04/2019).