Orchestrating Virtual Machine Migrations in Telecom Clouds

Joaquim Barrera, Marc Ruiz, and Luis Velasco*
Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
**e-mail:lvelasco@ac.upc.edu*

Abstract: A throughput model is experimentally assessed and then used to compare non-orchestrated against orchestration when multiple VMs are migrated among datacenters in a telecom cloud. Numerical results show migration time reductions as high as 71%.

OCIS codes: (060.4250) Networks; (060.4256) Networks, network optimization

1. Introduction

Network operators are creating their own cloud infrastructures by deploying datacenters (DC) and interconnecting them through their transport networks; in this paper we call those infrastructures as *telecom cloud*. The kind of applications that can benefit from telecom cloud include, in addition to those currently running in traditional cloud infrastructures, those related to virtualization of network services (NFV) and service chaining [1]. That kind of applications are distributed in nature and can be seen as a set of virtual machines (VMs) running in servers in geographically distant DCs. The application controller places application components running in VMs in specific DCs and, looking for improving the perceived quality of experience (QoE), decides the amount of resources needed for each VM in response to spikes in demand, etc. [2].

The transition from the current state to the new one generally consists in migrating VMs from their current server to the destination one. In that regard, the live-migration technique allows migrating VMs from one server to another without stopping them; VM memory pages are iteratively transferred to the final server until the VM is suspended at the current server to transfer the remaining state to the destination server [3]. When both servers are in the same DC, the live-migration process can be done in short times and thus downtimes are virtually zero; on the contrary, the downtime directly depends on the throughput of the path connecting the servers [4] as a result not only of the delay, but also because some of the already migrated pages are modified (known as dirty pages). In the case that the migration process involves two different DCs, persistent storage is rarely shared and so, application storage needs to be migrated before the memory.

Authors in [5] showed that managing the throughput of the memory migration path can reduce migration time. However, when several migrations need to be performed simultaneously, the throughput of each migration path decreases as a consequence of capacity sharing in those links supporting several migrations. In fact, the effective throughput of each migration path depends on the number of simultaneous migrations sharing each network link, as shown in [6]. In this paper, we first present a complete throughput model that includes TCP control and hard disk (HD) access. Next, we propose a scheduling algorithm, named as *network-aware SchEduLing For vIrtual machinE migratio*n (SELFIE), aiming at reducing individual migration times.

2. Migration paths and throughput modeling

Migration paths entail different parts of the servers, local area and wide area network links and switching elements, such as Ethernet switches. Our focus is in end-to-end paths, so we model DC architecture as a single Ethernet switch assuming that inter-DC connectivity is abundant. Since complete migrations involve two parts, disk storage and memory, different components are also involved. In the intra-server graph, we consider the HD (H) and the memory (M) connected to the CPU (C). The CPU is connected to the network interface card (N) through a PCI express (PCIe) interface. Each server is connected to the Ethernet switch representing the DC,

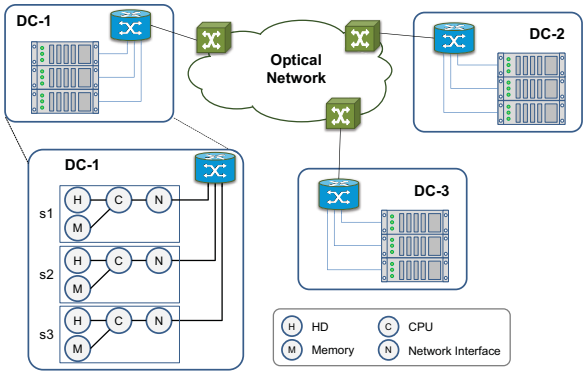

Fig. 1. DC and server modeling as a graph.

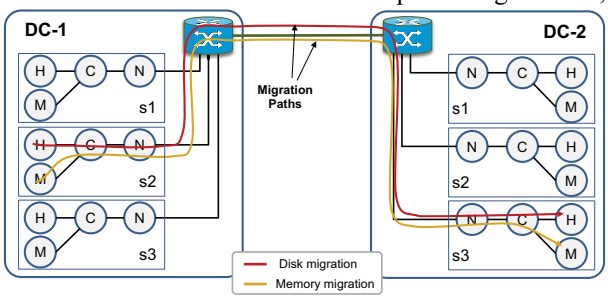

Fig. 2. Example of migration paths.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 317999 IDEALIST project.

and Ethernet switches are connected to switches in other DCs through an optical network (see Fig. 1). Two network links are then considered, intra-DC links connecting each server to the Ethernet switch and inter-DC links connecting Ethernet switches through the optical network. Fig. 2 shows an example of disk storage and memory migration paths between server 2 in DC-1 and server 3 in DC-2 in the considered graph model. Each link in the graph in Fig. 2 has a different throughput, e.g. the throughput of the link H-C is limited by the HD data rate, whereas link M-C depends on the type of memory, e.g. DDR3.

In a simultaneous VM migration, link throughput is shared among all active migrations whose path traverses that link at a given time *t*. Eq. (1) models how link throughput is shared among migrations, where binary variable x_{iet} is 1 only if migration *i* uses link *e* in time *t*, and Th_e is the total throughput. In addition, link throughput depends on the number of active migrations *n* using that link, represented by function $f_e(n)$. The throughput of any given active migration is then the minimum throughput of every link in the path (see eq. (2)).

$$th_e(t) = \frac{Th_e}{\sum_i x_{iet}} \cdot \left(1 - f_e\left(\sum_i x_{iet}\right)\right) \quad (1) \qquad th_i(t) = \min_{e \in p(i)} \{th_e(t)\} \quad (2)$$

In view of the throughput model described by the above equations, we realize that in a non-orchestrated scenario where all migrations are performed simultaneously in parallel, individual migration times will be really long as a result of link throughput sharing. This could derive into unacceptable downtimes when memory dirty rate increases. To mitigate this behavior, in the next section we propose scheduling migrations to reduce both, migration times and downtime.

3. The SELFIE problem

Instead of performing migrations in parallel (we call this as non-orchestrated, NOOR), SELFIE schedules migrations. An example of such orchestration is shown in Fig. 3, where each migration is performed to maximize the allocated throughput, thus reducing VM migration time and consequently, downtime. The SELFIE problem statement is as follows:

Given: *i*) a set of VMs to migrate. Data for each VM to be migrated includes the source and destination servers, disk and memory sizes, memory dirty rate; *ii*) a set of servers where a VM can be placed; *iii*) a set of available inter-DC connections with its throughput; *iv*) a maximum completion time to perform all the migrations.

Output: a migration schedule for every VM, specifying its starting time, completion time, and the throughput allocated at each time interval.

Objective: minimize the average migration time.

The SELFIE problem was formulated using a mixed integer linear programming model. Since its exact solution become impractical when real-sized scenarios are considered, we developed a heuristic algorithm that provides a much better trade-off between optimality and complexity. The next section presents the results from solving SELFIE with the heuristic algorithm.

4. Illustrative numerical results

In this section we first experimentally validate the throughput model proposed in section 2, and then we use the model to numerically compare SELFIE against the non-orchestrated migration.

The model was experimentally validated in a test-bed consisting of two Intel i7 -based servers (8 cores) with 16GB RAM and 1TB mechanical SATA3 HD, connected to a Cisco Catalyst 3750 Ethernet switch using 1Gb/s Ethernet links. Throughput between the servers was controlled by creating VLANs in the switch. We carried out experiments to find the specific f_e functions for Ethernet and H-C links in our test-bed and concluded with eqs. (3) and (4). Note that, as in [6], eq. (3) follows an exponential function but with different coefficients.

$$f_{Ethernet}(n) = 0.20 \cdot e^{-0.1 \cdot (n-1)} \quad (3) \qquad f_{H-C}(n) = 0.55 - 0.025 \cdot n \quad (4)$$

Once f_e functions have been experimentally obtained, the whole model (defined by eqs. (1)-(4)) needs to be validated. To that end, we generated sets of parallel VM migrations with 700MB RAM and 1GB disk and obtained experimental results to compare against model values. The Pearson's correlation coefficient (R^2) between model and experimental values is as high as 99.75%, while the relative average error is 4.2%, which indicates the good accuracy of the model. Fig. 4 plots model against experimental results. Fig. 4a plots total,

Fig. 3. NOOR vs. SELFIE.

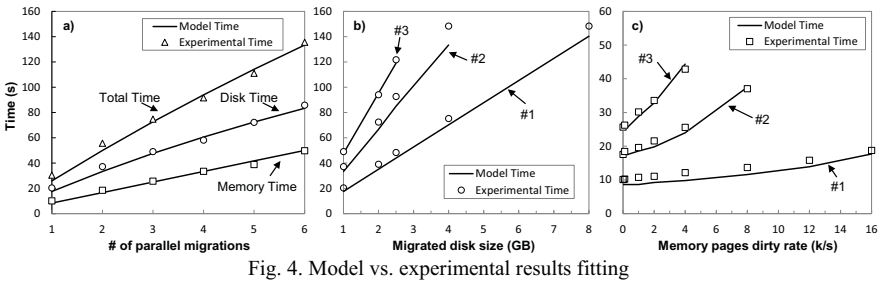

Fig. 4. Model vs. experimental results fitting

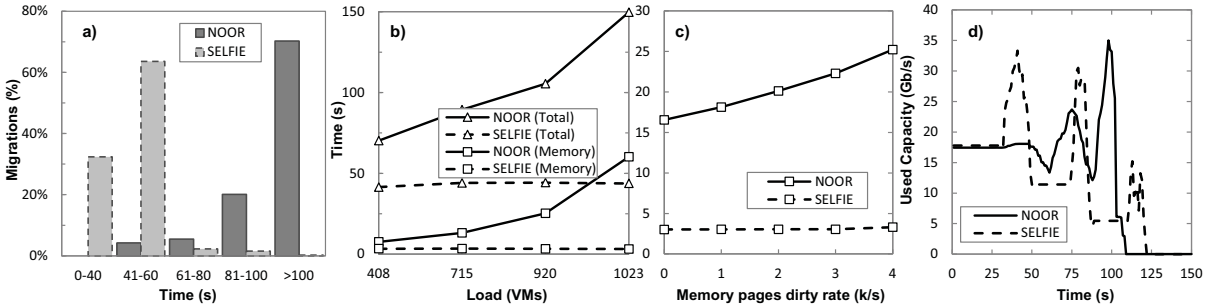

Fig. 5. NOOR vs. SELFIE. a) Migration time histogram. b) Average migration time. c) Average memory migration time vs. dirty rate. d) Used capacity in inter-DC links.

disk, and memory migration times against the number of VMs being migrated in parallel. As shown, the accuracy of the model is almost total since it returns virtually the same results as those obtained experimentally. Fig. 4b focuses on disk migration times for several disk sizes. The model returns the same results as in the experiments when 3 VMs are migrated in parallel, whereas slightly under fits experimental results for 1 and 2 parallel migrations; notwithstanding the model perfectly follows the experimental results slopes for all three cases. Fig. 4c concentrates on memory migration times for several dirty rates and for several parallel migrations. In this case, the model predicts also the experimental curve. It is worth noting that the model predicts whether a migration cannot be complete; it happens when the dirty rate is higher than the throughput of the migration path.

Once the model has been validated, we use it to compare the performance obtained with the non-orchestrated migration and with SELFIE. For the performance comparison, we assumed a scenario with 10 DCs connected through 40 Gb/s optical connections. Each DC follows the architecture depicted in Fig. 1 and was equipped with 128x Dell PowerEdge R715 Rack Server, with 32 cores and 512 GB of DDR3 RAM, x8 line PCIe Gen2, mechanical SATA3 HD, and a 4x1Gb/s network interface card. Problem instances were generated for a case of VM load distribution, where VMs running in few DCs need to be placed in most of DCs. VM sizes were randomly generated in the range [1800-2200] MB for the disk and [500-1000] MB for the memory. The initial placement of VMs followed an intra-DC balanced strategy, where all the servers were equally loaded.

Histogram in Fig. 5a compares NOOR and SELFIE migration time distributions when 920 VMs are migrated. Under the NOOR approach, almost every migration lasted more than 80s, being this time higher than 100s for 70% of migrations. In contrast, SELFIE allows reducing the average migration time since around 90% of migrations lasted less than 60s; in fact, this reduction is as remarkable as 58%. Fig. 5b confirms total and memory migration time reduction for 4 incremental load scenarios, where migration time reduction is as high as 71% under the most stringent scenario. In addition, migration times obtained by using SELFIE show a constant behavior as a result of the way that migrations are performed. This is in contrast to the increasing behavior shown when no orchestration is performed. The constant behavior obtaining using SELFIE ensures that the memory is transferred in the shortest time, thus minimizing the effects of memory dirtying, as shown in Fig. 5c. Note that memory migration times using NOOR show a clear increasing trend. All the above together facilitates noticeably estimating VM downtime when SELFIE is used.

Finally, Fig. 5d illustrates the use of capacity as a function of the time in an inter-DC link (recall that 40Gb/s inter-DC links were considered). Although NOOR and SELFIE show different capacity usage profiles, none of them saturate the link during the migration process. Therefore inter-DC links are not migration bottlenecks (H-C links are). This analysis helps dimensioning both, intra- and inter- DC link capacity for migration purposes avoiding capacity overprovisioning. Furthermore, the ability to predict future network link capacity usage opens the opportunity to use bandwidth-on-demand mechanisms, especially in intra-DC (e.g. by using a SDN controller), thus saving capacity to be used for other applications; this would be especially useful combined with SELFIE, where lower capacity is needed for disk migrations compared to memory migrations.

5. Conclusions

A model to predict the throughput of every VM migration when a set of them are performed in parallel has been proposed and experimentally validated. Applying the throughput model, we observed long migration times when those migrations are not orchestrated. In view of that, a migration scheduling was proposed, named as SELFIE. Exhaustive numerical results showed a remarkable reduction, as high as 71%, in the migration time compared to the non-orchestrated migration. From the results, bandwidth-on-demand mechanisms could potentially be applied to efficiently manage the intra-DC network.

References

[1] ETSI GS NFV 001, “Network Functions Virtualization (NFV): Use Cases”, October 2013.
[2] L. Velasco et al., “Elastic Operations in Federated Datacenters for Performance and Cost Optimization,” Comp. Comm., 2014.
[3] M. Mishra et al., “Dynamic resource management using virtual machine migrations,” IEEE Comm. Mag., vol. 50, pp. 34-40, 2012.
[4] E. Harney et al., “The Efficacy of Live Virtual Machine Migrations Over the Internet,” in Proc. VTDC, 2007
[5] U. Mandal et al., “Heterogeneous bw Provisioning for VM Migration over SDN-Enabled Optical Networks,” in Proc. OFC 2014.
[6] H. Chen et al., “Network-Aware Coordination of VM Migrations in Enterprise Data Centers and Clouds,” in Proc. IM 2013.