Performance Influence of Live Migration on Multi-Tier Workloads in Virtualization Environments

Xiaohong Jiang, Fengxi Yan, Kejiang Ye
College of Computer Science, Zhejiang University
Zheda Road 38#, Hangzhou 310027, China
{jiangxh, yanfengxi, yekejiang}@zju.edu.cn

***Abstract*—Live migration is a widely used technology for load balancing, fault tolerance, and power saving in cloud data centers. Previous research includes significant research work in the performance improvement of live migration. However, little work has been done to investigate the influence of live migration on virtual machine workloads that users care about most. We notice that these workloads can be classified into two categories: single-tier workloads and multi-tier workloads which is a typical type for internet applications. We conduct a series of deliberate experiments to investigate the influence of live migration on multi-tier workloads in a cloud environment and also on traditional physical machines for comparison. Our experimental results show that multi-tier workloads on virtual machines can work as well as those on traditional physical machines. However, in an unstable environment, if virtual machines migrate constantly, live migration will cause a profound performance decrease on multi-tier workloads. Also, it is best to avoid migrating virtual machines that are hosting memory intensive workloads in a virtualization environment due to bad downtime performance. Further, we perform experiments trying to find the turning point of the performance of a virtual machine, which might provide support evidence for future research on live migration policy.**

Keywords-virtualization; live migration; XEN; Muti-tier workload.

I. INTRODUCTION

In a cloud datacenter, virtualization technology is widely preferred because of its impressive advantages in cost savings, easy resource management, high resource utilization, high availability, and good scalability. Live migration [1] is a core technique to implement load balancing, fault tolerance, and power savings in a virtualization environment. Most virtualization systems such as XEN [2], KVM [3], and VMware [4] support the live migration of virtual machines. Many researchers have been attracted to the investigation of live migration performance [5, 6]. However, the influence of live migration on virtual machine workloads, especially complex interactive workloads, hasn't been considered. What type of workloads will be affected most by live migration? Which virtual machine (VM) should be migrated so that the influence on workloads will be as small as possible? These questions are important for data center management as they directly affect the Quality of Service (QoS).

There are many kinds of workloads in a cloud datacenter. We classify them into two categories: single-tier workloads and multi-tier workloads. A single-tier workload runs on one single host and does not exchange data with workloads on other hosts. Most traditional single machine applications belong to this category. Multi-tier workloads are composed of a set of workloads running on different hosts and are constantly interacting with each other through the network. Multi-tier workloads have the following obvious features:

- Group work.
  Multi-tier workloads are not alone. They are a group of workloads running on different hosts connected to each other and work together in a multiple tier architecture.
- Interactive.
  Multi-tier workloads interact with each other. For example, Tier A transfers data to Tier B, Tier B analyzes the data and transfers the result back to Tier A.
- Sensitive of Single-Node Failure.
  If one of the nodes in a multi-tier workload fails, the remaining workloads should be stopped and wait for the failed node to resume again.

A dynamic website is an example of a typical multi-tier workload, which is composed of a frontend web server and a backend database server. Dynamic websites are the main form of websites on the internet as they provide better communication between web users and the website. A dynamic website can capture web users' input, search or retrieve data from the database, return the data to the web server and display the data in the web browser in an easily understandable way. When a web user sends a HTTP request containing some parameters to the web server, the web server will execute scripts based on the parameters, make queries to the database, and format the result into HTML files, which will be transferred back to the client. Figure 1 shows the architecture of a dynamic website. In fact, most internet applications fall into the category of multi-tier workloads.

Some research work has been done to measure the influence of live migration on single-tier workloads [5, 6] instead of multi-tier workloads. However, in a real Cloud data center, multi-tier workload is one of the most commonly used

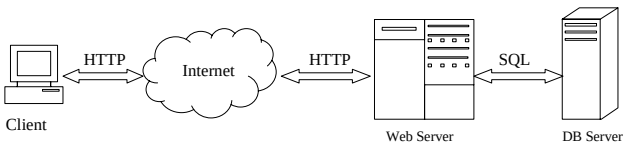

Figure 1. Architecture of Dynamic Website

 ISBN: 978-1-61208-216-5

application types other than the single-tier one. Research about the influence of live migration on multi-tier workloads has profound guiding significance to the choice of live migration policy. This paper is trying to determine how multi-tier workloads will behave when the host virtual machine is migrated to another physical machine in a virtualization environment. We conduct a series of experiments in a XEN virtualization environment with RUBiS [7, 8], a dynamic website benchmark. The experimental results show some useful information on virtual machine management that can be used as support evidence for a live migration policy.

The main contributions of this paper are summarized as follows:

- We study the performance effects of live migration on multi-tier workloads, including both web server and database server. And we analyze the migration overhead from downtime, total migration time, and the workload performance.
- We investigate the migration point issue or turning point issue at which the virtual machines should be migrated to other physical machines to avoid the performance degradation. It is the best migration point to reduce both the migration overhead and workload overhead.
- The experimental results show some meaningful suggestions to real cloud computing environments, and it is also meaningful to the further migration strategy development. For example, the memory-intensive workloads should avoid migrating first.

The rest of this paper is organized as follows. Section 2 gives a brief introduction to the benchmark "RUBiS" and the XEN hypervisor. Section 3 describes our experimental design. Section 4 describes the experimental results and our analysis. Finally, we summarize our conclusion in Section 5.

II. BACKGROUND

A. RUBiS Benchmark

RUBiS (Rice University Bidding System) [7, 8] is a free and open source benchmark of dynamic websites developed by Rice University. Its prototype is eBay and it is designed to evaluate application design patterns and the scalability of application servers using MySQL as its database.

The benchmark implements the main functions of an auction website: browsing, registering, selling, and bidding. There are three kinds of user sessions: visitor, buyer, and seller. A visitor does not need to register and is only permitted to browse. Buyers and sellers need to register. A buyer can bid on items and check the list of his or her current bids as well as any competitive bidding and comments left by other users. A seller can register an item for sale, sometimes with a reserve price, and view the list of his or her selling list.

RUBiS can be accessed by users from a browser, but for convenience, RUBiS implements a client emulator tool, which can emulate common users of this auction site. In fact, the client emulator can create many user sessions randomly. During a user session, RUBiS mass generates URLs for this user based on a pre-defined workload, and sends HTTP requests based on these URLs. With this mechanism, the client emulator behaves just like a real user: browses the homepage, browses items from categories and regions, registers to become a user, bids or buys an item, registers an item for sale and views his or her bidding and selling history.

There are three versions of RUBiS: a PHP version, a Java servlets version and an EJB version, which are for different usage. In our experiment, we use the PHP version for three reasons. First, this version is easy to install, maintain and use, so we can concentrate on our experiments. Second, PHP is one of the most popular languages used in web applications nowadays. Third, the PHP server Apache and its database server MySQL have the typical architecture of a dynamic website.

B. XEN and Live Migration Technique

In our experiment, we use XEN [2, 9] as our virtual machine monitor (or called Hypervisor). XEN is an open source project developed and maintained by Xenoserver research project at Cambridge University. It is a layer of software running directly on computer hardware replacing the operating system, thereby allowing the computer hardware to run multiple guest operating systems concurrently. Because of its support for x86, x86-64, Itanium, Power PC, and ARM processors, XEN hypervisor is able to run on a wide variety of computing devices and supports various operating systems (for example, Linux, NetBSD, FreeBSD, Solaris, Windows, and other common operating systems) as guest operating systems running on the hypervisor.

A virtual machine running on XEN hypervisor can be migrated to another physical machine, using the cold migration or the live migration technique. Cold migration needs the migrated virtual machine to be shut down completely in order to transfer the virtual machine disk image to the destination physical machine. The migrated virtual machine is restarted only after the disk image transfer is completed. This kind of migration takes too much time and the migrated virtual machine is not available during the period of migration. If the migrated virtual machine is undertaking some interactive workloads with other virtual machines, all workloads must stop running because of the disconnection to the migrated virtual machine.

Live migration handles the migration of a virtual machine in three aspects: network, disk, and memory. The network migration is just an IP address redirection, and the disk migration can be solved with storage net-share technology (for example, NFS[10], SAN[11], and NAS[12]). The main problem is the memory migration. It is done in 4 phases:

Phase 1: Pre-migration and Reservation

Assume a virtual machine is about to migrate from host A to host B. The XEN hypervisor first makes sure that host B has enough resources to hold the virtual machine, and

 ISBN: 978-1-61208-216-5

then reserves an empty VM container on host B for the virtual machine to be migrated

Phase 2: Iterative Pre-Copy

Dirty memory is transferred to host B in time intervals called iterations. During the first iteration, all memory pages will be transferred from host A to host B. Subsequent iterations only transfer the dirtied memory generated during the previous iteration.

Phase 3: Stop-and-Copy

This phase comes when the XEN hypervisor thinks that the remaining dirty memory can be transferred in a very short time interval or that there have been too many iterations of pre-copy in the previous phase. The virtual machine on host A will be shut down and its remaining dirty memory and CPU state will be copied to the virtual machine on host B. Now there are 2 copies of the virtual machines, one on A and the other on B.

Phase 4: Commitment and Activation

Host B informs Host A that it is ready to start the new virtual machine, and some post-migration code runs to attach the disk driver and IP address to the new virtual machine. The new virtual machine starts and the migration is complete.

Live migration can proceed seamlessly when the migrated virtual machine is running, and the virtual machine only stops for a very short time to restart. This period of time is called downtime which is so short that users and workloads on the virtual machine would not even be aware of it.

III. EXPERIMENT DESIGN

Our experiments are conducted on 4 physical machines (PM1, PM2, Client Emulator Server and Storage Server), which are connected by an Ethernet with the bandwidth of 1000Mbps. A network with such a high bandwidth will not become a bottleneck in the network transmission in our experiments. Every physical machine has enough memory so that memory will not become a bottleneck, either. Each machine has 8 CPU cores, with a clock rate of 2.27GHz.

We use Apache as our Web Server, MySQL as the Database (DB) Server, Debian Linux as the Operating System (OS). On PM1 and PM2, we deploy XEN hypervisor 4.0 to manage the virtual machines in the experiments. The virtual machines are also installed with Debian Linux as their OSes.

The Storage Server is a SAN server. All the virtual machine images are stored in this SAN server which is shared by PM1 and PM2 with an iSCSI access interface.

Our experiments are conducted in 4 phases:

In phase 1, we measure the performance of RUBiS on physical machines. We turn off 6 CPU cores on PM1 and PM2 respectively so that we have 2 CPU cores left on each of them. The RUBiS Web Server is deployed on PM1 and the RUBiS DB Server on PM2. The Client Emulator Server runs the RUBiS Client Emulator program to emulate common users who would visit the RUBiS website through HTTP connections. The architecture is shown in Figure 2. We call this phase ***PHYSICAL MODE***.

In phase 2, we measure the performance of RUBiS on virtual machines. In order to compare with the previous set of experiments, we allocate two VCPUs for each virtual machine to get an equivalent configuration compared with the ***PHYSICAL MODE***. A virtual machine with 2 VCPUs will be created on PM1, running the RUBiS Web Server, we call this virtual machine VM1; another virtual machine with 2 VCPUs will be created on PM2, running the RUBiS DB Server, we call this virtual machine VM2. Then enough memory is allocated for VM1 and VM2, so that memory will not become a bottleneck. The Client Emulator Server still runs the RUBiS Client Emulator program. The architecture is shown in Figure 3. We call this phase ***VIRTUAL MODE.***

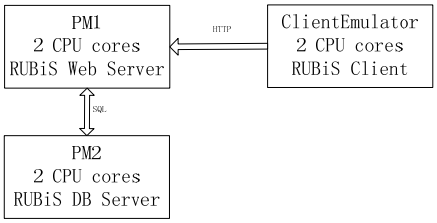

Figure 2. Experiment overlay of ***PHYSICAL MODE***

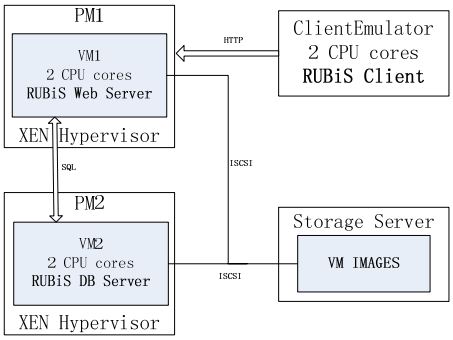

Figure 3. Experiment overlay of ***VIRTUAL MODE***

In phase 3, we measure the performance of RUBiS on virtual machines under live migrations. The experiment overlay is just the same as that in phase 2 (see Figure 3). But in the middle of every experiment, we conduct a live migration for VM2 which holds the RUBiS DB Server. VM2 migrates from PM2 to PM1. After the migration, we collect the migration time and downtime. This phase is named ***MIGRATION-DB MODE***.

Phase 4 is similar to phase 3. The difference is that VM1 is migrated instead of VM2. VM1 is migrated from PM1 to PM2. We collect the migration time and downtime of every single migration. This phase is named ***MIGRATION-WEB MODE.***

We collect the RUBiS throughput (requests per second) and the CPU usage rate in each experiment in all of the above 4 phases. Then we compare the collected data and

 ISBN: 978-1-61208-216-5

make a further analysis of the performance of multi-tier workloads in virtualization environment.

IV. Experimental Results And Analysis

A. Comparison of *PHYSCIAL MODE* and *VIRTUAL MODE*

The throughputs of RUBiS increasing with the number of clients in ***PHYSICAL MODE*** and ***VIRTUAL MODE*** are shown in Figure 4 and Figure 5 separately.

We can figure out from Figure 4 that the throughput of RUBiS benchmark in the ***PHYSICAL MODE*** goes up quickly before the number of clients reach 1400, slows down after reaching the number of 1400, and finally stabilizes after the number 1600.

Compared with Figure 4, it's easy to determine in Figure 5 that throughput in ***VIRTUAL MODE*** goes up almost the same way as that in ***PHYSICAL MODE***. It implies that virtual machines with equivalent configuration of hardware resources can achieve equivalent performance compared with traditional OS instances running on physical machines. In this circumstance, virtualization technology does not cause any obvious performance decrease for the multi-tier workloads.

Throughput reaches its maximum value when the number of clients is 1600, as the concurrent connections with RUBiS Web Server reaches the Apache Server's configured "MaxClients" attribute. In both ***PHYSICAL MODE*** and ***VIRTUAL MODE***, CPUs with 2 cores is powerful enough to run the RUBiS system, so CPU will not be a bottleneck.

The two curves shown in the graph series in Figure 6 and Figure 7 are CPU usage ratios of "Web Server" in the upper side and of "DB Server" in the lower side. From Figure 6 and Figure 7, we can determine that both the RUBiS Web Server and the DB Server use a small fraction of CPU even if throughput reaches its peak value.

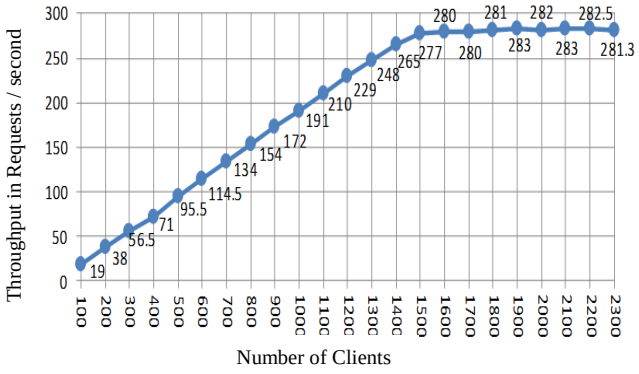

Figure 4. Throughput of RUBiS in ***PHYSICAL MODE***

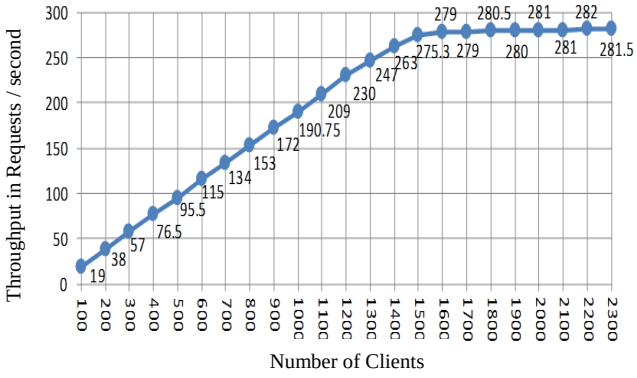

Figure 5. Throughput of RUBiS in ***VIRTUAL MODE***

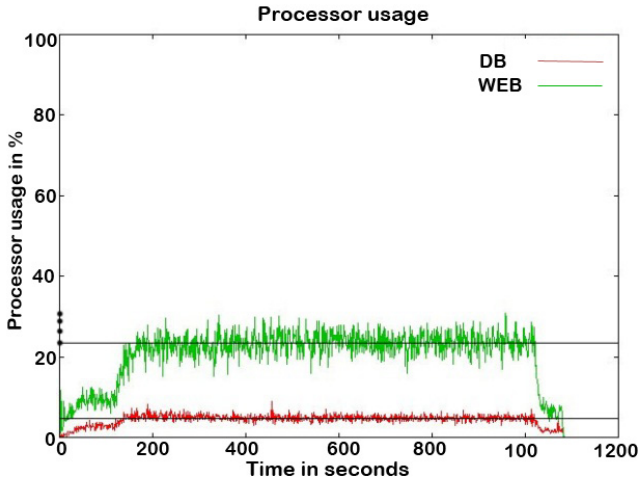

Figure 6(a). CPU usage when clients = 1300

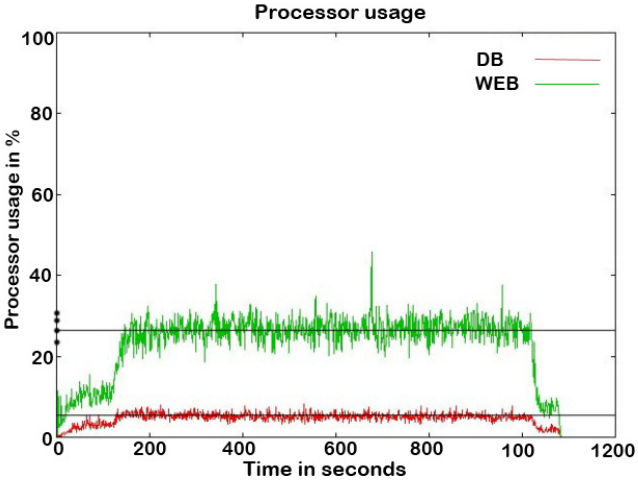

Figure 6(b). CPU usage when clients = 1400

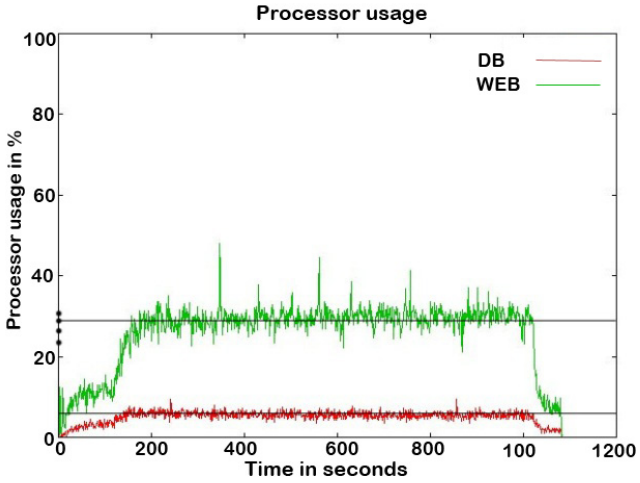

Figure 6(c). CPU usage when clients = 1500

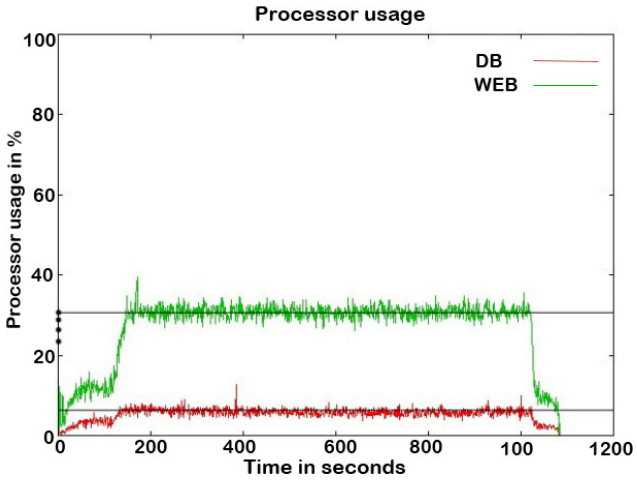

Figure 6(d). CPU usage when clients = 1600

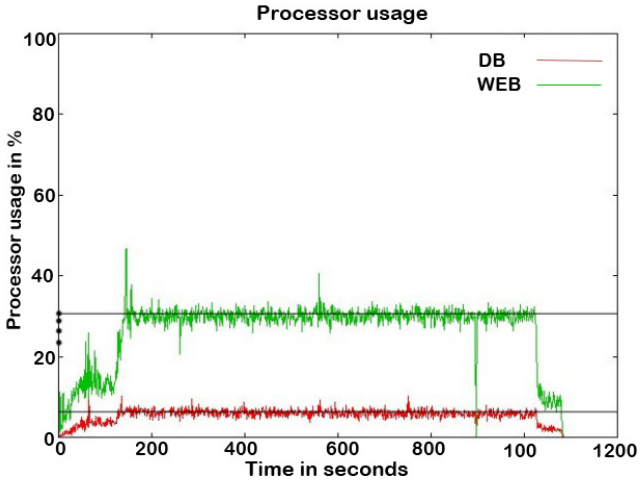

Figure 6(e). CPU usage when clients = 1700

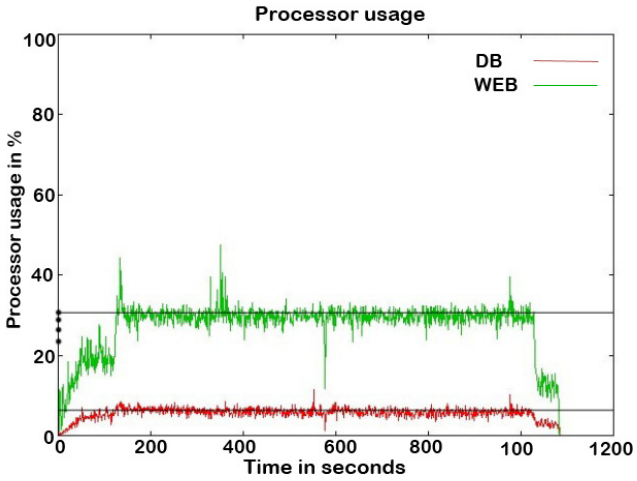

Figure 6(f). CPU usage when clients = 2300

Figure 6. CPU usage ration as a function of time in seconds in ***PHYSICAL MODE***

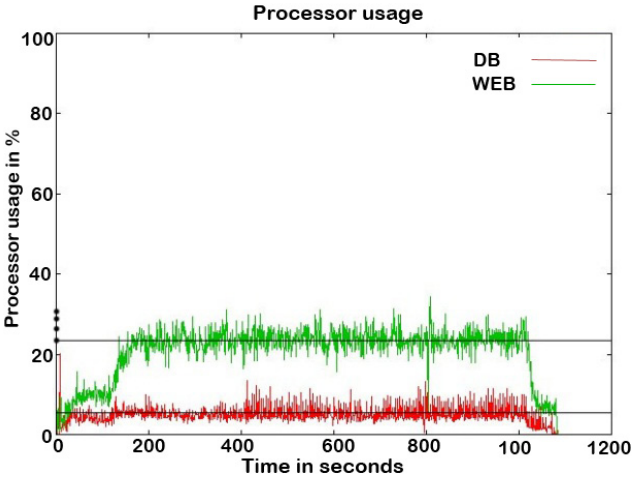

Figure 7(a). CPU usage when clients = 1300

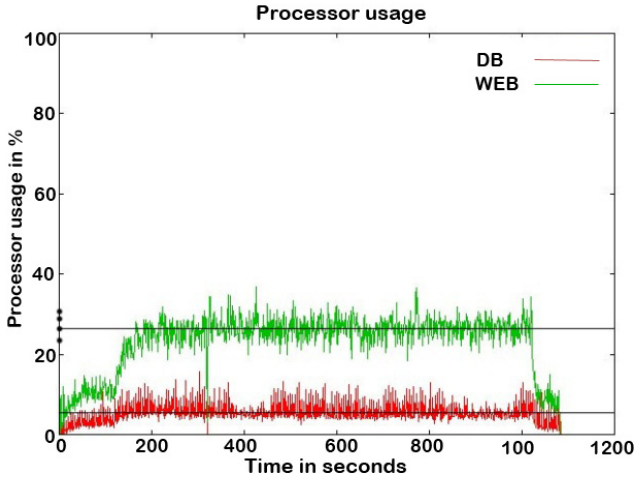

Figure 7(b). CPU usage when clients = 1400

 ISBN: 978-1-61208-216-5

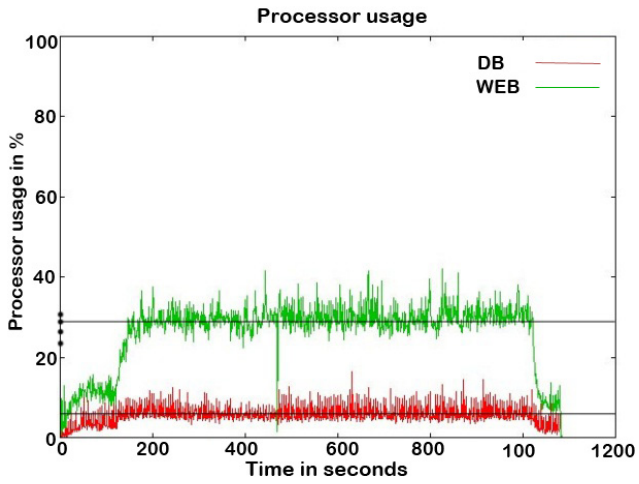

Figure 7(c). CPU usage when clients = 1500

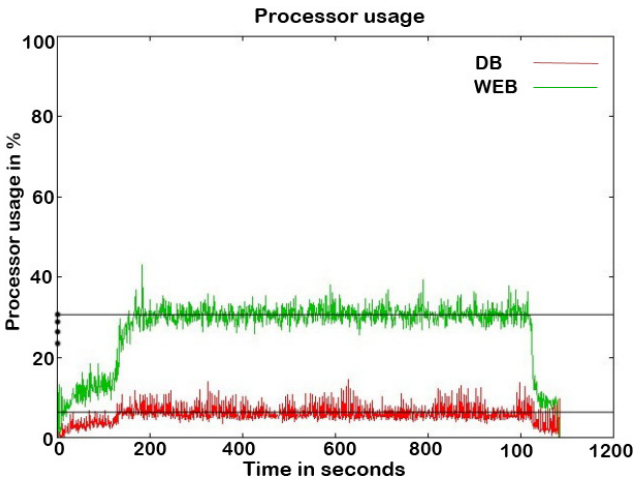

Figure 7(d). CPU usage when clients = 1600

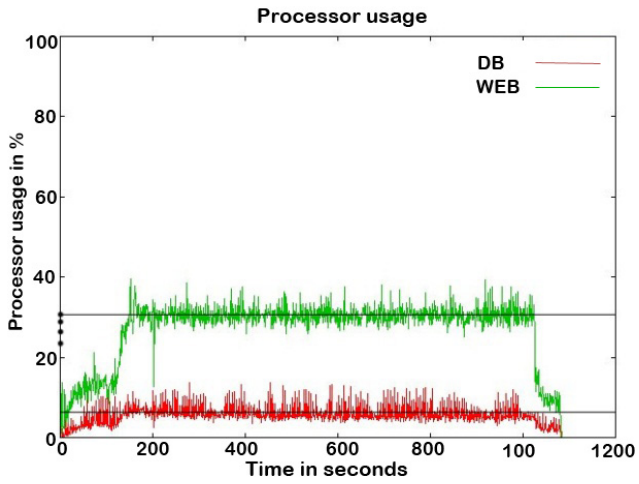

Figure 7(e). CPU usage when clients = 1700

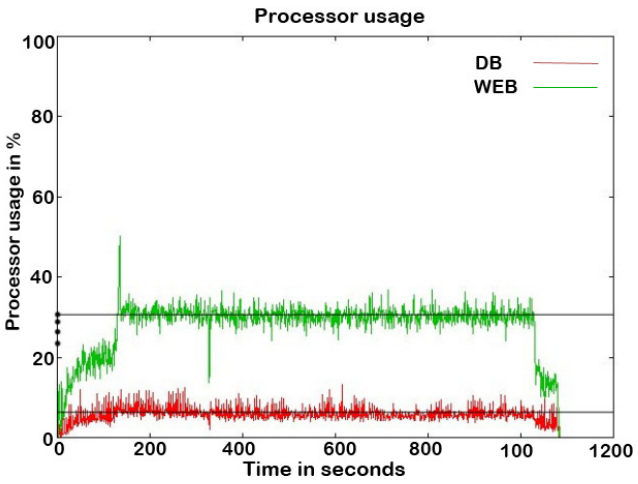

Figure 7(f). CPU usage when clients = 2300

Figure 7. CPU usage ratio as a function of time in seconds in ***VIRTUAL MODE***

Images in Figure 6 and Figure 7 are too small and it is hard to tell the number of CPU usage ratio, so we drew some sub lines to indicate the average CPU usage ratio comparatively. Figure 6 shows that the CPU usage ratio increases with the number of clients and reaches the maximum value when the client number reaches 1600 or more. The peak value of CPU usage is 31% for the RUBiS Web Server and 6% for the DB Server.

Figure 7 shows that in ***VIRTUAL MODE***, the CPU usage ratio is very similar to that in ***PHYSICAL MODE***. It also increases to the peak value when the client number reaches 1600 at about 31% for the RUBiS Web Server and 6% for the RUBiS DB Server.

From the above figures, the virtual machines show demonstration of wonderful performance: with equivalent configuration of hardware, they perform as well as the physical machines and do not consume more CPU resource than physical machines, even when running multi-tier workloads. We conclude that when multi-tier workloads are deployed on virtual machines, they can work as well as that on physical machines, without any extra CPU consumption.

However, we notice that the above conclusion for the ***VIRTUAL MODE*** can be drawn only in a somewhat stable circumstance. What will the result be if workloads run in an unstable circumstance? For example, how will the performance of multi-tier workloads be influenced when the host virtual machine is migrated? Experiments comparing ***MIGRATION-DB MODE*** *and* ***MIGRATION-WEB MODE*** try to answer this question and provide evidence support for a migration policy.

*B. Comparison of **MIGRATION-DB MODE** and **MIGRATION-WEB MODE***

In this subsection, we analyze the migration performance of virtual machines running RUBiS Web

 ISBN: 978-1-61208-216-5

Server and DB Server respectively. These two migration experiments show very different effects in migration time, downtime and throughput when migrating Web Server and DB server.

We first analyze the migration time difference. Figure 8 shows the memory usage in ***VIRTUAL MODE***. We obtain the memory usage when the client number is 1600 which is the turning point of the throughput. From Figure 8, we can see that the RUBiS Web Server (upper curve in Figure 8) consumes more memory than the DB Server (lower curve in Figure 8). So it is clear that the Web Server is more memory intensive than the DB Server. As mentioned in Section II, memory migration is the main task in virtual machine migration compared with network migration and disk migration, and is the decisive factor for migration time, downtime and throughput. So it's easy to jump to the conclusion that migration of virtual machine hosting memory intensive workloads will lead to longer migration time and downtime due to the migration of more dirty memory. However, experiment results turn out to be different from the above imprudent conclusion.

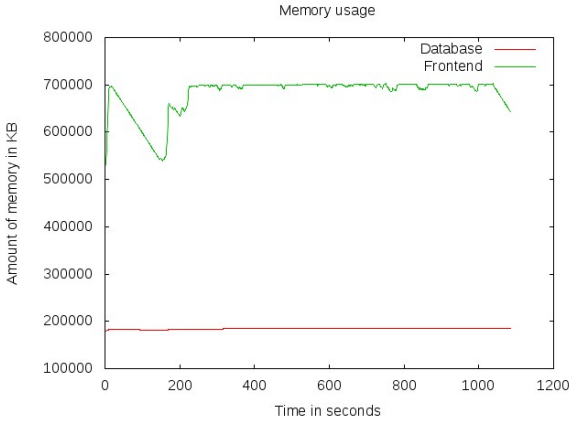

Figure 8. The memory usage graph when client=1600 in ***VIRTUAL MODE***

Figure 9 shows the migration time both in ***MIGRATION-DB MODE*** and ***MIGRATION-WEB MODE***.

The migration time in ***MIGRATION-WEB MODE*** is longer than that in ***MIGRATION-DB MODE*** when the client number is less than 1100, which is in accord with our prior intuitive conclusion. However, the migration time in ***MIGRATION-WEB MODE*** becomes the shorter one when the client number increases larger than 1100. Given the fact that the RUBiS Web Server is more memory intensive and more memory can be dirtied during the migration in the ***Iterative Pre-Copy*** phase in migration, the RUBiS Web Server will spend more time to copy the dirty memory than the DB Server. So it's very easy to understand why ***MIGRATION-WEB MODE*** has longer migration time. But why does it becomes the shorter one when the client number grows larger than 1100? In order to answer this question, we need to analyze the phases occurring in the live migration.

There are in total 4 phases in the live migration: 1) Pre-migration and reservation; 2) Iterative pre-copy; 3) Stop-and-Copy; 4) Commitment and Activation. Especially two conditions in the 2nd phase can trigger the 3rd Stop-and-Copy phase. One is the number of small dirty pages falling below the threshold; usually the dirty pages will become less when the dirty pages migrate by round. The second condition is the restriction of the number of iterations, in which when the number of iterations reaches a threshold, the virtual machine has to stop and copy all the remaining dirty memory. This happens when the dirty memory cannot be diminished as the iterative migration is performed.

In our experiment, as shown in Figure 9, there are very few clients accessing the Web Server at the beginning, so the memory used is very little. But when the client number increases, the dirty memory also increases and finally becomes a very large overhead. Because the RUBiS Web Server is memory intensive, the RUBiS Web Server has much more memory dirtied during the migration than the DB Server. The first condition of Stop-and-Copy that achieves a small dirty memory working set cannot be satisfied because dirty memory is generated faster than the memory has been migrated. So the virtual machine hosting Web Server ends the Iterative Pre-Copy phase in advance and makes the total migration time relatively shorter than the DB virtual machine when the client number increases more than 1100.

It can also be validated in Figure 10, from which we find the downtime in ***MIGRATION-WEB MODE*** is much longer than that in ***MIGRATION-DB MODE*** because the dirty memory working set of the Web server is larger than the DB server. It consumes more time to migrate the last of the dirty memory and incurs longer downtime. On the other hand, the DB server iterates more round cycles and the dirty memory can be relatively less, so the downtime can be short.

Based on the above evidence, we can conclude, contrary to our intuition, that the migration time of a VM hosting Web server is shorter than that of a VM hosting DB server when the client number is larger than a specific size. Nevertheless, it's better not to migrate the virtual machine hosting memory intensive workloads as the Web server in our experiment due to longer downtime.

In realistic situations, to achieve different goals of migration, we should use different methodologies accordingly. If we need to keep a stable performance of the workload involved, we should migrate the VM that is not memory intensive, because this would guarantee shorter downtime. Otherwise, if we want to finish the migration as soon as possible and keep a stable performance from the entire Cloud datacenter's sight, we should migrate the memory intensive ones because shorter migration time will occur.

Figure 11 depicts the throughputs in the last three modes. The throughput in ***MIGRATION-DB MODE*** and ***MIGRATION-WEB MODE*** is much smaller than that in the first two modes. During the downtime, the migrating virtual machine is entirely disconnected, and all clients'

 ISBN: 978-1-61208-216-5

accessing RUBiS will fail. What's more, the throughput in ***MIGRATION-WEB MODE*** is much less than that in the ***MIGRATION-DB MODE*** because of its longer downtime in migration. So it's better to avoid migrating virtual machines running memory intensive workloads (the Web server virtual machine in our experiment).

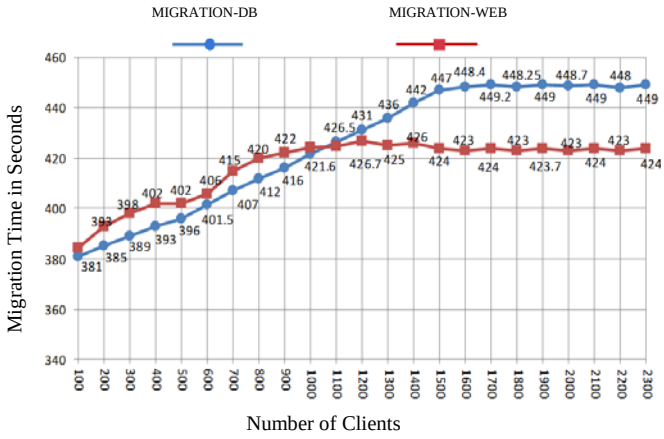

Figure 9. Migration Time as the Client Number Increases

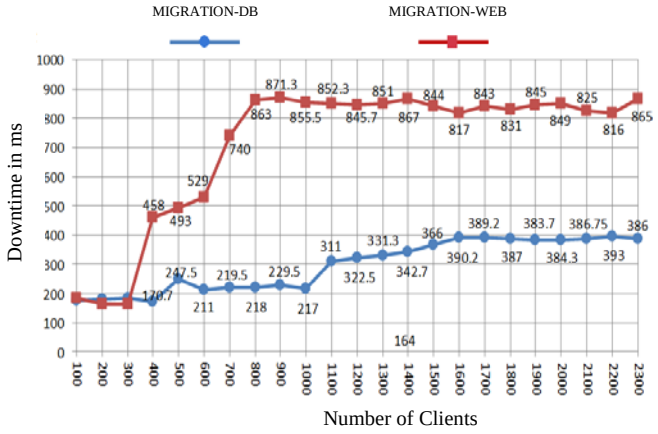

Figure 10. Downtime Time as the Client Number Increases

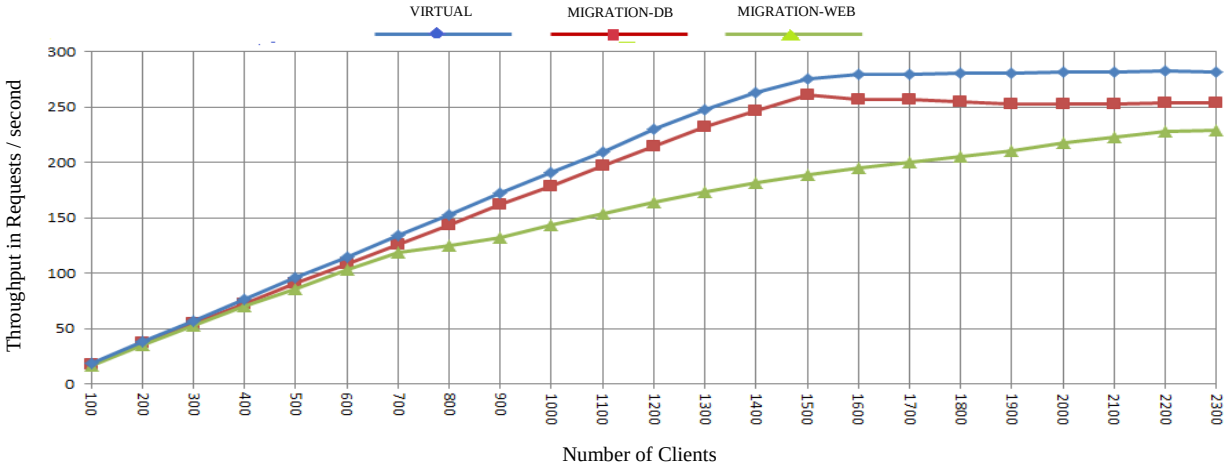

Figure 11. Throughput in ***VIRTUAL MODE, MIGRATION-DB MODE*** and ***MIGRATION-WEB MODE***

Based on the above analysis of ***MIGRATION-DB MODE*** and ***MIGRATION-WEB MODE***, we can determine that live migration indeed has an adverse effect on the running of multi-tier workloads. It's better to avoid the live migration of virtual machines as much as possible. When live migration cannot be avoided on demand of load balancing, fault tolerance, or power saving, it's better to not migrate the virtual machine hosting memory intensive workloads.

C. Analysis of Live Migration Point

From the above experiments, we can conclude that the performance of multi-tier workloads running on a virtual machine can be affected by the live migration process with different degrees. However, live migration of virtual machines indeed happens frequently in cloud computing environments to achieve the goals of dynamic resource management. For example, when the physical machine is nearly exhausted of CPU resource, it is better to migrate some of the virtual machines on this physical machine to other physical machines, because the lack of CPU resource also decreases the workloads' performance and might even lead to application failure. After the virtual machine is migrated to the physical machine rich with CPU resource, the performance of the physical machine will return to the normal level, and eventually avoid server or application failure. In this subsection, we will investigate performance issues in such scenarios.

In order to make the most of the CPU and start migration only when necessary, we should find a performance turning point(we name it **T**) of the application when the physical machine is about to be fully

loaded. The applications running on virtual machines work well before point **T**, and will turn bad after **T.** The turning point **T** might be a proper point to migrate the virtual machine on the nearly fully loaded physical machine. We conduct the following experiment to determine the **T** point specifically in our system.

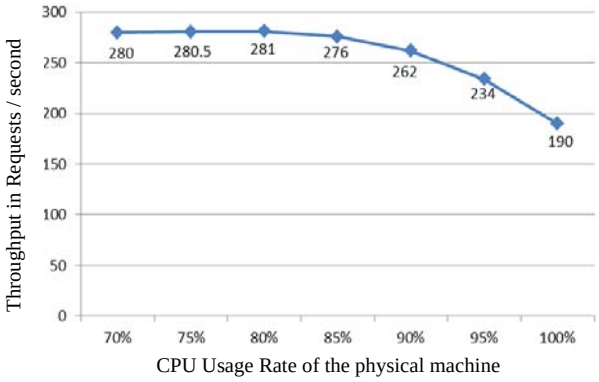

Figure 12. Throughput under different CPU usage rate when client=1600

First, we run VM1 on PM1 and VM2 on PM2 with both PM1 and PM2 having abundant CPU resource. The RUBiS Client emulates 1600 client sessions. And we can get a throughput of 281 requests per second.

Then, we start running a CPU intensive program on PM1, which can use up as much of the CPU resource as we set. We conduct a series of experiments under different CPU usage rates, and get the throughput accordingly. The results are shown in Figure 12. The x axis indicates the CPU usage rate of the physical machine hosting the virtual machines running the benchmark RUBiS. The y axis indicates the throughput of the multi-tier workloads.

From Figure 12, we can determine that the throughput of the virtual machine hosting RUBiS starts to decrease at 80%. So 80% CPU usage rate might be the **T** point. So Xen's migration policy might be improved to start live migration only when necessary at the turning point of 80% CPU usage rate. It is better for a Cloud administrator to find the turning point specifically in his Cloud data center to make the live migration policy more efficient. This experiment provides a basis for further research work in finding a proper T point of live migration in the cloud data center, which might be more precisely defined as the sub-healthy state of the virtual environment.

V. RELATED WORK

The workload performance issue incurred by virtualization technology in cloud computing environments has been widely investigated. Researchers have studied the performance overheads from both a single virtual machine perspective [2, 13, and 14] and a multiple virtual machine perspective [15, 16]. However, there is relatively little work referring to multi-tier workloads with interactive characterization.

A Multi-tier application is a typical kind of internet workload and has specific characterization. Urgaonkar et al. presented an analytical model for this application by using network of queues [17]. Bi et al. employed a hybrid queuing model to understand the performance of virtualized multi-tier applications and determine the number of virtual machines at each tier in a virtualized application [18]. However, they didn't consider the factors of live migration.

Live migration of virtual machines is used widely in today's cloud data center to achieve the goal of load balancing, fault tolerance, and saving energy. Xen and VMware primarily use the pre-copy technology to implement the live migration of virtual machines [1, 19]. After that many efforts have been made to improve the performance of live migration. Hines et al. presented a post–copy technique to implement the live migration of virtual machines which is different with the pre-copy technique [20]. Jin et al. proposed an adaptive memory compression method to reduce the overhead of memory transfers and improve the migration performance [21]. Liu et al. used the technique of full system trace and replay to optimize the migration efficiency [22]. Luo et al. solved the problem of whole-system migration in which both the memory and disk states were migrated [23]. Ye et al. investigated the issue of multiple virtual machine migration and proposed a method based on resource reservation to optimize the overall migration efficiency [24]. In order to evaluate the performance of different live migration techniques, Huang et al. designed a benchmark for live migration [25]. However, all the above work has not considered the characteristics of the multi-tier virtual machine workloads.

VI. CONCLUSION

We have made a deliberate analysis about multi-tier workloads and found that very little work has been done to measure the influence of virtualization technology especially live migration on multi-tier workloads running on VMs. Because multi-tier workloads comprise most of the workloads in a real Cloud data center, determining the influence detail is significant to the choice of live migration policy.

To achieve this goal, we have conducted a comprehensive performance analysis of multi-tier workloads in a virtualization environment, especially the performance characterization under the live migration. Based on the experimental analysis, we find that virtualization technology, especially the live migration technique, has some hidden influences on multi-tier workloads. The experimental results tell us that virtual machines can achieve nearly equivalent performance with the same system configuration compared with traditional OS instances running on physical machines. That is to say that multi-tier workloads can work well in a virtual machine environment. However, the live migration of virtual machines can cause some performance decrease

 ISBN: 978-1-61208-216-5

due to migration overhead and the downtime during which the migrating virtual machine should be shutdown. This decrease is especially obvious to those virtual machines running memory intensive multi-tier workloads. It is necessary to balance the migration benefits and overheads. In order to answer the question when the virtual machines should be migrated, we designed an experiment to find the proper migration point under different hardware resource configuration (for example, CPU utilizations in the experiment). Experimental results show that at the turning point of 80% CPU usage rate, the migration can benefit the workloads' performance. A Cloud administrator should determine the turning point of their Cloud data center specifically and adjust the live migration policy.

Future work will include developing adaptive migration framework for cloud computing and designing intelligent live migration strategies to improve the overall workloads' performance.

VII. ACKNOWLEDGMENT

This work is funded by the National High Technology Research 863 Major Program of China (No.2011AA01A207) and MOE-Intel Information Technology Foundation (No.MOE-INTEL-11-06). We would like to thank the reviewers for their insightful comments. We would also like to thank Mr. Jeff Wood for his careful revision of our paper.

REFERENCES

[1] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines", in Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2 (NSDI'05), Vol. 2. USENIX Association, Berkeley, CA, USA, pp. 273-286, 2005.

[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," ACM SIGOPS Operating Systems Review, vol. 37, no. 5, pp. 164–177, 2003.

[3] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "KVM: the Linux virtual machine monitor", in Proceedings of the Linux Symposium, vol. 1, pp. 225-230, 2005.

[4] C. Waldspurger, "Memory resource management in VMware ESX server", ACM SIGOPS Operating Systems Review, vol. 36 (SI), pp. 194, 2002.

[5] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation", in Proceedings of the international conference on Cloud Computing (CloudCom), pp. 254-265, 2009.

[6] S. Akoush, R. Sohan, A. Rice, A.W. Moore, and A. Hopper, "Predicting the performance of virtual machine migration", 2010 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 37-46, 2010.

[7] RUBiS home page. http://rubis.ow2.org/, [retrieved: March, 2012]

[8] C. Amza, A. Chanda, A.L. Cox, S. Elnikety, R. Gil, K. Rajamani, W. Zwaenepoel, E. Cecchet, and J. Marguerite, "Specification and implementation of dynamic Web site benchmarks," 2002 IEEE International Workshop on Workload Characterization, pp. 3-13, 2002.

[9] XEN community. http://www.XEN.org/, [retrieved: March, 2012]

[10] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network Filesystem", USENIX, 1985.

[11] J. Tate, F. Lucchese, and R. Moore, "Introduction to Storage Area Networks- Exhaustive Introduction into SAN", IBM redbook. www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf

[12] G. A. Gibson and R. V. Meter, "Network Attached Storage", Communications of the ACM, vol. 43, no. 11, pp. 37-45, 2000.

[13] B. Clark, T. Deshane, E. Dow, S. Evanchik, M. Finlayson, J. Herne, and J. Matthews, "Xen and the art of repeated research," USENIX annual Technical Conference, pp. 135-144, 2004.

[14] A. Menon, J. Santos, Y. Turner, G. Janakiraman, and W. Zwaenepoel, "Diagnosing performance overheads in the xen virtual machine environment," in VEE: Proceedings of the 1st ACM Conference on Virtual Execution Environments, pp. 13-23, 2005.

[15] M.F. Mergen, V. Uhlig, O. Krieger, and J. Xenidis, "Virtualization for high-performance computing", ACM SIGOPS Operating Systems Review, vol. 40(2), pp. 8-11, 2006.

[16] K. Ye, X. Jiang, S. Chen, D. Huang, and B. Wang, "Analyzing and modeling the performance in xen-based virtual cluster environment", 12th IEEE International Conference on High Performance Computing and Communications (HPCC), pp. 273-280, 2010.

[17] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An analytical model for multi-tier internet services and its applications", ACM SIGMETRICS Performance Evaluation Review, vol. 33(1), pp. 291-302, 2005.

[18] J. Bi, Z. Zhu, R. Tian, and Q. Wang, "Dynamic provisioning modeling for virtualized multi-tier applications in cloud data center", in Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 370-377, 2010.

[19] M. Nelson, B. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in Proceedings of the annual conference on USENIX Annual Technical Conference, p. 25, 2005.

[20] M. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pp. 51-60, 2009.

[21] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live virtual machine migration with adaptive memory compression," in Proceedings of the IEEE International Conference on Cluster Computing, pp. 1-10, 2009.

[22] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in Proceedings of the 18th ACM international symposium on High performance distributed computing, pp. 101-110, 2009.

[23] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and incremental whole-system migration of virtual machines using block-bitmap," in Proceedings of the IEEE International Conference on Cluster Computing, pp. 99-106, 2008.

[24] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, "Live migration of multiple virtual machines with resource reservation in cloud computing environments", in Proceedings of the 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 267-274, 2011.

[25] D. Huang, D. Ye, Q. He, J. Chen, and K. Ye, "Virt-LM: a benchmark for live migration of virtual machine", in Proceeding of the second joint WOSP/SIPEW international conference on Performance engineering, pp. 307-316, 2011.

 ISBN: 978-1-61208-216-5