Network-aware migration control and scheduling of differentiated virtual machine workloads

Alexander Stage and Thomas Setzer
Technische Universität München (TUM)
Chair for Internet-based Information Systems (I18)
Boltzmannstr. 3, 85748 Garching, Germany
{stage, setzer}@in.tum.de

Abstract

Server virtualization enables dynamic workload management for data centers. However, especially live migrations of virtual machines (VM) induce significant overheads on physical hosts and the shared network infrastructure possibly leading to host overloads and SLA violations of co-hosted applications. While some recent work addresses the impact of live migrations on CPUs of physical hosts, little attention has been given to the control and optimization of migration algorithms and migration-related network bandwidth consumption. In this paper we introduce network topology aware scheduling models for VM live migrations. We propose a scheme for classifying VMs based on their workload characteristics and propose adequate resource and migration scheduling models for each class, taking network bandwidth requirements of migrations and network topologies into account. We also underline the necessity for additional migration control parameters for efficient migration scheduling.

1 Introduction

Hosting transaction processing enterprise applications in data centers operating based on a dedicated hosting model has been notoriously afflicted with the physical server underutilization problem. Nowadays server virtualization based workload consolidation is increasingly used to raise server utilization levels and to ensure cost-efficient data center operations.

While static consolidation relies on reliable workload prediction, unforeseen spikes or shifts in workloads require dynamic workload management to continuously align placements of virtual machines (VMs) to avoid server overload. The goal is to achieve continuous, high utilization levels of physical servers while meeting SLAs.

On the one hand, VM live migration realizes dynamic resource provisioning and load balancing, on the other hand it imposes significant overheads that need to be considered and controlled. While some work has been published on load balancing between physical hosts [17] through frequent live migrations, only CPU overheads on source hosts are taken into account. However, multiple consolidated workloads on a physical host require a corresponding multiple of network capacity. Given the current network topologies in data centers [6, 1], designed as multi-rooted trees as depicted in Figure 1, higher workload density in combination with network bandwidth intensive migrations can lead to network contention. Hence, in order to prevent performance degradations, network overheads and network topologies need to be taken into account. The leafs of the network tree

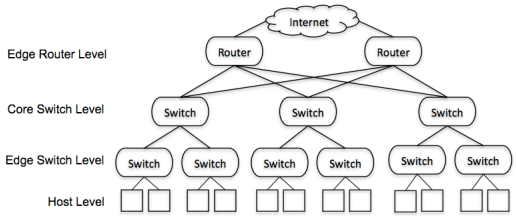

Figure 1. Network topology in data centers

are physical hosts that are linked to edge switches (typically via 1 Gb/s links). Edge switches are further connected to core switches via 10 Gb/s links. Although new switches such as rack integrated edge switches [14]

have been designed to connect hosts directly to core switches via 10 Gb/s links, links to core switches and from core switches to routers are invariably oversubscribed, but suffice since only a few application experience their peak demands in parallel. Increasing oversubscription factors (ratio of the sum of peak demands of applications sharing a link and its capacity) increase the risk for contention. Even though higher bandwidth links/switches are becoming available (100 Gb/s Ethernet [8]), network links will remain oversubscribed due to higher computing density levels at the leafs owed to virtualization.

Xen [2] and VMware [15] are examples for VM monitors that support live migration using iterative, bandwidth adapting pre-copy memory page transfer algorithms [5, 12]. This technique aims at minimizing VM downtime while keeping total migration time low and tries to lower the aggregated bandwidth consumption for a migration. As shown by Clark et al. [5], live migrations can consume significant bandwidth for several seconds (500 Mb/s for 10 seconds for a trivial web server VM). These non-neglectable overheads need to be considered when scheduling migrations, all the more as advances in virtualization also allow for live migrations including virtual disks with even higher bandwidth demands [4].

As an example, we consider a scenario, requiring the execution of 20 VM migrations within 5 minutes since several VMs expose sudden workload increases that would possibly lead to resource shortages. Each migration consumes 1 Gb/s for 20 seconds (2,5 GB transfer volume). Sequentially scheduling them over a single 10 (1) Gb/s link saturates the link completely for 40 (400) seconds, which is clearly unacceptable and emphasizes the need for migration scheduling.

In this vision paper, we introduce VM workload type, network topology and bandwidth requirements aware scheduling and control models for VM live migrations. We propose a workload classification scheme for transaction processing applications and group workloads of the same classes together on a cluster of hosts. For each class, we propose adequate resource and migration scheduling models. Furthermore, we show that for efficient migration scheduling, advanced, currently unavailable, but easily implementable migration control parameters are required.

2 Related Work

Data migration problems have been studied in the context of storage systems [7]. Here, data object re-allocations are triggered by changing data access patterns that require re-balancing of workloads across disks. Based on a target data object allocation derived from predicted workloads, a migration schedule is established that minimizes the total completion time. In contrast to the described and comparable work, we deal with non-uniformly sized data objects, varying migration times and limited bandwidths. Kim [10] also considers non-uniform migration times for different data objects with the objective to minimize the total completion time over all migrations. Lu et al. [11] introduce a control-theoretical approach that aims at dynamically adjusting the speed of data migrations to meet SLAs while maximizing the data migration rate. Again, these approaches do not take data transmission constraints into consideration.

Bichler et al. [3] propose a consolidation model by finding migration-schedules allowing for continous optimized consolidation plans and, thus, aim at minimizing the number of required servers. However, the authors assume predictable workload development for deriving static, but optimal allocations for server consolidation. Migration duration, migration overheads, migration control like bandwidth utilization is not considered.

Khanna et al. [9] discuss server consolidation with a focus on application performance. Resource utilization thresholds are defined to prevent application performance degradation. By reducing the number of necessary migrations and issuing VM migrations that cause low migration costs (estimated from resource utilization), migration overheads should be minimized, without taking the network topology or details of migration algorithm into account. Wood et al. [17] use VM migrations to equalize resource demands between physical hosts, but do not aim at minimizing overall server costs.

To summarize, in contrast to our work, none of the approaches mentioned above addresses the integrated problem of schedule-based resource provisioning for VMs with the goal of reducing the overall server costs and the controlled scheduling of live migrations in order to avoid network congestion.

3 Migration scheduling architecture

One of the goals of migrations is to adapt VM allocations to changes in workloads in order to minimize the number of required servers over time while meeting SLAs. Therefore we propose the architecture depicted in Figure 2, consisting of a VM workload classifier, an allocation planner, a non-conformance detector, and a live migration scheduler that use monitoring data collected

by resource sensors in order to control the data center (control plant) operations. The VM workload classifier

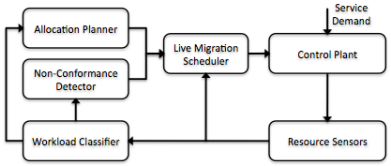

Figure 2. Control system architecture

is needed for efficient workload management. It is advisable to host VMs of a specific workload class in the same server cluster and to apply dedicated control and planning models to each class. Based on the classified workloads (VMs), the allocation planner predicts over- or underload situations that require migrations of VMs and proposes VM re-allocations to other servers on a tactical level. Finally, the live migration scheduler is used to determine operational live migration plans so as to avoid migration-related SLA violations. Additionally, a non-conformance detection component handles unexpected situations such as sudden surges in resource demands of VMs, that would lead to host overload. The main components and their interplay are described in more detail in the following subsections.

3.1 Workload classifier

For workload management and migration scheduling for resource allocation, we identify the following main workload attributes for our classification:

1. *Predictability*: A workload is predictable if its behavior can be reliably forecasted for a given period of time (forecasting errors are tightly bounded).

2. *Trend*: Refers to the degree of upward or downward leading demand trends.

3. *Periodicity*: Indicates the length (time scale) and the power of recurring patterns.

For example, when grouped together in a cluster, predictive, low-variable, low-trend afflicted workloads can be consolidated (co-hosted) more aggressively by exploiting workload complentarities, while highly non-predictive or stochastic ones require certain buffer capacity on hosts so as to guarantee overload-avoidance. Here, migrations might be triggered whenever predefined safety-margins regarding the overall host utilization are exceeded. Trend afflicted workload clusters require more proactive mechanisms aiming at balancing out trends and postpone migrations until a certain utilization threshold is reached in order to guarantee system stability and high host utilization.

To assign a workload to a cluster, it has to be supervised for a period of time, and a class-assignment decision has to be made, which is not the scope of this paper. If a VM exposes changed workload behavior it is reclassified and moved to another class (and sooner or later to a host cluster).

3.2 Allocation planner

Based on the workload and host utilization predictions generated by the workload classifier, the allocation planner determines expected resource bottlenecks as well as low utilization levels.

For predictive workload classes exposing stable or periodic patterns, static and efficient VM allocations plans can be pre-computed for a period of time. By co-hosting VMs with complementary workloads high resource utilization can be achieved while avoiding overload. Consolidation leads to an initial VM allocation (VM to host mapping). VM re-allocation plans, requiring live migrations, are then used during runtime to continuously optimize the VM allocation to further decrease the number of required hosts. Furthermore, additional migrations may be triggered by the non-conformance detection component during runtime if for example a bottleneck due to unexpected workload increase is expected. In such a case, VMs may be migrated to standby or to lower-utilized hosts. This model is targeting rather predictive and moderately volatile and trend afflicted workloads.

For non-predictable workloads, we detect bottlenecks by setting a rather conservative threshold value regarding overall host utilization to avoid overload. If thresholds are exceeded, one or multiple VMs are selected as migration candidates and a request for online scheduling the migrations of these VMs is submitted to the migration scheduler.

For trend-afflicted workload classes which are at least moderately predictable, we proceed in a comparable way by setting host utilization thresholds. However, as this workload class is well predictable, bottlenecks can be anticipated and migration schedules can be precomputed early for longer planning periods.

3.3 Migration scheduler

The migration scheduler applies a global data center view. Requests for migrations are issued by the allocation planner or the non-conformance detector. These

components propose the re-allocation of a VM before e.g. a resource bottleneck occurs. The migration scheduler, aware of migration durations, deadlines and earliest possible starting times, then determines optimal time slots to start the migrations (scheduling) and configures migration control parameters (control). Required migration control parameters are described in the following subsection. Scheduling models are described in detail in the next section.

3.4 Migration control

Pre-copy migration algorithms work iteratively. In the first iteration all main memory pages are transfered, in subsequent iterations, only memory pages are transfered from the source to the target host that have been written to (dirtied) during the previous iteration. Bandwidth usage is adaptively increased in each iteration in order to reduce the amount of pages left to transfer, until the set of dirtied memory pages is sufficiently small or the upper bandwidth limit is reached. This condition is called the iteration termination condition. Once it is met, the last pre-copy iteration is started. The duration of the last iteration determines the service downtime of a VM that is migrated. Clearly, a trade-off exists between service downtime, migration duration and aggregated bandwidth consumption. During a migration, bandwidth usage is increased from iteration to iteration up to a user defined maximum limit. The adaptation rate for each iteration, can not be controlled externally and depends on the memory page dirtying rate during the previous iteration and a fixed increment). The only control parameter of current pre-copy migration algorithms is the maximum bandwidth usage level, that is applied in the last iteration of a live migration.

Currently, a migration can neither be executed at a constant bandwidth level, throttled down nor accelerated by an external controller. Once started, the control over the migration is taken over by the VM monitor. This lack of migration control features is exemplified in Figure 3. There, we assume to execute migrations over a dedicated, bandwidth limited network link and try to minimize the maximum bandwidth usage at each point in time while holding migration deadlines. Live migrations are always non-preemtive. The upper migration schedule shows an offline computed migration schedule with two migration scheduling requests, A and B (C can be neglected for now), that are migrated using a current pre-copy migration algorithm. Both migrations run in parallel and their deadlines (A's is at t_5, B's at t_6) are kept. On the one hand, migration deadlines specify priorities. On the other hand deadlines are required by the

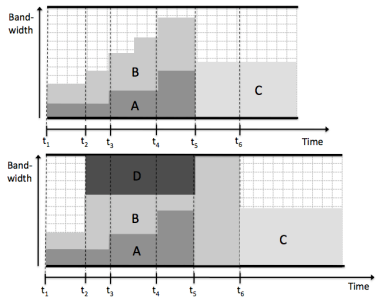

Figure 3. Scheduling example

resource allocation scheduler and the non-conformance detector in order to prevent resource shortages. At t_2 and t_3 for example, the VM monitor increases autonomously the bandwidth usage of migration B and A respectively. By defining the maximum bandwidth usage level a priori, it is possible to obey the bandwidth capacity limits of the used link between t_4 and t_5.

Without advanced control of the bandwidth consumption for all migration iterations, deadlines may not be achievable for high priority migrations, since existing migration algorithms may not adapt the bandwidth usage quickly enough to ensure short migration durations.

In the lower schedule we assume that at t_2 a high priority migration D is requested for online scheduling by the non-conformance detector. It is now required to run A, B and D in parallel. If we can not exercise control over the bandwidth usage of all three migrations (e.g. we can not set a constant bandwidth usage level for D, or throttle down B or A), the link's capacity limits would not allow for completing D until t_5, since the available bandwidth does not suffice for the three migrations in parallel. Therefore the migration scheduler sets a high constant bandwidth usage level for D. Lower priority migration B is throttled down at t_2 and t_3 (the bandwidth usage is controlled for the respective migration iterations). At t_5, B gets a high amount of bandwidth allocated in order to finish in time at t_6. C is now started delayed at t_6.

For efficient migration scheduling, we require to have control over the bandwidth adaptation behavior, the minimum and maximum bandwidth usage and the iteration termination condition. By having control over those migration parameters, various values like the migration duration and the service downtime can be calculated.

In the following, we will exemplarily describe scheduling models when setting a constant bandwidth usage level $b_i > 0$ for all n_i migration iterations of VM $i \in V$ (V is the set of requested migrations). When

setting b_i too low, the migration duration rises and the VMs service downtime becomes high. On the other hand, only a smaller band of bandwidth is used during migration. Let o_i^{min} be the predefined, sufficiently small minimum service downtime of i (due to network latency). i_q denotes the duration of the q-th iteration of i ($q \leq n_i$), m_i the static memory entitlement, r_i the constant memory dirtying rate of VM i. r_i can be monitored and estimated based on observations of past migration iterations. The migration termination condition that determines the start of the last iteration n_i of i needs to be relaxed to allow constant bandwidth usage overall all iterations. The iteration termination condition is met if statement 1 holds.

$$(\frac{i_q \cdot r_i}{b_i} \leq o_i^{min}) \vee (i_{q-1} \cdot r_i \leq i_q \cdot r_i) \qquad (1)$$

Without modification, current iteration termination conditions only allow for two migration iterations of constant bandwidth usage, leading to high service downtimes. The actual service downtime o_i is the duration of the last iteration n_i. By setting $q = n_i$, it can be calculated by equation 2, which requires knowledge of r_i and a fixed b_i. The total duration of i is the sum of the durations of all i_q.

$$i_q = \frac{m_i \ \cdot \ (r_i)^{q-1}}{(b_i)^q} \qquad (2)$$

4 Scheduling Models

Depending on the VM cluster, offline computed schedules for migrations (highly predictable), or online scheduling (or simple heuristics like greedy algorithms) are required. Note that typically we will have a combination of offline and online scheduling for the predictive or trend workload cluster as anomalies and mid-term shifts in workload behavior of VMs need to be handled during a pre-calculated period as well. The latter issues single migrations (migrations requests arrive sequentially) as corrective actions for workload anomalies that require online-scheduling or fast heuristics.

A vast body of work on scheduling problems for different problem domains with various objectives and assumptions exists [13]. In this vision paper we exclude the detailed mathematical variants of the scheduling models due to space constraints. Our aim is to show how advanced migration control can be used, that allow for sophisticated workload management and scheduling models.

4.1 Offline scheduling plans

We start out with a simplified model where an administrator controls the bandwidth that can be used for VM migrations on all links. Here, a fixed available bandwidth on each link is reserved for VM migrations, which frees us from considering stochastically varying bandwidth utilization caused by VM workloads. We allow for different amounts of reservations on different links.

Offline scheduling can be used for predictive VM workload clusters with periodicity or for clusters with trend. The objective of a schedule is to minimize the maximum bandwidth usage on all links for all time slots of a planning period (e.g. one hour), as the goal of migration scheduling is to avoid the risk of overloading network links by migration-related bandwidth consumption. The problem can be understood as a load balancing problem of the overall bandwidth demand on all links of the same level in a network tree, which requires information about the network topology.

In case that there is no reserved bandwidth for VM migration, migrations can only consume available bandwidth capacities on each link. The objective of a scheduling model is to minimize the migration-related risk of network congestions with respect to bandwidth demand fluctuations. As the available bandwidth is not known exactly in advance, and migrations should not use up bandwidth that is required by applications, we exploit the advanced migration control parameters in the following way. We predict the average utilization of network links for all time slots (e.g. via the Network Weather Service [16]). When scheduling, we constantly adjust the bandwidth usable for migrations so as to achieve targeted bandwidth utilization levels of all links. If multiple migrations are using the same link concurrently, the available bandwidth is shared amongst the migrations. However, note that given the uncertainty in future workload behavior, a more conservative available-bandwidth prediction is advisable, increasing with increasing volatility, trend and distribution of workloads using a given link.

4.2 Online scheduling

As already discussed, in particular if a request for migration is issued by the non-conformance detector, the scheduler has to make fast decisions within seconds or minutes. The migrations are revealed to the migration scheduler in an a priori undefined sequence. Migrations can be delayed as long as migration-finishing deadlines (time of expected bottlenecks) are not violated. Note

that a migration might be rejected in case it can not be executed (because running high priority migrations can not be delayed or slowed down) once its presented to the migration scheduler. As shown in Example 3 online scheduling requires dynamic control of active migrations. Emergency migrations may temporarily supersede bandwidth allocations of lower priority migrations. However, a newly arrived request for migration of a VM requires bandwidth on all links along the migration path. Some links might be used by other active migrations. Hence, allocating more bandwidth along the links in the migration paths reduces the bandwidth for all migrations with intersecting links on their migration paths. Opportunity costs of slowing down other migrations using the same network path or links, perhaps missing deadlines, need to be compared to the benefits of finishing the arrived migration in time. The prioritization problem in network revenue management is similar to this issue. Again, network topology knowledge is important for efficient migration path selection and revenue maximization decisions.

5 Summary and outlook

In this paper we introduced network topology aware scheduling models for VM live migrations as well as a scheme for classifying VM workloads. We proposed adequate migration and resource scheduling models for each class, taking explicitly bandwidth requirements and the network topology into account. To our best knowledge, no existing work addresses networking issues as we do. Furthermore, we claimed that for efficient migration scheduling, additional migration control parameters besides the currently available ones are advisable. We proposed scheduling and advanced migration control models and sketched how to put the pieces together to an overall architecture. In co-opertaion with a commercial data center operator we are currently implementing the proposed architecture.

References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. *SIGCOMM Comput. Commun. Rev.*, 38(4):63–74, 2008.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM.

[3] M. Bichler, T. Setzer, and B. Speitkamp. Capacity management for virtualized servers. In *Proc. of Intern. Conf. on Information Systems (ICIS)*, 2006.

[4] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schiöberg. Live wide-area migration of virtual machines including local persistent state. In *VEE '07: Proc. of 3rd intern. conf. on Virtual execution environments*, pages 169–179, New York, NY, USA, 2007. ACM.

[5] C. Clark, K. Fraser, S. H, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proc. of 2nd ACM/USENIX Symp. on Network Systems Design and Implementation*, pages 273–286, 2005.

[6] A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Towards a next generation data center architecture: scalability and commoditization. In *PRESTO '08: Proc. of ACM workshop on Programmable routers for extensible services of tomorrow*, pages 57–62, New York, NY, USA, 2008. ACM.

[7] 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*, pages 620–629, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[8] IEEE. P802.3ba 40gb/s and 100gb/s ethernet task force. http://www.ieee802.org/3/ba/public/index.html, December 2008.

[9] G. Khanna, K. Beaty, G. Kar, and A. Kochut. Application performance management in virtualized server environments. In *10th IEEE/IFIP Netw. Operations and Management Symp.*, pages 373 – 381, Vancouver, BC, Ca, April 2006. IEEE Computer Society.

[10] Y.-A. Kim. Data migration to minimize the total completion time. *J. Algorithms*, 55(1):42–57, 2005.

[11] C. Lu, G. A. Alvarez, and J. Wilkes. Aqueduct: Online data migration with performance guarantees. In *FAST '02: Proc. of 1st USENIX Conf. on File and Storage Techn.*, page 21, Berkeley, CA, USA, 2002. USENIX Association.

[12] M. Nelson, B.-H. Lim, and G. Hutchins. Fast transparent migration for virtual machines. In *ATEC '05: Proc. of the annual conf. on USENIX*, pages 25–25, Berkeley, CA, USA, 2005. USENIX Association.

[13] M. L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, Heidelberg, third edition, 2008.

[14] B. N. Technologies. Nec 10gb intelligent l3 switch, 2008.

[15] VMware, Inc. Vmware esxi. http://www.vmware.com/products/esxi/, 2008.

[16] R. Wolski. Dynamically forecasting network performance using the network weather service. *Journal of Cluster Computing*, 1(119-132), 1998.

[17] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and gray-box strategies for virtual machine migration. In *4th USENIX Symp. on Networked Systems Design and Impl.*, pages 229 – 242, 2007.