Planning Live-Migrations to Prepare Servers for Maintenance

Vincent Kherbache, Eric Madelaine, Fabien Hermenier

Planning live-migrations to prepare servers for maintenance

Vincent Kherbache[1], Eric Madelaine[1], and Fabien Hermenier[2]

[1] INRIA Sophia Antipolis
vincent.kherbache@inria.fr, eric.madelaine@inria.fr
[2] University Nice Sophia Antipolis, CNRS, I3S, UMR 7271, France
fabien.hermenier@unice.fr

Abstract. In a virtualized data center, server maintenance is a common but still critical operation. A prerequisite is indeed to relocate elsewhere the Virtual Machines (VMs) running on the production servers to prepare them for the maintenance. When the maintenance focuses several servers, this may lead to a costly relocation of several VMs so the migration plan must be chose wisely. This however implies to master numerous human, technical, and economical aspects that play a role in the design of a quality migration plan.
In this paper, we study migration plans that can be decided by an operator to prepare for an hardware upgrade or a server refresh on multiple servers. We exhibit performance bottleneck and pitfalls that reduce the plan efficiency. We then discuss and validate possible improvements deduced from the knowledge of the environment peculiarities.

1 Introduction

In data centres, virtualization has become a cornerstone. On one side, it raised the hosting capabilities thanks to performance isolation [2] and consolidation techniques [19,12]. On the other side, live migration [5] permitted the operators to perform server maintenance more easily. Indeed, maintenance operations such as server updating, hardware or software upgrade are critical tasks to perform on production servers. It is then recommended to operate on idle or offline servers to prevent any failure or mis-configuration to alter client virtual machines (VMs). Thanks to live migration, it is now possible to prepare the servers by migrating their VMs elsewhere in prior, with a negligible downtime for the VMs.

Maintenance tasks can occur at the level of a single server as well at the scale of an entire blade-center or rack. With the ever increasing number of servers and VMs per server in a data center, planning efficiently numerous migrations over multiple servers becomes problematic [17]. Indeed the notion of efficiency has many facets: an operator may expect short completion times, small migration durations or low energy usage for example. However, many technical, environmental or even human aspects dictate these optimisation criteria and today, all these parameters but also their interactions must be mastered to design migration plans of quality.

In this paper we analyze different realistic migration plans to exhibit common pitfalls and discuss some levers to improve their quality. Our results are derived from experiments on a real testbed involving up to 45 servers connected through a hierarchical network. In both scenarios we compare two migration strategies that consist to execute all the tasks in parallel or sequentially. We analyze the pros and cons of both approaches with regards to performance, energy efficiency and duration optimization criteria. Finally, we discuss and validate possible improvements that consider the infrastructure and the workload properties.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 presents our experimental analysis of migrations plans. Section 4 discusses possible solutions to improve the plan efficiency. Finally, Section 5 presents our conclusions and future research directions.

2 Related Works

Live migrations efficiency: Many efforts have been made to improve the live migration efficiency and many research papers proved that the network speed and the VM's dirty page rate are the main factors affecting the live migrations behavior in pre-copy migration architecture. Based on these findings, Sherif et al. [1] offer good predictions of the duration and the workload service interruptions arising from live migrations. Also, to help administrators at making optimal migrations decisions, Liu et al. [14] define a performance model to predict the energy consumed by a live migration at different transmission rates. Although being linked to our work, these work study the migration of a single VM while we focus on issues related to the concurrent migration of multiple VMs.

Accordingly, several works have been conducted to improve the performance of multiple live migrations. Among the studies that most closely match our work, Kejiang et al. [20] consider the live migration efficiency of multiple VMs with different strategies (sequential / parallel migrations) by investigating resources reservation methods on target servers. Nevertheless their study does not include network management or information about the topology which are the preeminent aspects that we consider in this paper. Sarker et al. [16] propose an algorithm to schedule the migrations of a given set of VMs by minimizing the total migration time and the VMs downtime. The novelty of their approach is to take into account the network topology and the inter-VM data dependencies. In this paper we also focus on the need to reduce individual migration durations and energy consumption. Furthermore, despite all their experiments were performed in a simulated environment, we focus exclusively on a real testbed. Deshpande et al. [7] introduce Live gang migration of VMs to speed up the parallel migration of co-located VMs to the same destination server thanks to memory deduplication. Nevertheless the proposed technique requires a deep modification of the underlying hypervisor and does not address the migration of VMs over a complex network topology. Zheng et al. [22] propose a centralized architecture to coordinate the migration of multi-tiers applications between distant sites inter-connected through a slow network path. The objectives are to ensure the convergence of

all migrations and to minimize the impact of inter-VMs communications on migrations duration. In contrast, in this paper we consider an isolated network dedicated to migrations within the same data-centre, which greatly reduces the impact of inter-VMs communications on the migrations performance.

Maintenance operations in virtualized data centers: The new considerations related to the virtualization for management operations in data centers have been introduced in [17]. The authors explain that the management operations constitute themselves a workload over the applications running in VMs and becomes more and more critical with increasing multi-core architectures. They analyze 5 common management tasks in virtualized data centres, although they do not investigate the blade-center maintenance or server upgrading scenarios which are the main interests of this paper.

In the best of our knowledge, our work is the first study to tackle multiple migrations plans in the context of critical maintenance operations such as replacing a whole blade-center in a real infrastructure, and to propose solutions to automate these operations with the aim to facilitate the work of administrators.

3 Analysis of migrations plans

In this section, we experiment on a testbed the effects of 2 intuitive migration plans in the case of a blade-center maintenance or a server upgrading. In practice, we evaluate the impact of the migration plans on the completion time, the individual migration duration, the instantaneous power and the energy consumption.

3.1 Environment

The experimental testbed is composed of three Bullx B500 blade-centers. Each blade-center consists of 15 servers with 2 Intel quad-core Xeon E5520 2.27 GHz processors and 24 GB RAM each. All servers run Debian Wheezy with a Linux 3.2.0-4 amd64 kernel and the KVM/Qemu hypervisor 1.7.50. The testbed hosts 60 VMs. Every single VM uses 2 VCPUs, 2 GB RAM and runs a Ubuntu 13.10 desktop distribution. Each VCPU is mapped to a dedicated physical core.

Figure 1 depicts a testbed fragment. In a single blade-center, each server is connected to a switch through a Gigabit Ethernet interface. The bandwidth between the blade-centers is however limited to 3 Gb/s by an aggregation of 3 Gigabit links. All the servers are also connected to a 10 Gb/s Infiniband network that share the VM disk images exported by a dedicated NFS server. To only analyze the migration related traffic, only the live migrations operate over the Ethernet network.

The VM workload is generated by the Web server benchmark tool `httperf` [15]. Inside each VM, the benchmark repeatedly retrieves a static Web page from a local Apache Web server. Two workloads, equally distributed between the VMs, retrieve the Web page at a rate of 100 or 200 requests per second.

During experiments, the power consumption of each server is retrieved every second from a remote dedicated server through its management board.

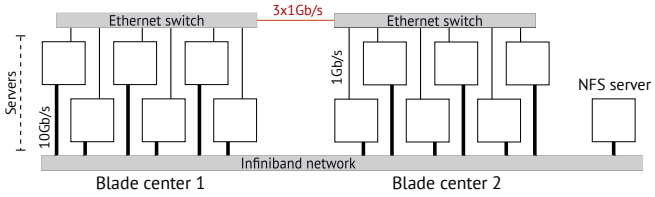


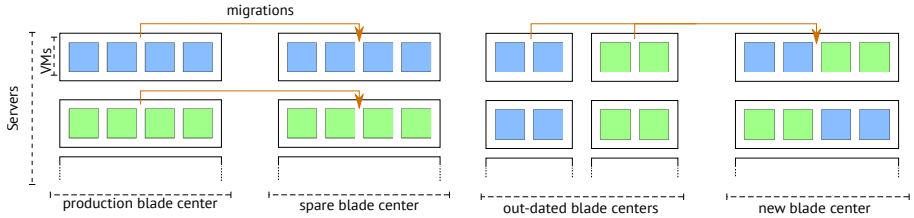
Fig. 1: Testbed design

3.2 Experiments

We consider two maintenance scenarios that reflect common situations:

Scenario 1 - Blade-center maintenance: This scenario simulates the preparation of a maintenance on a whole blade-center that need to be powered down. The 60 VMs are relocated to a spare blade-center having the same hardware specification. The spare servers are initially offline to save power. Each server to put into maintenance hosts 4 VMs. All the VMs of a source server are migrated to a specific destination server (see Figure 2a).

Scenario 2 - Server upgrading: This scenario simulates the replacement of two out-dated blade-centers by a single one that is more powerful. Each deprecated server has 4 cores while each new server has 8. Initially, each deprecated server hosted 2 VMs while each new server will host 4 VMs (see Figure 2b). Initially the servers in the new blade-center are offline. Once the migration terminated, the old blade-center is shut down. To simulate the low performance of the out-dated servers, half the cores are disabled using linux `procfs`.



(a) Scenario 1 - Blade-center maintenance (b) Scenario 2 - Server upgrading

Fig. 2: Experimental scenarios

In both scenarios, we evaluated 2 migration strategies that can be inferred naturally by an operator. The first strategy launches all the migrations sequentially. This has the benefits of being safe and easily trackable. The second strategy launches all the migrations in parallel to reduce the completion time, a common objective of reconfiguration algorithms to increase their reactivity [12,22,19]. Table 1 shows the experimental results.

We observe that with parallel strategies, the average VM migration duration is about 15 times longer, but also less stable, than with the sequential strategies.

	Scenario 1		Scenario 2	
Metrics	Sequential	Parallel	Sequential	Parallel
Time to completion (sec.)	2871	446	3467	384
Mean migration duration (sec.)	12.2	192.9	11.2	158.0
standard deviation	5.41	45.12	4.81	52.97
Server boot time (sec.)	113.1	116.5	114.9	115.0
Server shutdown duration (sec.)	29.5	28.8	32.2	32.1
Energy consumption (kWh)	2098.4	366.4	3317.5	548.1
Max. peak power (kW)	2.70	4.47	4.24	6.05

Table 1: Scenarios comparison

This difference is explained by the network interlink that restricts the throughput between the blade-centers to 3 Gb/s when the maximum rate could be up to 15 Gb/s. In contrast, the interlink bandwidth is under-utilized in sequential strategies as the maximum throughput between the two blade-centers equals 1 Gb/s. In practice, long migration durations are not desirable as they lead to performance issues. Indeed, a migration consumes resources on the involved servers and this additional load reduce the VM performance. Furthermore, the links aggregation that composes the interlink does not balance the traffic fairly. Indeed, the negotiation protocol distributes the traffic with a XOR hash-based on the source and the destination MAC addresses. As a consequence in parallel strategies, multiples migrations can share a single 1 Gb link while others will have a dedicated one. Therefore the main issue of parallel strategies is related to the network overload but also the network topologies. Both must be carefully investigated to use them to their best.

Another limitation of parallel strategies occurs when a software license is needed for each running server [4]. Indeed, parallel strategies bring online 15 additional servers simultaneously, which means that 15 additional licenses must be acquired for a short utilisation period. On the other side, only 1 spare license is required when the migration plan is performed server by server. It might therefore be important to adapt the level of parallelism, so the number of servers simultaneously online, to the number of server licenses [6].

We observe that with sequential strategies, booting a server (respectively shutting down) is about 10 times longer (resp. 3 times) than the average migration duration. As each action is executed sequentially, the time spent to boot and shutdown the servers is not used to migrate VMs. In the scenario 1, 2139 sec. or 74.5 % of the completion time is then wasted waiting for power switching actions (boot and shutdown of 15 servers). It is usually not desirable to have long standing critical operations as the operator in charge must be continuously available to fix potential failures. It is then important to parallelize as much as possible the power-switching actions to reduce the waiting time to a minimum. Likewise, the longest completion times in scenario 2 are essentially due to the time spent to shutdown the 15 additional servers.

We finally observe a higher energy consumption in the scenario 2 due to the higher number of servers. More important, we observe significant power

consumption peaks in the parallel strategies. These peaks occur at the beginning of each experiment during the simultaneous boot of all the destination servers. This situation is problematic when the energy is a scarce or expensive resource. For example, when the energy price market is volatile [18] or when the data center is partially powered by renewable energies [13,9]. In these cases, a solution is to delay some boot actions to more *energy-friendly* periods. Such a delay must however be considered carefully with regards to the priority of the maintenance operation. These results demonstrate the need to control the energy consumption during the maintenance task to be adaptive to external energy constraints. One of the consequences will be to choose the best sequence of power switching actions.

4 Toward Smarter Migration Plans

Experiments exhibited that pure parallel and pure sequential strategies have their own benefits and drawbacks. Pure parallel strategies provide short completion time but long migrations while pure sequential strategies provide the opposite. In practice, the efficiency of each approach is strongly related to the environment and the workload peculiarities. This advocates for a smart composition of both approaches to provide finer migration plans. In this section, we explore hybrid strategies to prepare servers for a blade-center maintenance according to the network and the workload peculiarities and verify their effectiveness.

Metrics	Scenario A1	Scenario A2	Scenario A3
Mean migration duration (sec.)	284.2	63.66	50.62
standard deviation	251.78	33.15	23.17
Time spent to migrate (sec.)	604	213	148
Energy consumption (kWh)	286.27	156.35	132.52

Table 2: Optimisations according to the network interlink peculiarities

The first experiment considers the network interlink in a testbed reduced to 6 servers per blade-center. We chose this smaller and more manageable set of servers to easily analyze the behavior of the 3 links aggregation. In all subsequent experiments, `httperf` is configured at a rate of 200 requests per second for all VMs. Table 2 shows the results. In Scenario A1, all the VMs are migrated in parallel. Similarly to previous experiments, we observe long and unstable migration due to the interlink saturation. Furthermore, some migrations did not complete in live. This happens when the dirty page rate of a VM is greater than the bandwidth available for the migration. In this case, KVM cannot guarantee a VM downtime lesser than 30 ms, the maximum allowed by default. It then suspends the VM after 10 minutes for a possible long period to terminate the migration. We note that this behavior does not occur in the previous scenarios involving a whole blade-center, this is mainly explained by the less intensive workloads on VMs. In Scenario A2, the source servers are freed 3 by 3. We then

observe the migration time is 4 time faster and the completion time is 3 times shorter. This is explained by the interlink that is no longer saturated as each server has in theory a 1Gbit/s bandwidth to migrate its VMs. Dirty-pages are then send faster and the number of rounds to synchronize the memory is reduced. We however reported in Section 3 that the link aggregation protocol is not fair. In Scenario A3, we then decided to probe the interlink topology using `iperf` to choose for each source server, a destination server reachable through a dedicated Gigabit link. This micro-optimisation reduced again the total migration time by 65 sec. and the average migration duration by 13 sec. This experiments reveals that a fine grain optimisation of the level of parallelism between the migration of different servers allows to reduce by up to 4 the time spent to migrate but also by 5 the average migration duration. We also observe that the energy consumption is lower in connection with the reduced completion time.

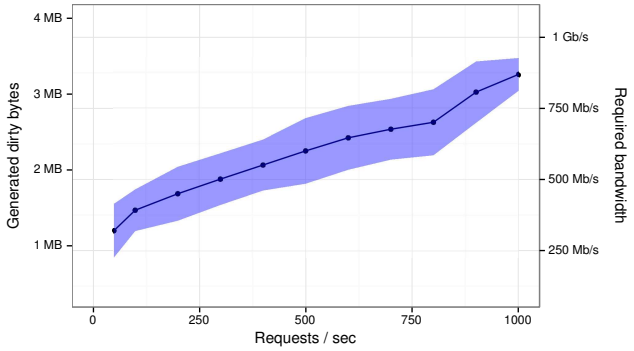


Fig. 3: dirty pages generated by `httperf` in 30 ms (95% confidence interval)

We finally refines the migration plan according to the workload peculiarity. With regard to results in Section 3, we observe the average migration duration is 7 times longer and less stable than when migrations are performed sequentially. This reveals despite the network interlink is not saturated, the parallel migration of 4 VMs on a single Gigabit link still saturates the network. According to the pre-copy algorithm used in Qemu, a bandwidth below a certain threshold causes the re-transmission of the set of dirty pages that are quickly updated, named `Writable Working Set` (WWS) [5]. The minimum bandwidth that guarantee the termination of a migration depends therefore of the WWS size and the memory dirtying rate. Figure 3 shows the number of pages that are made dirty by `httperf` in 30 ms depending on the request rate. This indicates that with a 200 requests per second rate, at most 2 VMs should be migrated simultaneously on a Gigabit link to ensure their termination. We verified this assumption in an experiment that varies the number of VMs migrated simultaneously on a Gigabit link. It consists to migrate the VMs of 3 servers from a server to another.

The servers were selected to ensure an equitable sharing of the 3 Gb/s interlink between them.

Metrics	Scenario B1	Scenario B2	Scenario B3
Mean migration duration (sec.)	49.51	15.41	7.66
standard deviation	34.51	0.47	0.45
Time spent to migrate (sec.)	94	31	31
Energy consumption (kWh)	46.36	33.49	33.28

Table 3: Optimisations according to the workload peculiarities

Table 3 shows the results. In Scenario B1, the 4 VMs on each server are migrated in parallel. In Scenario B2, the VMs are migrated 2 by 2. With regard to Scenario B1, the migration duration is 3 times shorter and stable. This indicates the network is no longer saturated and the bandwidth available for the migration is sufficient to prevent a repetitive copy of the dirty pages. In Scenario B3, the VMs are migrated one at a time. With regard to Scenario B2, the migration duration has been only divided by 2 while the completion time remains unchanged. The last two scenarios ensure then a fully effective migration management by dealing with workloads and network specificities. We were able to reduce by up to 7 the average migration duration and by up to 3 the time spent to migrate. However, it may be judicious to prefer Scenario B3 over Scenario B2 depending on the VMs peculiarities. For example, when a group of VMs communicates extensively, it is preferable to migrate the VMs in parallel and synchronize the migration terminations to restrict the amount of data exchanged over a high-latency interlink [22]. In contrast, when the VMs are independent, it is wise to migrate the VMs one by one to reduce the average migration duration so the impact on the workload.

5 Conclusion and Future Work

Server maintenance is a common but still critical operation that must be prepared by operators. It requires to plan the migration of numerous VMs but also the management of the server state. With the ever increasing complexity of the datacenter infrastructure, it becomes difficult to define plans that are fast, reliable or simply fitting the environment peculiarities.

In this paper, we experimented migration plans involving up to 45 servers. This exhibited performance bottlenecks but also evaluation metrics to qualify the quality of a migration plan. We then show how the knowledge of the environment peculiarities can improve the migration plan quality. In practice, we adapted the number of migrations to perform in parallel between the servers, but also inside each server. These decisions were applied manually from the knowledge of the network topologies, and the workload particularities.

As future work, we then want to automatize the creation of efficient plans. We first need to model the aspects that qualify a migration plan. Based on the

experiments, we conclude our model must consider the workload characteristics such as the dirty page rate and the estimated migration durations, the network topology but also external and possible evolving side constraints such as a possible power budget, a completion deadline or a server licensing policy. We already patched Qemu to retrieve the VM dirty page rate but we also planned to use an approach similar to Pacer [21] to predict the duration of a live migration. With regards to the network, it is possible to extract the network topology using standard monitoring tools. In addition, dynamic aspects such as the practical decomposition of the traffic made by an aggregation protocol can be observed from benchmarks. We plan to implement this model over the VM manager BtrPlace [10,11]. BtrPlace is an extensible VM manager that can be customized to augment its inferring capabilities. It provides a composable VM placement algorithm that has already been used to address energy-efficiency [8], or scheduling concerns such as the continuous respect of server licensing policies [6]. The use of BtrPlace might also be beneficial to support side constraints that have to be expressed by the operators. It already provides a support for configuration scripts to state easily constraints over servers and VMs. Furthermore, the implementation of the constraints is usually short.

We also want to investigate on another common maintenance operation that is the usage of anti-virus over VM disk images, a very storage intensive operation that must be planned carefully to maintain the performance of the storage layer. More generally, we think that while advanced algorithms have been proposed to optimize the datacenter usage, there is a large pace for innovation to assist operators at doing their job. Typically, how to automatically improve the preparation of maintenance operations from high-level expectations while hiding the complex technical peculiarities that are today required to be mastered.

Acknowledgments

This work has been carried out within the European Project DC4Cities (FP7-ICT-2013.6.2). Experiments presented in this paper were carried out using the Grid'5000 experimental testbed [3][3], being developed by INRIA with support from CNRS, RENATER and several universities as well as other funding bodies.

References

1. Akoush, S., Sohan, R., Rice, A., Moore, A.W., Hopper, A.: Predicting the performance of virtual machine migration. In: MASCOTS. IEEE (2010)
2. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: 19th SOSP (2003)
3. Bolze, R., Cappello, F., Caron, E., Daydé, M., Desprez, F., Jeannot, E., Jégou, Y., Lanteri, S., Leduc, J., Melab, N., Mornet, G., Namyst, R., Primet, P., Quetier, B., Richard, O., Talbi, E.G., Touche, I.: Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed. Int. Journal of High Performance Computing Applications 20(4) (Nov 2006)

[3] `https://www.grid5000.fr`

4. Citrix store. http://store.citrix.com
5. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of the 2nd NSDI. USENIX Association (2005)
6. Dang, H.T., Hermenier, F.: Higher SLA Satisfaction in Datacenters with Continuous VM Placement Constraints. In: Proceedings of the 9th Workshop on Hot Topics in Dependable Systems. HotDep '13, ACM, New York, NY, USA (2013)
7. Deshpande, U., Wang, X., Gopalan, K.: Live gang migration of virtual machines. In: Maccabe, A.B., Thain, D. (eds.) HPDC. ACM (2011)
8. Dupont, C., Schulze, T., Giuliani, G., Somov, A., Hermenier, F.: An energy aware framework for virtual machine placement in cloud federated data centres. In: Proceedings of the 3rd International Conference E-energy. ACM, NY, USA (2012)
9. Goiri, I.n., Katsak, W., Le, K., Nguyen, T.D., Bianchini, R.: Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy. In: Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems. ASPLOS '13, ACM, NY, USA (2013)
10. Hermenier, F., Demassey, S., Lorca, X.: Bin repacking scheduling in virtualized datacenters. Principles and Practice of Constraint Programming (2011)
11. Hermenier, F., Lawall, J., Muller, G.: BtrPlace: A Flexible Consolidation Manager for Highly Available Applications. IEEE Transactions on Dependable and Secure Computing 10(5) (2013)
12. Hermenier, F., Lorca, X., Menaud, J.M., Muller, G., Lawall, J.: Entropy: a Consolidation Manager for Clusters. In: Proceedings of the ACM SIGPLAN/SIGOPS Intl. Conference on Virtual Execution Environments. ACM, NY, USA (2009)
13. Li, C., Qouneh, A., Li, T.: iSwitch: Coordinating and optimizing renewable energy powered server clusters. In: 39th Annual International Symposium on Computer Architecture (ISCA) (June 2012)
14. Liu, H., Jin, H., Xu, C.Z., Liao, X.: Performance and energy modeling for live migration of virtual machines. Cluster Computing 16(2) (2013)
15. Mosberger, D., Jin, T.: httperf - a Tool for Measuring Web Server Performance. SIGMETRICS Performance Evaluation Review 26(3) (1998)
16. Sarker, T., Tang, M.: Performance-driven live migration of multiple virtual machines in datacenters. In: IEEE International Conference on Granular Computing (2013)
17. Soundararajan, V., Anderson, J.M.: The Impact of Management Operations on the Virtualized Datacenter. SIGARCH Comput. Archit. News 38(3) (Jun 2010)
18. U.S. Energy Information Administration: Wholesale Electricity and Natural Gas Market Data . http://www.eia.gov/electricity/wholesale/ (May 2014)
19. Verma, A., Ahuja, P., Neogi, A.: pmapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. Middleware '08, Springer-Verlag New York, Inc., New York, NY, USA (2008)
20. Ye, K., Jiang, X., Huang, D., Chen, J., Wang, B.: Live migration of multiple virtual machines with resource reservation in cloud computing environments. In: Liu, L., Parashar, M. (eds.) IEEE CLOUD. IEEE (2011)
21. Zheng, J., Ng, Sripanidkulchai, K., Liu, Z.: Pacer: A Progress Management System for Live Virtual Machine Migration in Cloud Computing. IEEE Transactions on Network and Service Management 10(4) (Dec 2013)
22. Zheng, J., Ng, T.S.E., Sripanidkulchai, K., Liu, Z.: COMMA: Coordinating the Migration of Multi-tier Applications. In: Proceedings of the 10th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. VEE '14, ACM, New York, NY, USA (2014)