Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation

William Voorsluys[1], James Broberg[1], Srikumar Venugopal[2], and Rajkumar Buyya[1]

[1] Grid Computing and Distributed Systems Laboratory
Department of Computer Science and Software Engineering, The University of Melbourne, Australia
{williamv,brobergj,raj}@csse.unimelb.edu.au
[2] School of Computer Science and Engineering, The University of New South Wales, Australia
srikumarv@cse.unsw.edu.au

Abstract. Virtualization has become commonplace in modern data centers, often referred as "computing clouds". The capability of virtual machine live migration brings benefits such as improved performance, manageability and fault tolerance, while allowing workload movement with a short service downtime. However, service levels of applications are likely to be negatively affected during a live migration. For this reason, a better understanding of its effects on system performance is desirable. In this paper, we evaluate the effects of live migration of virtual machines on the performance of applications running inside Xen VMs. Results show that, in most cases, migration overhead is acceptable but cannot be disregarded, especially in systems where availability and responsiveness are governed by strict Service Level Agreements. Despite that, there is a high potential for live migration applicability in data centers serving modern Internet applications. Our results are based on a workload covering the domain of multi-tier Web 2.0 applications.

Key words: Virtual machines, performance evaluation, migration, Xen

1 Introduction

Virtual machine (VM) technology has recently emerged as an essential building-block for data centers and cluster systems, mainly due to its capabilities of isolating, consolidating and migrating workload [1]. Altogether, these features allow a data center to serve multiple users in a secure, flexible and efficient way. Consequently, these virtualized infrastructures are consider a key component to drive the emerging Cloud Computing paradigm [2].

Migration of virtual machines seeks to improve manageability, performance and fault tolerance of systems. More specifically, the reasons that justify VM migration in a production system include: the need to balance system load, which can be accomplished by migrating VMs out of overloaded/overheated

servers; and the need of selectively bringing servers down for maintenance after migrating their workload to other servers.

The ability to migrate an entire operating system overcomes most difficulties that traditionally have made process-level migration a complex operation [3, 4]. The applications themselves and their corresponding processes do not need to be aware that a migration is occurring. Popular hypervisors, such as Xen and VMWare, allow migrating an OS as it continues to run. Such procedure is termed as "live" or "hot" migration, as opposed to "pure stop-and-copy" or "cold" migration, which involves halting the VM, copying all its memory pages to the destination host and then restarting the new VM. The main advantage of live migration is the possibility to migrate an OS with near-zero downtime, an important feature when live services are being served [3].

1.1 Background

On Xen, as described by Clark et al. [3], live migrating a VM basically consists of transferring its memory image from a source server to a destination server. To live migrate a VM, the hypervisor pre-copies memory pages of the VM to the destination without interrupting the OS or any of its applications. The page copying process is repeated in multiple rounds on which dirty pages are continuously transferred. Normally, there is a set of pages that is modified so often that the VM must be stopped for a period of time, until this set is fully transferred to the destination. Subsequently, the VM can be resumed in the new server.

It has been observed that live migration of VMs allows workload movement with near zero application downtime. Nevertheless, the performance of a running application is likely to be negatively affected during the migration process due to the overhead caused by successive iterations of memory pre-copying [3]. For the duration of the pre-copying process extra CPU cycles are consumed on both source and destination servers. An extra amount of network bandwidth is consumed as well, potentially affecting the responsiveness of Internet applications. In addition, as the VM resumes after migration, a slowdown is expected due to cache warm-up at the destination [5].

Moreover, downtime and application performance are likely to be affected in different ways for different applications due to varying memory usages and access patterns. Previous studies have found that actual downtime may vary considerably between applications, ranging from as low as 60 ms when migrating a Quake game server [3] to up to 3 seconds in case of particular HPC benchmarks [5]. Regarding the overhead due to migration activity, earlier studies have shown that experienced slowdown ranged between 1% and 8% of wall-clock time for a particular set of HPC benchmarks [5].

In other scenarios using Xen, a 12% to 20% slowdown on the transmission rate of an Apache Web server running a VM with 800MB of memory and serving static content was reported [3]. In the case of a complex Web workload (SPECWeb99) the system under test could maintain the conformity to the benchmark metrics [3]. In all cases, it has been concluded that, for the particular

set of applications considered, the bad effects of migration were acceptable or negligible in contrast to its potential benefits to system fault tolerance [5].

1.2 Our Contribution

We have identified that a case study taking into consideration live migration effects in the performance of modern Internet applications, such as multi-tier Web 2.0 applications, is lacking in the current literature. However, such a study would aid researchers and practitioners currently evaluating the deployment of this class of application in clouds. Our contribution is a case study that quantifies the effect of VM live migrations in the performance of one example, yet representative, of a modern Internet application. Our study will be potentially useful to environments where metrics, such as service availability and responsiveness, are driven by Service Level Agreements (SLAs). In such systems service providers and consumers agree upon a minimum service level and non-compliance to such agreement may incur in penalties to providers [6]. More importantly, an SLA directly reflects how end-users perceive the quality of service being delivered.

The rest of this paper is organized as follows: Section 2 positions our study among related work; Section 3 describes why modern Internet applications are different than traditional workloads; Section 4 describes our objectives, experimental testbed, workload and metrics; Section 5 presents the results of our performance evaluation; finally, we conclude the paper in Section 6.

2 Related Work

The advent of innovative technologies, such as multicore [7], paravirtualization [1], hardware-assisted virtualization [8] and live migration [3], have contributed to an increasing adoption of virtualization on server systems. At the same time, being able to quantify the pros and cons of adopting virtualization in face of such advancements is a challenging task. The impact of virtualization in a variety of scenarios has been the focus of considerable attention. A number of studies have presented individual and side by side measurements of VM runtime overhead imposed by hypervisors on a variety of workloads [1, 9].

Apparao et al. [10] present a study on the impact of consolidating several applications on a single server running Xen. As workload the authors employed the vConsolidate benchmark [11] defined by Intel, which consists of a Web server VM, a database server VM, a Java server VM and mail server VM. An idle VM is also added to comply with real world scenarios, on which servers are hardly fully utilized.

The studies presented by Zhao & Figueiredo [12] and Clark et al. [3] specifically deal with VM migration. The former analyzes performance degradation when migrating CPU and memory intensive workloads as well as migrating multiple VMs at the same time; however such study employs a pure stop-and-copy migration approach rather than live migration. The later introduces Xen live migration and quantifies its effects on a set of four applications common to

hosting environments, primarily focusing on quantifying downtime and total migration time and demonstrating the viability of live migration. However, these works have not evaluated the effect of migration in the performance of modern Internet workloads, such as multi-tier and social network oriented applications.

A few studies propose and evaluate the efficacy of migrating VMs across long distances, such as over the Internet. For instance, Travostino et al. [13] have demonstrated the effectiveness of VM live migration over an WAN connected by dedicated 1Gbps links; application downtime has been quantified at 5-10 times greater than that experienced on an intra-LAN set-up, despite a 1000 times higher RTT. Besides its feasibility, the concept of WAN live migration is still to be implemented in commercial hypervisors, which demands all involved machines to be in the same subnet and share storage. Our work focuses only on migrating VMs within a data center or cluster.

The Cloudstone benchmark [14] aims at computing the monetary cost, in dollars/user/month, for hosting Web 2.0 applications in cloud computing platforms such as Amazon EC2. From this work we borrow the idea of using Olio [15] and Faban [16] to compose our target workload for Web 2.0 applications. However, Cloudstone does not define a procedure to evaluate the cost of virtual machine migration and, to the best of our knowledge, no previous work has considered using this type of workload in migration experiments.

3 Characteristics of Modern Internet applications

The domain of applications that can potentially take advantage of the Infrastructure as a Service paradigm is broad. For instance, Amazon [17] reports several case studies that leverage their EC2 platform, including video processing, genetic simulation and Web applications. In particular, such platforms are especially useful for multi-tier Web applications, generally including a Web server (e.g. Apache), an application server/dynamic content generation (e.g. PHP, Java EE), and a backend database (e.g. MySQL, Oracle). Virtual machine technology adds extra flexibility to scaling of Web applications, by allowing dynamic provisioning and replication VMs to host additional instances for one the application tiers.

Social networking websites are perhaps the most notable example of highly dynamic and interactive Web 2.0 applications which gained popularity over the past few years. Their increasing popularity has spurred demand for a highly scalable and flexible solution for hosting applications. Many larger sites are growing at 100% a year, and smaller sites are expanding at an even more rapid pace, doubling every few months [18]. These web applications present additional features that make them different from traditional static workloads [14]. For instance, their social networking features make each users' actions affect many other users, which makes static load partitioning unsuitable as a scaling strategy. In addition, by means of blogs, photostreams and tagging, users now publish content to one another rather than just consuming static content.

Altogether, these characteristics present a new type of workload with particular server/client communication patterns, write patterns and server load. However, most available performance studies use extremely simple static file retrieval tests to evaluate Web servers, often leading to erroneous conclusions [18]. In this work we have this trend into account during the workload selection process, resulting in the selection of Olio as a realistic workload.

4 Evaluation of Live Migration Cost

This study aims at achieving a better understanding of live migration effects on modern Internet applications. We have designed benchmarking experiments to evaluate the effect of live migration on a realistic Web 2.0 application hosted on networked virtual machines.

4.1 Testbed Specifications

Our testbed is a cluster composed of 6 servers (1 head-node and 5 VM hosts). Each node is equipped with Intel Xeon E5410 (a 2.33 GHz Quad-core processor with 2x6MB L2 cache and Intel VT technology), 4 GB of memory and a 7200 rpm hard drive. The servers are connected through a Gigabit Ethernet switch.

The cluster head-node runs Ubuntu Server 7.10 with no hypervisor. All other nodes (VM hosts) run Citrix XenServer Enterprise Edition 5.0.0. Our choice for a commercial hypervisor is based on the assurance of an enterprise class software in accordance with the needs of target users, i.e. enterprise data centers and public application hosting environments.

All VMs run 64-bit Ubuntu Linux 8.04 Server Edition, paravirtualized kernel version 2.6.24-23. The installed web server is Apache 2.2.8 running in prefork mode. PHP version is 5.2.4-2. MySQL, with Innodb engine, is version 5.1.32.

4.2 Workload

We use Olio [15] as a Web 2.0 application, combined with the Faban load generator [16] to represent an application and workload set. Olio is a Web 2.0 toolkit that helps developers evaluate the suitability, functionality and performance of various Web technologies, devised by Sun Microsystems from its understanding of the challenges faced by Web 2.0 customers [18]. It has been successfully deployed and evaluated in a reasonably sized high-end server infrastructure [18], as well as in rented resources from Amazon EC2 [14].

The Olio Web application represents a social-events website that allows users to perform actions such as loading the homepage, logging into the system, creating new events, attending events and searching for events by date or tag. It currently provides implementations using three technologies: PHP, Ruby on Rails and J2EE. For our experiments, we have chosen to use Olio's PHP implementation, thus employing the popular LAMP stack (Linux Apache MySQL PHP).

Faban is an open-source Markov-chain load generator used to drive load against Olio; it is composed by a master program which spawns one or more load drivers, i.e. multi-threaded processes that simulate actual users. The master presents a Web interface through which it is possible to submit customized benchmark runs and monitor their results. This Olio/Faban combination was originally proposed as part of the Cloudstone benchmark [14].

The load level driven against the application may be varied by changing the number of concurrent users to be served by the application. Total time for each run is configured by adjusting three different durations, namely ramp-up, steady state and ramp-down. Resulting metrics reported by Faban only take into account the steady state period.

The main metric considered in our experiments is a Service Level Agreement defined in Cloudstone. The SLA defines minimum response times for all relevant user actions. Thus, at any 5-minute window, if a certain percentile of response times exceeds the maximum, an SLA violation is recorded. The 90th and 99th percentiles are considered in this study, representing a more relaxed and a stricter SLA, respectively. Table 1 lists the details of the SLA.

Table 1. Cloudstone's SLA: The 90th/99th percentile of response times measured in any 5-minute window during steady state should not excess the following values (in seconds):

User action	SLA	User action	SLA
Home page loading	1	User login	1
Event tag search	2	Event detail	2
Person detail	2	Add person	3
Add event	4		

4.3 Benchmarking Architecture

The architecture of our benchmarking setup is depicted in Figure 1. Based on the observation that MySQL tends to be CPU-bound when serving the Olio database, whereas Apache/PHP tends to be memory-bound [14], we have designed our system under test (SUT) by splitting the workload into two networked VMs, hosted in different cluster nodes, in order to better partition the available physical resources.

All nodes share an NFS (Network File System) mounted storage device, which resides in the head-node and stores VM images and virtual disks. In particular, a local virtual disk is hosted in the server that hosts MySQL.

The load is driven from the head-node, where the multi-threaded workload drivers run, along with Faban's master component.

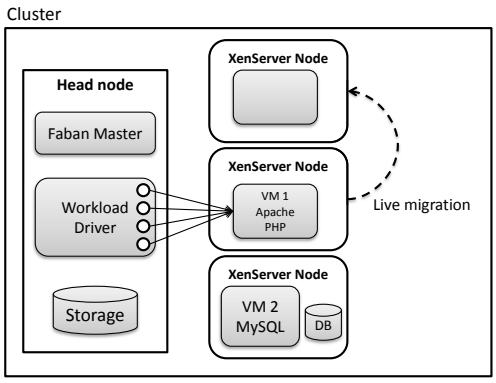

Fig. 1. Benchmarking architecture

4.4 Experimental Design

The overall objective of our experiments is to quantify slowdown and downtime experienced by the application when VM migrations are performed in the middle of a run. Specifically, we quantify application slowdown based on values generated by the above-mentioned SLA calculation.

In all experiments, cluster nodes and their interconnection were dedicated to the application under test. A migration experiment consisted of migrating a single VM between two dedicated physical machines. In each run, the chosen destination machine was different from the source machine in the previous run, i.e. a series of runs did not consist of migrating a VM back and forth between the same two machines.

Preliminary Experiments Exact VM sizes were obtained by preliminary experiments, in which we have run the application without performing any VM migration. We have driven load against Olio and gradually increased the number of concurrent users between runs, in 100 users increments, while using 2 identically sized VMs with 2 vCPUs and 2GB of memory. By analyzing the SLA (both 90th and 99th percentile of response times for all user actions), we have found that 600 is the maximum number of concurrent users that can be served by our SUT. We have observed memory and CPU usage to find the minimum VM sizes capable of serving 600 users. We have then aimed at reducing the size (vCPUs and memory) of the VMs to the minimum required to serve 600 users. Thus, in the final configuration the first VM, which exclusively hosts Apache/PHP, has 1 vCPU and 2GB of memory; the second VM, which hosts MySQL, has 2 vCPUs and 1GB of memory.

In the same preliminary experiments we have noticed performance issues when hosting the MySQL server on NFS. The application would not scale to more than 400 concurrent users, which has lead us to host MySQL in a local disk, thus scaling up to 600 concurrent users. For this reason, our experiments do

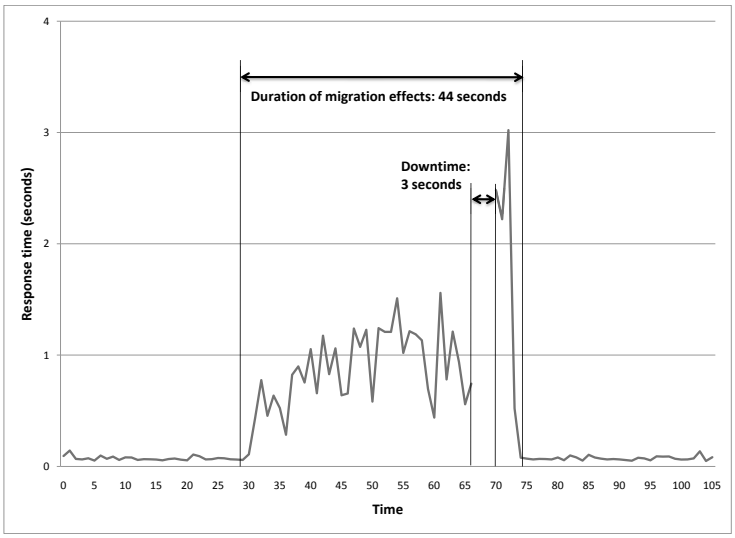

Fig. 2. Effects of a live migration on Olio's homepage loading activity

not include migrating the VM that hosts the database server, since XenServer requires all storage devices to be hosted in a network storage in order to perform live migrations.

Migration Experiments In our first set experiments with Olio we have performed 10-minute and 20-minute benchmark runs with 600 concurrent users. During these experiments, live migrations of the Web server VM were performed. The goal of experimenting with this load level is to evaluate how the pre-defined SLAs are violated when the system is nearly oversubscribed, but not overloaded. Also, we aim at quantifying the duration of migration effects and the downtime experienced by the application.

Subsequently, in a second round of experiments, we have run the benchmark with smaller numbers of concurrent users, namely 100, 200, 300, 400 and 500, aiming at finding a "safe" load level on which migrations can be performed at lower risks of SLA violation, especially when considering the more stringent 99th percentile SLA.

5 Results and Discussion

Overall, our experimental results show that overhead due to live migration is acceptable but cannot be disregarded, especially in SLA-oriented environments requiring more demanding service levels.

Figure 2 shows the effect of a single migration performed after five minutes in steady state of one run. A downtime of 3 seconds is experienced near the end of

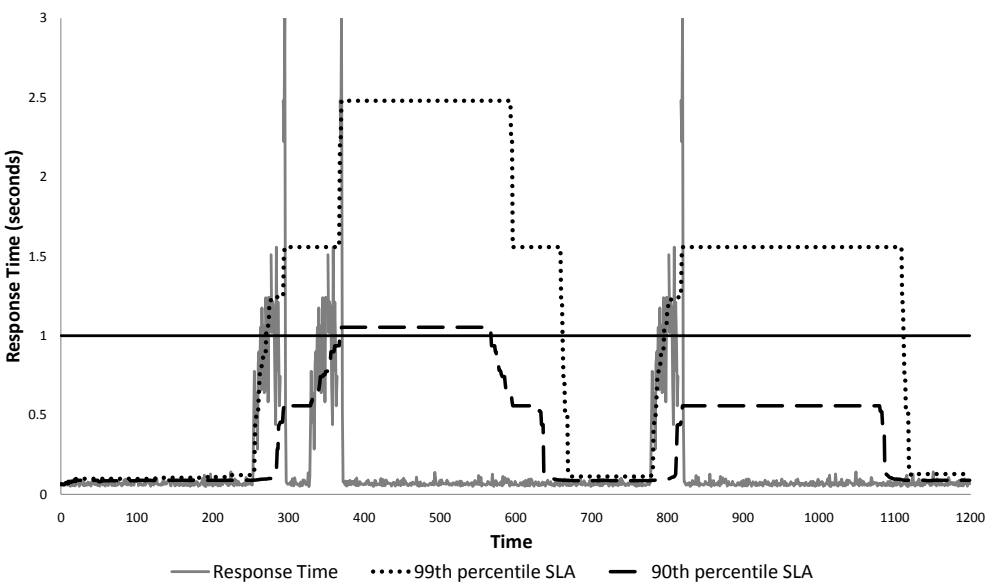

Fig. 3. 90th and 99th percentile SLA computed for the homepage loading response time with 600 concurrent users. The maximum allowed response time is 1 second

a 44 second migration. The highest peak observed in response times takes place immediately after the VM resumes in the destination node; 5 seconds elapse until the system can fully serve all requests that had initiated during downtime. In spite of that, no requests were dropped or timed out due to application downtime. The downtime experienced by Olio when serving 600 concurrent users is well above the expected millisecond level, previously reported in the literature for a range of workloads [3]. This result suggests that workload complexity imposes a unusual memory access pattern, increasing the difficulty of live migrating the virtual machine.

Figure 3 presents the effect of multiple migrations on the homepage loading response times. These result corresponds to the average of 5 runs. We report the 90th and 99th percentile SLAs. We can observe that the more stringent 99th percentile SLA is violated a short moment after the first migration is performed indicating that when 600 concurrent users are being served, a single VM migration is not acceptable. The 90th percentile SLA is not violated when a single migration occurs, but is violated only when two migrations are performed in a short period of time. This figure also indicates that more than one migration might not cause violation of the 90th percentile SLA. A way of preventing such violation is allowing sufficiently spacing between migrations in order to allow the SLA formula to generate normal response time levels. Thus, it is paramount that this information is employed by SLA-oriented VM-allocation mechanisms with

the objective of reducing the risk of SLA non-compliance in situations when VM migrations are inevitable.

From the above mentioned results we can conclude that, in spite of a significant slowdown and downtime caused by live migration of VMs, our SUT is resilient to a single migration when the system responsiveness is governed by the 90th percentile SLA. In other words, provided that migrations are performed at correct times, there is no cost associated with them. However, this is not the case for the 99th percentile SLA. For this reason, we have performed a new series of experiments with smaller number of concurrent users. The objective of such experiments is to gauge a safe level on which a migration could be performed with low risk of SLA violation.

Table 2. Maximum recorded 99th percentile SLA for all user actions when one migration is performed for 500, 400, 300, 200 and 100 concurrent users

Action	500	400	300	200	100
HomePage	0.32	0.18	0.25	0.25	0.13
Login	0.32	0.33	0.42	0.28	0.14
TagSearch	0.46	0.32	0.35	0.39	0.29
EventDetail	0.48	0.27	0.22	0.24	0.14
PersonDetail	1.53	0.62	0.69	0.61	0.32
AddPerson	2.28	1.00	1.51	1.73	0.66
AddEvent	2.26	1.02	1.30	1.81	0.98

Table 2 presents more detailed results listing maximum response times for all user actions as computed by the 99th percentile SLA formula when one migration was performed in the middle of a 10 minute run. In these runs the load varies from 100 to 500 users. In all cases, our SUT was able to sustain an acceptable performance even in the presence of a live migration of the Web server. For instance, the maximum value observed for homepage loading is 0.32 seconds, which corresponds to approximately 1/3 of the maximum value allowed, i.e. 1 second. The maximum value observed for the adding a new person to the system (2.28 seconds), which is more than half of the maximum allowed, but still does not indicate risk of SLA violation. These results indicate that a workload of 500 users is the load level at which a live migration of the Web server should be carried out (e.g. to a least loaded server) in order to decrease the risk of SLA violation.

6 Conclusion

Live migration of virtual machines is a useful capability of virtualized clusters and data centers. It allows more flexible management of available physical resources by making it possible to load balance and do infrastructure maintenance without entirely compromising application availability and responsiveness.

We have performed a series of experiments to evaluate the cost of live migration of virtual machines in a scenario where a modern Internet application is hosted on a set of virtual machines. Live migration experiments were carried out in scenarios where several levels of load were driven against the application.

Our results show that, in an instance of a nearly oversubscribed system (serving 600 concurrent users), live migration causes a significant downtime (up to 3 seconds), a larger value than expected (based on results previously reported in the literature for simpler, non Web 2.0 workloads) Also, this service disruption causes a pre-defined SLA to be violated in some situations, especially when two migrations are performed in a short period of time. On the other hand, we have found the most stringent SLA (99th percentile) can still be met when migrations are performed when the system load is slightly decreased to less concurrent users (500 in our case study).

In conclusion, we see a high potential of live migration applicability in data centers serving modern Internet services. This performance evaluation study is the first step towards the broader objective of studying the power of live migration of virtual machines for the management of data centers and clusters. We plan to use the insights of this study to develop smarter and more efficient SLA-based resource allocation systems.

6.1 Future Work

We are currently planning to conduct further migration experiments with new scenarios based on different application configurations, such as: using MemCached to alleviate database server load and allow for its migration. Moreover, we plan to expand our testbed to represent a large-scale clouds. As a consequence we expect to obtain a better generalization of the results, resulting in a performance model of live migration of virtual machines in clouds.

Acknowledgements We would like to thank Marcos Assunção, Alexandre di Constanzo and Mohsen Amini, Carlos Varela and the anonymous reviewers for their comments and assistance in improving this paper. This work is partially supported by grants from the Australian Research Council (ARC), the Australian Department of Innovation, Industry, Science and Research (DIISR) and the University of Melbourne Early Career Researcher (ECR) scheme.

References

1. 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: SOSP '03: Proceedings of the 19th ACM Symposium on Operating Systems Principles, New York, NY, USA, ACM (2003) 164–177
2. Buyya, R., Yeo, C., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems **25**(6) (June 2009) 599–616

3. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: NSDI'05: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation, Berkeley, CA, USA, USENIX Association (2005) 273–286
4. Milojicic, D., Douglis, F., Paindaveine, Y., Wheeler, R., Zhou, S.: Process migration survey. ACM Computing Surveys **32**(3) (2000) 241–299
5. Nagarajan, A.B., Mueller, F., Engelmann, C., Scott, S.L.: Proactive fault tolerance for HPC with xen virtualization. In: ICS '07: Proceedings of the 21st Annual International Conference on Supercomputing, New York, NY, USA, ACM (2007) 23–32
6. Barbosa, A.C., Sauve, J., Cirne, W., Carelli, M.: Evaluating architectures for independently auditing service level agreements. Future Generation Computer Systems **22**(7) (July 2006) 721–731
7. Iyer, R., Illikkal, R., Zhao, L., Makineni, S., Newell, D., Moses, J., Apparao, P.: Datacenter-on-chip architectures: Tera-scale opportunities and challenges. Intel Technology Journal **11**(03) (2007)
8. Uhlig, R., Neiger, G., Rodgers, D., Santoni, A.L., Martins, F.C.M., Anderson, A.V., Bennett, S.M., Kagi, A., Leung, F.H., Smith, L.: Intel virtualization technology. Computer **38**(5) (2005) 48–56
9. Cherkasova, L., Gardner, R.: Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In: ATEC '05: Proceedings of the USENIX Annual Technical Conference, Berkeley, CA, USA, USENIX Association (2005) 24–24
10. Apparao, P., Iyer, R., Zhang, X., Newell, D., Adelmeyer, T.: Characterization & analysis of a server consolidation benchmark. In: VEE '08: Proceedings of the fourth ACM SIGPLAN/SIGOPS Iinternational Conference on Virtual Execution Environments, New York, NY, USA, ACM (2008) 21–30
11. Casazza, J.P., Greenfield, M., Shi, K.: Redefining server performance characterization for virtualization benchmarking. Intel Technology Journal **10**(3) (2006) 243–251
12. Zhao, M., Figueiredo, R.J.: Experimental study of virtual machine migration in support of reservation of cluster resources. In: VTDC '07: Proceedings of the 3rd International Workshop on Virtualization Technology in Distributed Computing, New York, NY, USA, ACM (2007) 1–8
13. Travostino, F., Daspit, P., Gommans, L., Jog, C., de Laat, C., Mambretti, J., Monga, I., van Oudenaarde, B., Raghunath, S., Wang, P.Y.: Seamless live migration of virtual machines over the man/wan. Future Generation Computer Systems **22**(8) (2006) 901–907
14. Sobel, W., Subramanyam, S., Sucharitakul, A., Nguyen, J., Wong, H., Patil, S., Fox, A., Patterson, D.: Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0. In: CCA 08: Proceedings of the 1st Workshop on Cloud Computing. (2008)
15. Apache Software Foundation: Olio. http://incubator.apache.org/olio
16. Sun Microsystems: Project Faban. http://faban.sunsource.net
17. Amazon Web Services LLC: Amazon Web Services. http://aws.amazon.com
18. Subramanyam, S., Smith, R., van den Bogaard, P., Zhang, A.: Deploying web 2.0 applications on sun servers and the opensolaris operating system. Technical report, Sun Microsystems (2009)