Virtual Machine Re-Assignment Considering Migration Overhead

Thomas Setzer and Andreas Wolke
Decision Sciences and Systems
Technische Universität München
85748 Garching, Germany
Email: setzer, wolke@in.tum.de

***Abstract*—We introduce a mathematical model formulation for scheduling virtual machine (VM) migrations in data centers. The model is aimed at minimizing the number of active physical host servers over time. Our goal is not only to avoid overload situations resulting from aggressive consolidation mechanisms but also overload situations caused by overhead-intenisve VM migrations. Although various VM scheduling approaches have been proposed in the literature, so far predictable resource demands caused by VM migrations are not directly considered in mathematical scheduling models, which can easily entail unplanned overload situations and resulting performance degradation. We propose a new model formulation, which explicitly takes the migration overheads into account while recalculating and executing VM allocations over time. Experimental results based on VM resource demand time series from a data center show that the model allows for significant server savings compared to a static assignment of VMs to physical hosts, while avoiding overload situations.**

***Keywords*-VM Migration, Server Consolidation, Cloud Infrastructure Management, Virtualization Management**

I. Introduction

Energy usage has become one of the key cost drivers in data centers accounting for 30-50% of the total operational costs. Servers and the cooling systems are considered to be responsible for a large portion of the energy consumption. As a consequence, energy consumption of servers and cooling equipment gets into the focus of data center operators. One obvious approach to save energy is to increase the overall resource utilization of servers while at the same time reducing the number of active servers.

Large data centers run thousands of business applications with varying resource demand. In general, server applications are increasingly hosted in virtual machines (VM), which allows for server consolidation by hosting multiple VMs jointly on the same physical server. As multiple VMs share a server's physical resources, it is possible to increase the overall utilization and reduce the number of required servers.

For static server consolidation based on fixed assignment of VMs to physical servers, mathematical problem formulations have been proposed. For instance, assuming the availability of predictable VM workload profiles, Speitkamp et al. modelled the problem using a multidimensional bin-packing formulation (MBP) [4]. The authors have shown that the consideration of daily workload cycles using a MBP formulation can lead to 31% savings in physical servers compared to simple heuristics. Such mathematical approaches to static server consolidation have been successfully applied to reduce the total number of servers required in a data center [3], [4], [5]. Here, VM to server assignments are calculated for an extended period of time such as weeks and months including holidays, weekends and night time. However, resource demands of enterprise applications are typically much lower during these periods and a smaller set of servers would be sufficient to host the VMs.

Amongst others, Setzer et al. [9] show that for example during night time only one out of three servers is required to host a set of VMs compared to the number of servers required during working hours of a day. Unfortunately, such static allocation approaches do not support a variation in the number of servers over time.

Besides providing the foundation for server consolidation, virtualization allows for VM-reassignments during runtime by live migration [8]. Therefore, on the one hand, the technical foundation for implementing adaptive VM assignments is available and a server may be deactivated after migrating away all hosted VMs. On the other hand, each migration entails additional resource overheads on the source system as well as on the target system, especially regarding CPU and memory capacity [10]. This is because the live algorithm used by all major hypervisors is based on the tracking of memory write operations and memory transfers over the network that requires significant CPU and network capacity [11]. Depending on the frequency a business application writes data into the main memory, the CPU overhead can easily exceed 30% of the application's current CPU demand [11].

Due to the huge impact of migration overhead, VM reallocations must be scheduled in a controlled fashion in order to avoid server overloads for the time of a live migration. In particular, migrations are only allowed to be triggered if there are enough resources available. In this paper we introduce an Integer Program considering migration overhead to describe the VM migration scheduling problem. Numerical experiments are presented showing that the model can lead to significant energy server savings over time.

The remainder of this article is structured as followed. In the next section we briefly review related work on VM reassignment. In Section III we introduce the scheduling model. In Section IV we present experimental outcomes. Finally, Section V concludes and discusses the impact of this work.

II. Related Work

Basically, two approaches for VM-reassignments have been proposed in the literature: reactive, feedback-driven control techniques and pre-determined migration periods. Reactive control techniques for VM-reassignment decisions are introduced, amongst others, by [5]. Here, ad-hoc migrations are triggered using a fuzzy controller operating on predefined utilization thresholds on host servers. If thresholds are exceeded, a VM is moved to another server in order to avoid an overload on that particular server. The intuition of threshold-based techniques is to avoid overload by leaving a "safety capacity buffer" on each server, large enough to guarantee overload avoidance even when migrations are triggered. The downside of such approaches is that rather volatile workload behavior with frequent peaks might lead to many costly reallocations.

Therefore, IT managers often prefer to search for stable allocations, which minimize the number of servers and do not require high-frequent migrations. Gmach et al. [12], amongst others, proposes a cyclic re-computation of VM assignments after periods of some hours in order to further reduce the required overall capacity and align to workload changes. Longer cycles such as days, weeks and months have been proposed by Rolia et al. [13] or Bichler et al. [14]. Shorter cycles allow for more frequent alignment and faster evacuations of hosts in phases of low demand, but might lead to a large number of unnecessary migrations and even oscillations of VMs entailing the risk of overload situations.

The proposed approaches have in common that migration overhead is considered only in an indirect fashion by either defining threshold values resulting in (sufficiently) large safety buffers or in restricting migrations to be allowed only after certain periods of time. However, assuming deterministic or at least a well predictable VM workload behavior, migration overheads as well as migration durations can be estimated with high accuracy as done in paper [11].

In this paper we introduce a mathematical optimization model which aims to reduce the number of server hours while avoiding overload situations due to live migrations. The server hours metric is used as a proxy for total energy costs and describes the accumulated runtime in hours of all servers in a data center. The model does not rely on static, predefined thresholds or migration cycles but determines a migration schedule based on reliable workload predictions.

III. Model Formulation

We will now formulate an Integer Program (IP) aimed at minimizing the total amount of server hours by determining a VM migration schedule. The model assumes well predictable future workload behavior of the VMs. This obviously is not the case for all VMs but for the majority of VMs that run operational business applications. Such VMs with reliable workload behavior are considered in this paper. The IP considers migration overheads in order to avoid migration-related overload situations.

To allow for a formal discussion, let us first introduce some notation. Suppose that we are given J VMs $j = 1, \ldots, J$ to be hosted by I servers $i = 1, \ldots, I$ or fewer. Different types of resources $k = 1, \ldots, K$ such as CPU, I/O, or main memory may be considered and each server has a certain capacity s_{ik} of each resource k. The planning period is divided into τ discrete time periods t, $t = 1, \ldots\tau$. These intervals might be for example one or five-minute intervals. y_{it} is a binary decision variable indicating whether server i is used in interval t or not, c_i describes the operational costs (energy costs) of a running server per time interval, and the binary decision variable x_{ijt} indicates which VM j is allocated to which server i in a time interval t. Let r_{jkt} be the capacity that VM j requires of resource k in time period t, and let m^-_{jkt} (m^+_{jkt}) denote the absolute migration-related overhead of demand for resource k on a source server (target server) when migrating j in t. In order to account for different migration durations of VMs m^-_{jkt} is multiplied by a factor of the estimated migration duration relative to the length of a time period. Hence, $\sum_{r_{jkt}+m^-_{jkt}}$ is j's total demand for k in t during j's migration process on the source server, and m^+_{jkt} on the target server. The consideration of migration overheads is vital, as for example CPU overheads on the source as well as on the target server may easily exceed 30 or even 40% of a VM's current CPU load the moment before a migration is triggered. Ignoring overheads would therefore result in unwanted situations of overload and associated performance degradation. The binary decision variable z^-_{ijt} is true if j is migrated away from i in t. The binary decision variable z^+_{ijt} is true if j is migrated to a server i in t.

$$
\begin{array}{lcll}
\min \sum\limits_t \sum\limits_i c_i y_{it} & & & \\
s.t. & & & \\
\sum\limits_i x_{ijt} & = & 1 & \forall j, \forall t \\
\sum\limits_j r_{jkt} x_{ijt} + m^-_{jkt} z^-_{ijt} & & & \\
\quad + m^+_{jkt} z^+_{ijt} & \leq & s_{ik} y_{it} & \forall i, \forall k, \forall t \\
-x_{ij(t+1)} + x_{ijt} - z^-_{ijt} & \leq & 0 & \forall i, \forall j, \forall t < \tau \\
x_{ij(t+1)} - x_{ijt} - z^+_{ijt} & \leq & 0 & \forall i, \forall j, \forall t < \tau \\
y_{it}, x_{ijt} & \in & \{0,1\} & \forall i, \forall j, \forall t \\
z^-_{ijt}, z^+_{ijt} & \in & \{0,1\} & \forall i, \forall j, \forall t < \tau
\end{array}
\tag{1}
$$

The resulting deterministic problem considering CPU overhead can now be formulated as shown in (1). The objective function minimizes the total operational server costs. The first set of constraints ensures that each VM is allocated to one of the servers at any point in time. The second set of constraints ensures that the aggregated resource demand of multiple VMs does not exceed a server's capacity for all resource types and during all time intervals. The term on the left-hand side of the sum describes the resource demand due to the operation of VMs. The term on the right-hand side determines migration-related overheads. With the third and fourth constraint type the slack-variables z^-_{ijt} and z^+_{ijt} used in the second set of constraints are set to one if a VM is migrated.

In the next section we will evaluate this model using available VM workload traces from a real world data center.

IV. Evaluation

In this section, we study the solution quality in terms of energy savings realized by the proposed scheduling model using the best possible static assignment as a benchmark. We compare the model against static assigments as all approaches that allow for VM migrations found in the literature do not guarantee overload avoidance even in case of deterministic VM workload behavior.

Given deterministic workload, the static assignment of VMs is equal to the largest amount of required, active servers. It can be calculated by using a multi-dimensional bin packing (or vector packing) approach as done in [6]. We consider scenarios with different sets of VM workload traces to be assigned to physical servers. Before discussing the main outcomes, we describe the data set used in the experiments and the research design.

A. Experimental Design and Available Data

The data provided by a large data center operator describes the CPU utilization of 450 VMs measured in intervals of five minutes over multiple consecutive weeks. The business applications hosted in the VMs stem from a diverse set of customers and include different server types such as application servers, web servers or SAP modules.

Similar to [4] and [15], we found that the majority of the applications have rather regular and cyclic diurnal workload patterns. Hence, we consider a planning period of one day, as the workload patterns are repeated in this interval to a great share. The dataset used in this work has been described intensively by Speitkamp and Bichler [4].

We mainly focus on *operational server hours* as a metric for solution quality of the VM assignment model. We will explore the impact of different covariates such as different migration overheads, which in practice depend on the specific application and it's workload characteristics. In our numerical experiments, we consider CPU overhead on the (source, target) servers of (20%, 10%), (30%, 20%), and (40%, 30%). In addition, we analysed the impact of the CPU capacity of the host servers, starting with small servers (*S*) which could host 3 VMs on average, to medium servers (*M*) which where able to host around 5 VMs on average, to large servers (*L*) hosting around 8 VMs on average. In summary, we conducted 120 experimental runs (ten different, arbitrarily selected VM sets for each of the three server types and the three different overhead-tupels). Each setup was solved with the static VM assignment model and the proposed VM scheduling model.

B. Results

Aggregated results of our experiments are shown in Figure 1. The dark-gray bars show the average percentage of energy costs with VM migration scheduling compared to the energy costs achieved with the best static allocation with the same set of VMs (scenario). Each scenario has been computed with the three different migration overheads. For example, the height of 45% of the outer-left bar indicates a reduction of 55% of energy costs on average when applying the VM migration scheduling model in scenarios with 20% CPU overhead on the source, and 10% CPU overhead on the target servers. The light-grey bars show the standard-deviation of the percentage of energy costs with the proposed model.

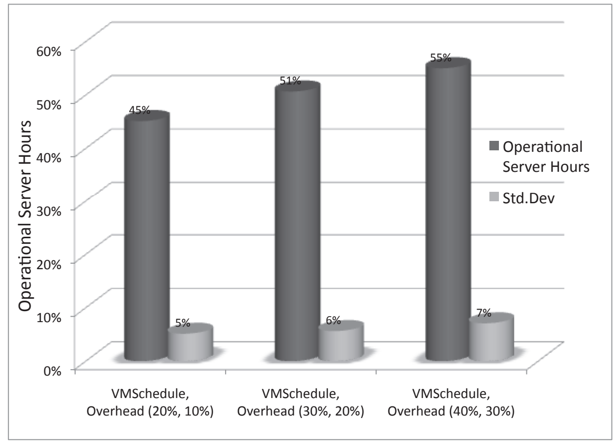

Fig. 1. Aggregated Results Depending on Migration Overhead

Overall, VM migration scheduling resulted in energy cost reductions of around 55% compared to static allocations if the migration overhead was low, of 49% with medium migration overhead, and up to 45% with a high migration overhead. We compared these results pairwise and found all differences to be significant at a 1% level using a Wilcoxon signed rank test. That is mainly because the standard deviation of the relative energy cost was relatively low, varying between 5 and 7%.

The positive correlation between migration overhead and energy costs results from the fact that with decreasing overhead migrations become cheaper. For example, an overhead of zero would basically mean that VMs can be moved around without restrictions to evacuate host servers whenever possible, while a very large overhead would not permit migrations at all. Accordingly, the average number of migrations increased with decreasing migration overhead. For example, in small server scenarios with high migration overheads, we counted 73 migrations in total, while we had 84 with medium overheads, and over 100 migrations when assuming small migration overheads. However, even for relatively large overheads assumed, almost 45% of operational server hours could be saved with the proposed model.

We will now look at the impact of the server size on the solution quality. Figure 2 shows aggregated experimental outcomes with S, M, and L sized servers.

The bar heights vary only slightly between 49% to 51% between scenarios with different server size. Also, our statistical tests did not find significant differences of relative energy costs depending on the size of the servers. We conclude that the cost savings with the VM migration scheduling approach are around 50% with different sets of VM workloads.

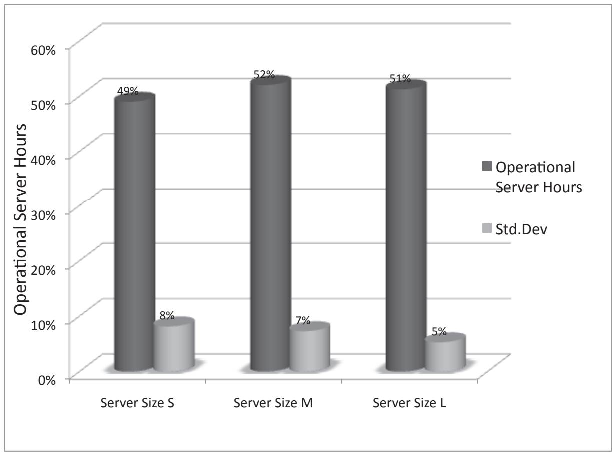

Fig. 2. Aggregated Results depending on Server Size

V. Conclusion and Discussion

We introduced a mathematical model for scheduling VM migrations in a virtualized data center aimed at minimizing the overall energy consumption while avoiding server overload situations due to migration overheads. In literature some work on re-assigning VMs has been discussed, but to our best knowledge no papers consider migration-related overheads directly in mathematical model formulations. Outcomes of numerical experiments based on real world workload traces show that around 50% of operational server hours can be saved using the proposed scheduling model compared to optimal static VM allocations. These outcomes are based on numerical experiments with deterministic workload behaviour of rather small sets of VMs. The outcomes are promising, in particular as the majority of VMs we analysed exhibits a well-predictable, repetitive pattern on a daily and weekly basis. Furthermore, migration overheads can be predicted with sufficient accuracy as long as the workload behaviour of the business application hosted in the VM is known. Hence, the parametrization of the model is possible in many real-world scenarios, which makes the scheduling model a promising tool for IT administrators.

Unfortunately, the problem formulation is NP-complete and only small problems with up to 20 or 30 VMs can be solved within a time frame of one hour. In the future we will work on approximation algorithms to solve larger problem instances of practical relevance. In addition, we set up a prototype of a small data center with 6 host servers, and over 20 VMs running standard industry benchmarks such as SPEC and TPC. Using agents generating variable service demand, in experimental runs of 24-hours duration we found that the approach lead to almost no overload situations (no response time violations). In the future we will process a large set of experiments, including experiments with more stochastic demand behaviour.

References

[1] D. Filani, J. He, and S. Gao, "Technology with the environment in mind," INTEL Corporation., Tech. Rep., 2008.

[2] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, pp. 33–37, 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1339817.1339894

[3] J. Rolia, X. Zhu, M. Arlitt, and A. Andrzejak, "Statistical service assurances for applications in utility grid environments," *Performance Evaluation*, vol. 58, no. 2+3, pp. 319–339, 2004.

[4] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *IEEE Transactions on Services Computing*, vol. 3, no. 4, pp. 266–278, 2010.

[5] S. Seltzam, D. Gmach, S. Krompass, and A. Kemper, "Autoglobe: An automatic administration concept for service-oriented database applications," in *Int. Conference on Data Engineering (ICDE 2006)*, Atlanta, USA, 2006.

[6] M. Bichler, T. Setzer, and B. Speitkamp, "Capacity management for virtualized servers," in *Workshop on Information Technologies and Systems (WITS)*, Milwaukee, USA, 2006.

[7] D. Gmach, S. Krompass, A. Scholz, M. Wimmer, and A. Kemper, "Adaptive quality of service management for enterprise services," *ACM Transactions on the Web (TWEB)*, vol. 2, 2008.

[8] C. Clark, K. Fraser, S. H, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *In Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI*, 2005, pp. 273–286.

[9] T. Setzer, "Decision support for virtual machine reassignments in enterprise data centers," in *IEEE Network Operations and Management Symposium Workshops (NOMS)*. IEEE, 2010, pp. 88–94.

[10] J. Hall, J. Hartline, A. R. Karlin, J. Saia, and J. Wilkes, "On algorithms for efficient data migration," in *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001, pp. 620–629.

[11] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Predicting the performance of virtual machine migration," *Modeling, Analysis, and Simulation of Computer Systems, International Symposium on*, vol. 0, pp. 37–46, 2010.

[12] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource pool management: Reactive versus proactive or let's be friends," *Comput. Netw.*, vol. 53, no. 17, pp. 2905–2922, 2009.

[13] J. Rolia, L. Cherkasova, M. Arlitt, and A. Andrzejak, "A capacity management service for resource pools," in *WOSP '05: Proceedings of the 5th international workshop on Software and performance*. New York, NY, USA: ACM, 2005, pp. 229–237.

[14] M. Bichler, T. Setzer, and B. Speitkamp, "Provisioning of resources in a data processing system for services requested," 2006.

[15] K. Parent, "Consolidation Improves IT's Capacity Utilization," Court Square Data Group, Tech. Rep., 2005.