2011 IEEE 4th International Conference on Cloud Computing

Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments

Kejiang Ye, Xiaohong Jiang, Dawei Huang, Jianhai Chen, Bei Wang

College of Computer Science, Zhejiang University
Hangzhou 310027, China
{yekejiang,jiangxh,tossboyhdw,chenjh919,wb}@zju.edu.cn

***Abstract*—Virtualization technology is currently becoming increasingly popular and valuable in cloud computing environments due to the benefits of server consolidation, live migration, and resource isolation. Live migration of virtual machines can be used to implement energy saving and load balancing in cloud data center. However, to our knowledge, most of the previous work concentrated on the implementation of migration technology itself while didn't consider the impact of resource reservation strategy on migration efficiency. This paper focuses on the live migration strategy of multiple virtual machines with different resource reservation methods. We first describe the live migration framework of multiple virtual machines with resource reservation technology. Then we perform a series of experiments to investigate the impacts of different resource reservation methods on the performance of live migration in both source machine and target machine. Additionally, we analyze the efficiency of parallel migration strategy and workload-aware migration strategy. The metrics such as downtime, total migration time, and workload performance overheads are measured. Experiments reveal some new discovery of live migration of multiple virtual machines. Based on the observed results, we present corresponding optimization methods to improve the migration efficiency.**

***Keywords*-virtual machine; live migration; resource reservation; performance;**

I. Introduction

Cloud computing [1, 2] has recently received considerable attention in both academic and industrial community as a new computing paradigm to provide dynamically scalable and virtualized resource as a service over the Internet. By this means, users will be able to access the resources, such as applications and data, from the cloud anywhere and anytime on demand. Currently, several large companies, such as Amazon, Google, Yahoo!, Microsoft, IBM, and Sun are developing their own cloud platforms for consumers and enterprises to access the cloud resources through services.

Recently, with the rapid development of virtualization technology [3, 4], more and more data centers use this technology to build new generation data center to support cloud computing [5, 6] due to the benefits such as server consolidation, live migration, and resource isolation [7]. Live migration of virtual machines [8, 9] means the virtual machine seems to be responsive all the time during the migration process from the clients' perspective. Compared with traditional suspend/resume migration, live migration holds many benefits such as energy saving, load balancing, and online maintenance. Many live migration methods are proposed to improve the migration efficiency [10–13]. As the live migration technology widely used in modern cloud computing data center, live migration of multiple virtual machines becomes more and more frequent. Different from the single virtual machine migration, the live migration of multiple virtual machines faces many new problems, such as migration failures due to the insufficient resources in target machine, migration conflicts due to the concurrent migrations, and the migration thrashing due to the dynamic changes of virtual machine workloads. All the above issues should be overcome to maximize the migration efficiency in virtualied cloud data center environments.

In this paper, we study the live migration efficiency of multiple virtual machines from experimental perspective and investigate different resource reservation methods and migration strategies in the live migration process. We first describe the live migration framework of multiple virtual machines with resource reservation technology. Then we perform a series of experiments to investigate the impacts of different resource reservation methods on the performance of live migration in both source machine and target machine. Additionally, we also analyze the efficiency of parallel migration strategy and workload-aware migration strategy. The metrics such as downtime, total migration time, and workload performance overheads are measured. Experiments reveal some new discovery of live migration of multiple virtual machines. Based on the observed results, we present corresponding optimization methods to improve the migration efficiency.

The rest of the paper is structured as follows. In Section II, we introduce the resource reservation based live migration framework of multiple virtual machines. In Section III, we perform a comprehensive evaluation and analysis on different resource reservation methods and migration strategies and propose corresponding optimization methods. Section IV presents the related work. Finally we give our conclusion and future work in Section V.

978-0-7695-4460-1/11 $26.00 © 2011 IEEE
DOI 10.1109/CLOUD.2011.69

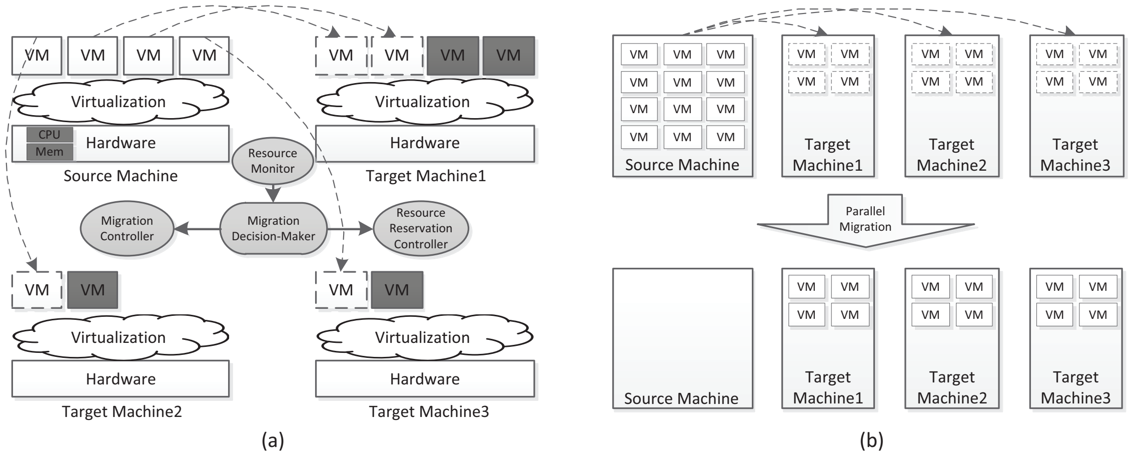

Figure 1. Resource Reservation Based Live Migration of Multiple Virtual Machines: (a) Live Migration Framework of Multiple Virtual Machines with Resource Reservation Technology. The VM in dark represents the reserved resources in the form of virtual machines in the target machines, the CPU and Mem in dark represents the reserved resources in the source machine; (b) An Example of Parallel Live Migration of Multiple Virtual Machines.

II. Resource Reservation Based Multiple VMs Live Migration

In this section, we propose the resource reservation based live migration framework of multiple virtual machines, as shown in Figure 1.

A. The Framework of VM Live Migration

Figure 1(a) illustrates the live migration framework of multiple virtual machines with resource reservation technology. It consists of four main modules: *Migration Decision-Maker*, *Migration Controller*, *Resource Reservation Controller*, and *Resource Monitor*. The resource reservation in both source machine and target machine are illustrated in the figure. While Figure 1(b) shows an example of parallel live migration of multiple virtual machines. In this example, the virtual machine workloads in source machine are migrated to target machines simultaneously according to the migration request. After the migration, source machine is vacated for particular purposes such as online maintenance or deployment of new workloads, each target machine holds 4 virtual machines.

Migration Decision-Maker: is a key module in the live migration framework of multiple virtual machines. It is responsible for making effective migration decision. The effectiveness and efficiency of migration strategies can reflect the quality of this module. What's more, this module can also have some intelligence to make efficient migration strategies by learning the historical data using machine learning methods. Currently, some simple migration strategies are included in this module, they are sequential migration, parallel migration, and workload-aware migration strategies.

Migration Controller: controls the real migration process. It will choose the right target machine from the candidate target machine list and trigger the migration at particular time. This module relies on the *Migration Decision-Maker* module, and executes the migration strategies made by the *Migration Decision-Maker* module. It is implemented by encapsulating existing migration interfaces such as *xm migration –live* in Xen virtualization platform.

Resource Reservation Controller: implements different resources reservation strategies for both source machine and target machine, such as CPU resource and memory resource reservation in the source machine or the whole virtual machine resource reservation in the target machine. This module is very important in the live migration of multiple virtual machines to avoid migration failures because of the insufficient resources in the target machine for the migrated virtual machine. The resource reservation in the source machine is implemented by dynamically adjusting the migrated virtual machine's CPU cycle and the memory resource. And the resource reservation in the target machine is implemented by temporarily creating particular virtual machines to occupy a certain number of system resources for the migrated virtual machines.

Resource Monitor: is responsible for monitoring the resource status of both virtual machines and physical machines, including the resource utilization, virtual machine configuration information (workload characteristics, VCPU number, memory size, image size) which is essential to make migration decisions. It can also be used to analyze the workload stability which is useful to avoid migration thrashing. This module is implemented in the *vTestkit* toolkit that is a performance management tool for virtualization environments [14].

B. The Resource Reservation Methods

We implement the resource reservation methods in both source machine and target machine. The reserved resource in source machine includes CPU and memory resource, while in target machine includes the whole virtual machine resources. Because the virtual machine images are usually stored in a separate storage server (such as SAN) containing large storage space, the disk space reservation is not considered in this paper.

1) Resource Reservation in Source Machine: The live migration process consumes some system resources in the source machine. By flexibly controlling the system resources in the source machine for the migrated VM, the live migration performance can be improved. We implement the CPU resource reservation of source machine in Xen virtualization platform by adjusting the *CAP* value to limit the maximum utilization of the CPU resource. And the memory resource reservation is implemented by dynamically adjusting virtual machine memory size.

2) Resource Reservation in Target Machine: The available resources in the target machine may affect the migration of virtual machines. We write a CPU-bound program in C language to continuously consume 100% CPU resource. According to the migrated virtual machine workloads, we create a certain number of virtual machines running CPU-bound workloads inside to achieve the goal of CPU resource reservation. The memory resource reservation is implemented by creating one or multiple virtual machines allocated a certain amount of memory resource. When the migrated virtual machines are ready to migrate to the target machine, these virtual machines are shutdown to vacate system resources for the migrated virtual machines. In general, the insufficient resources in the target machine will result in the failure of migration which means the migrated virtual machine can not be successfully migrated to this physical machine. However, by using the method of resource reservation in the target machine, if there are not enough resources for the reserved virtual machines, the physical machine should be removed from the list of candidate target machines. And another target machine will be selected from the candidate list to support the migrated virtual machines.

III. Experimental Evaluation & Analysis

A. Experimental Setup

The experiments are performed on two Dell OPTIPLEX 755 machines, with Intel Core2 Quad CPU at 2.4GHz and 2GB physical memory. One is for source migration server, the other is for target migration server. The virtual machine images are stored in the NFS (Network File System). We use Ubuntu 8.10 with kernel version 2.6.27 in domain0, and the version of Xen hypervisor is 3.3.1. The virtual machines are configured with 1VCPU and 512GB memory size. The benchmarks and workloads used are SPECjbb2005, IOzone, Sysbench, and Webbench.

There are usually three typical metrics to quantify the performance or efficiency of live migration technologies [15]: *downtime*, *total migration time*, and *workload performance overheads*. The *downtime* reflects the time during which the migrating virtual machine is stopped with no response. The *total migration time* reflects the period from migrating start to finish. The *workload performance overheads* reflects the performance degradation of the running virtual machine workloads because extra machine resources are consumed to perform the migration that affects the workload performance. All the above three metrics are measured in this paper to evaluate the efficiency of different resource reservation strategies and migration strategies.

In order to ensure the data precision, each of the showed experiment results were obtained via running benchmarks five times on the same configuration, the highest and lowest values for each test were discarded, and the remaining three values were averaged.

B. Performance Overheads of Live Migration of Single VM

Table I
The Workload Performance Overheads of Live Migration of Virtual Machines

	Normal	**Migration**	**Degradation**
SPECjbb(bops)	16195	14853	8.29%
Database OLTP(s)	66.14	88.14	33.26%
Disk I/O Write(Kb/s)	450863	415080	7.93%
Web I/O(Bytes/sec)	1991748	1413509	29.03%

Table II
The Downtime and Migration Time of Different Virtual Machine Workloads under Live Migration

	Downtime (ms)	**Migration Time (s)**
Idle VM	16	47.53
SPECjbb	460	66.39
Sysbench	18	118.75
IOzone	19	59.51
Webbench	27	53.91

In this subsection, we investigate the performance overheads of live migration of different virtual machine workloads. Table I shows the raw performance data of four typical workloads running in normal virtual machine and migrating virtual machine. From the table, we find that the workload performance has a certain degradation, with 8.29%, 33.26%, 7.93%, and 29.03% respectively. It is because when in the short downtime period of the migration process, the workloads are out of service that affects the overall performance. What's more, the migration process also consumes some system resource thereby affects the workload performance. Table II shows the downtime and total migration time of different workloads under live migration. From the table, we find the downtime is very short with about 20ms while the SPECjbb's downtime is relatively long (460ms). It is because the SPECjbb 2005 is a huge and complex workload compared with other three workloads and the dirty data of memory pages is generated very quickly which affects the downtime. We also can find that the total migration time of

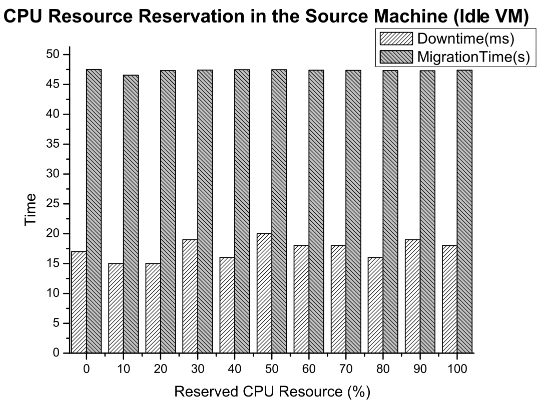

Figure 2. CPU Resource Reservation in Source Machine when Migrating Idle Virtual Machine.

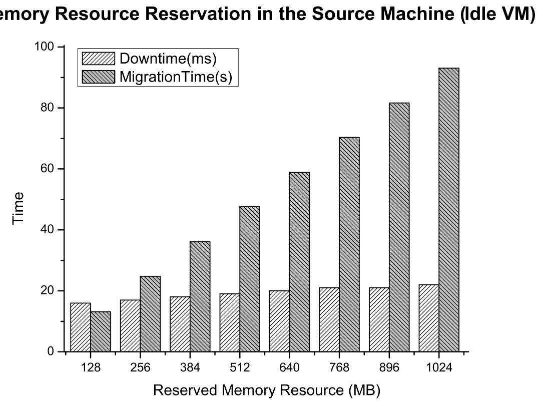

Figure 3. Memory Resource Reservation in Source Machine when Migrating Idle Virtual Machine.

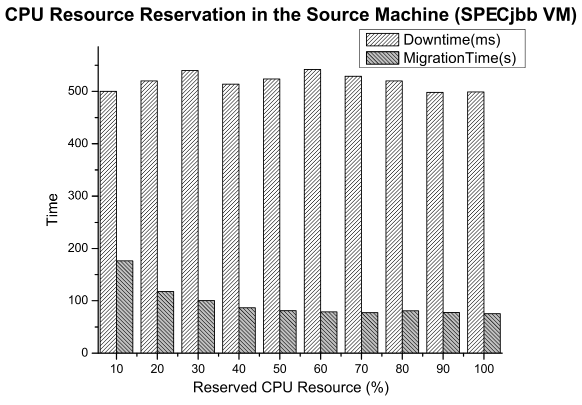

Figure 4. CPU Resource Reservation in Source Machine when Migrating SPECjbb Virtual Machine.

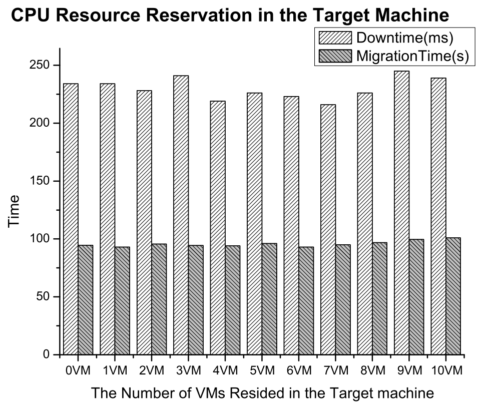

Figure 5. CPU Resource Reservation in Target Machine by Creating CPU-Bound Virtual Machines.

sysbench and SPECjbb is relatively longer. The reason is that both the two workloads consume relatively more memory resource which affects the total migration time.

C. Resource Reservation in Source Machine

In this subsection, we investigate the impact of resource reservation strategy in source machine on the live migration performance. Figure 2 and 3 illustrate the migration performance effects with different CPU and memory resource reservation strategies in source machine when migrating an idle virtual machine (running no workloads in the virtual machine). We adjust the *CAP* value of the migrated virtual machine to limit the CPU resources consumed by the migrated virtual machine. And the memory resource reservation is dynamically adjusted by the virtual machine memory size with *xm mem-set* interface in Xen. From the Figures, we find the reserved CPU resource has little effect on downtime and total migration time because the idle virtual machine consumes little CPU resource and is not sensitive to the CPU resource changes. While in Figure 3, the total migration time increases obviously as the virtual machine memory size increases from 128MB to 1024MB, and the downtime also increases but with a slow growth rate. It is because the more virtual machine memory incurs more amount of dirty data needed to transfer thereby prolongs both the downtime and total migration time.

To study the effect of reserved CPU resource on the live migration performance of virtual machine when running CPU intensive workloads, we perform another experiment with SPECjbb as the migrated virtual machine workload, as shown in Figure 4. Different from the idle virtual machine migration (as shown in Figure 2), the migration time is affected by the reserved CPU resource. When more CPU resource is reserved the shorter migration time is achieved. However when the reserved CPU resource exceeds 50%, no more performance improvement can be achieved since the live migration process only needs a certain amount of CPU resource (50% in this case), not the all resource.

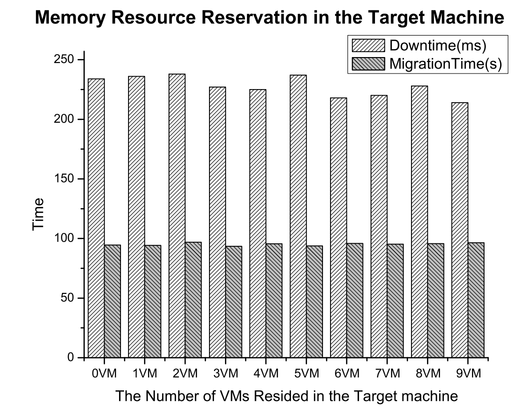

Figure 6. Memory Resource Reservation in Target Machine by Creating Idle Virtual Machines with Particular Memory Size.

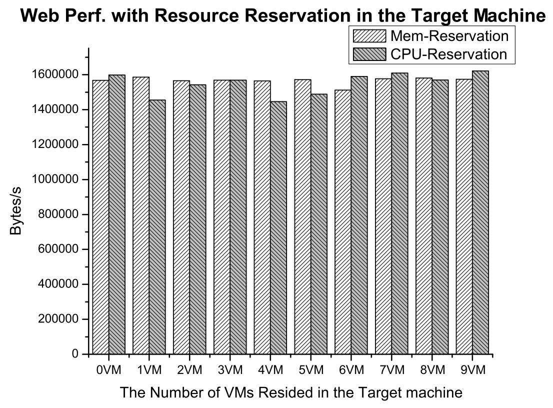

Figure 7. The Web Performance with CPU and Memory Resource Reservation in Target Machine.

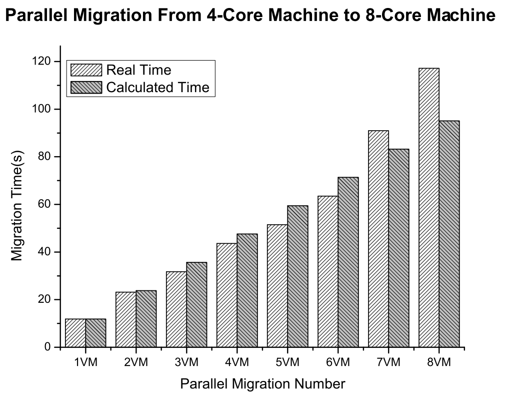

Figure 8. The Performance of Parallel Migration From 4-core Machine to 8-core Machine.

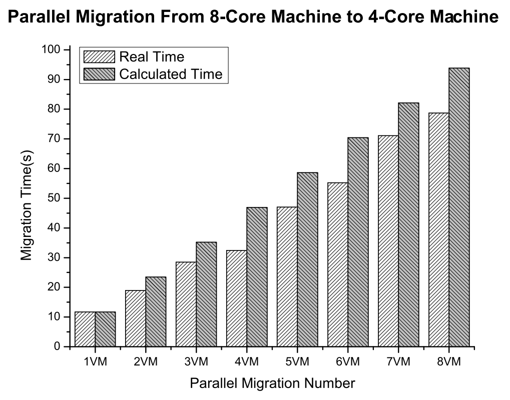

Figure 9. The Performance of Parallel Migration From 8-core Machine to 4-core Machine.

D. Resource Reservation in Target Machine

Different from the above subsection, in this subsection we study the resource reservation strategy in the target machine by creating virtual machines to reserve resources for the migrated virtual machines. The CPU resource reservation is implemented by creating CPU-bound virtual machines while the memory resource reservation is implemented by creating idle virtual machines with certain amount of memory resource.

Figure 5 and 6 show the performance effects of live migration with different CPU resource and memory resource reservation strategies in the target machine. Obviously, the downtime and total migration time maintain a stable state. Figure 7 shows the Webbench result when running an Apache web service in the migrated virtual machine which achieves the similar stable result with Figure 5 and 6. It is because the resource consumption of live migration process is only related to the source system resources. However, the resource reservation in the target machine is indispensable in the live migration of multiple virtual machines since the unaware of remaining resources in the target machine may incur the migration failure, especially when the target machine has limited resources which cannot support the migrated virtual machines.

E. Parallel Live Migration of Multiple Virtual Machines

Live migration of multiple virtual machines, including parallel migration, is very common in today's cloud data center. In this subsection, we investigate the parallel live migration performance of multiple virtual machines. Figure 8 shows the total migration time when migrating idle virtual machines from a 4-core machine to 8-core machine. The *Real Time* means the actual measured migration time which is an average migration time of all the parallel migration virtual machines. And the *Calculated Time* means the calculated migration time which is calculated based on the migration time of migrating one virtual machine. From the figure, we see that the real migration time is longer than the calculated time, and the gap becomes larger as the

Table III
WORKLOAD-AWARE LIVE MIGRATION STRATEGIES PERFORMANCE COMPARISON OF SPECJBB, IOZONE, SYSBENCH, AND WEBBENCH WHEN MIGRATING TO EACH OTHER VIRTUAL MACHINES.

	Baseline	**Mig. to Jbb VM**	**Mig. to File VM**	**Mig. to Database VM**	**Mig. to Web VM**
SPECjbb (bops)	14849	11996	13770	14069	12820
IOzone (Kbytes/sec)	414560	243604	109350	142072	213602
Sysbench (s)	88.10	110.96	126.86	97.04	117.32
Webbench (Pages/min)	1994840	1565262	1677740	1807098	1115668

VM number of parallel migration increases. Because the 4-core machine owns fewer system resources than 8-core machine, when the virtual machine migrated from 4-core machine to 8-core machine, the limited resources of 4-core machine cannot support the parallel migration process well. When the parallel number is small, the resource bottleneck is not reflected on the migration time. As the parallel number increases, there is a resource bottleneck in the source machine affecting the parallel migration number. This conclusion can be demonstrated again in Figure 9, in which the virtual machines are migrated from 8-core machine to 4-core machine in parallel. Due to the adequate system resources in 8-core source machine, the parallel live migration performs better and achieves shorter migration time than calculated time. One interesting result can be found by comparing the two figures that the migration time is shorter when migrating from 8-core machine to 4-core machine. For example the migration time of 8VMs parallel migrating from 8-core to 4-core machine is 78.73s, while migrating from 4-core to 8-core is 117.18s. From this experiment, we can conclude that if the source machine owns adequate system resources, more parallel number can be performed to improve the live migration efficiency of multiple virtual machines, otherwise the performance of parallel migration can become even worse than the sequential migration.

F. Workload-Aware Live Migration Strategies

In traditional migration scenario, if the target machine owns enough system resources to support migrated virtual machines, then the virtual machine can be migrated to this target machine. This is a simple and convenient migration strategy that doesn't consider the existing workload types running in the target machine. If we take into account the workload characteristics, more efficient migration strategy can be made. In this subsection, we will investigate the effects of virtual machine workload performance when migrating to consolidate with different virtual machine workloads.

Table III compares the performance of SPECjbb, IOzone, Sysbench, and Webbench under different live migration strategies. The baseline mode means the performance result of workloads when migrating to idle target machine that holds no virtual machine. Obviously, migrating to consolidate with other workloads will lead to the performance loss with varying degrees due to the competition for shared resources. From the table, we find all the workloads achieve the worst performance when migrating to the same workloads except Sysbench. It is not suitable to migrate to the same workloads in a single platform because of the contention of the same kind resource. For example, the SPECjbb is CPU intensive, when migrating to another SPECjbb virtual machine, there will be a heavy pressure of CPU resource, while leaving other system resources with a low utilization like network bandwidth resource. Sysbench gets a good performance because the database workload doesn't consume only one kind of system resources. From this experiment, we also can find that which two kinds of workloads can co-exist well, and which two kinds of workloads are mutually exclusive. We see the SPECjbb and Sysbench workloads are most friendly with each other, leading to least performance loss, while IOzone and Sysbench workloads are least suitable to consolidate together. It is because that the SPECjbb consumes a lot of CPU resource while Sysbench consumes a lot of I/O resource and also consumes a small amount of CPU, migrating the two virtual machine workloads together will make the best use of the different aspects of system resources. On other hand, IOzone and Sysbench are both I/O intensive, when migrating the two virtual machine workloads together, I/O becomes the main performance bottleneck. Besides, we find the IOzone is the most vulnerable to any other workload due to its heavy and slow I/O operations. So it's better to migrate the IOzone virtual machine workload in a stand-alone server without affecting the performance of other virtual machine workloads.

G. Experimental Analysis and Optimization Methods

From the above experiments, we have found some interesting results, based on which some optimization methods are proposed to improve the live migration efficiency of multiple virtual machines.

The experimental results show that: (1) Live migration of virtual machine workloads bring some performance overheads due to the unavailable service in the downtime and the resource competition between virtual machine workloads and the migration process; (2) The performance overheads of live migration is affected by memory size, CPU resource, and the workload types. The large-scale complex workloads, specially the memory-intensive workloads can result in long downtime and total migration time. Large memory will lead to a long migration time. The adequate CPU resource in source machine will help improve the migration efficiency; (3) Although the resource reservation in target machine will not help improve the migration efficiency, it is necessary to avoid the migration failures due to the possibility of insufficient resources in the target machine; (4) The adequate system resources in the source machine can make more parallel number of migrations and can obtain better migration efficiency. Compared with sequential migration, the parallel migration acquires better efficiency when there are enough system resources in the source machine, otherwise the parallel migration will lead to even worse performance than sequential migration; (5) The performance of migrated virtual machine workloads can be affected by the workload characteristics of the existing virtual machines running in the target machine. When taking into account the existing virtual machine workload characteristics in the target machine, better workload performance can be obtained.

The optimizations can be achieved by the following methods (the improved efficiency is already reflected in the above experimental results): (1) *Optimization in the source machine.* One optimization method to improve the live migration efficiency can be implemented by dynamically adjusting CPU and memory resources in the source machine, for example allocating more CPU resource to the virtual machine being migrated. Another optimization can be achieved by adjusting the migration sequence that letting the virtual machines with small memory migrated first. By using this migration sequence optimization method, an overall shortest migration time can be achieved; (2) *Parallel migration of multiple virtual machines.* Parallel migration can obtain better performance than sequential migration if there are sufficient system resources in the source machine, otherwise the parallel migration performance can be even worse; (3) *Workload-aware migration strategy.* When making efficient migration decision (from which source machine to which target machine), it is better to migrate the virtual machine workloads according to the virtual machine workload characteristics running in the target machine.

IV. Related Work

Live migration of virtual machines is an important technology in modern cloud data center to achieve load balancing, online maintenance, and energy saving. Many efforts have been made to improve the live migration efficiency. Pre-copy [8, 9] technology is the prevailing live migration approach that has been implemented in Xen and VMware virtualization platform. The extended pre-copy algorithm is also used in wide area network environments [10, 16]. Besides, the post-copy technology [13], log-based migration technology [11], and memory compression based migration technology [12] are developed to improve the migration efficiency. Huang et al. [17] improved the efficiency of VM migration by using RDMA (Remote Direct Memory Access) as the protocol to transfer VM migration traffic.

In addition, many researchers have applied the live migration technology into different areas for particular purposes. Travostino et al. [18] integrated the live migration technology into the Grid to achieve the seamless management of virtual machine resources over MAN/WAN. Wood et al. [19] implemented a system named Sandpiper using the live migration technology to eliminate hotspots in data centers. The live migration technology is also used to integrate with server consolidation to achieve the energy-saving goal [20, 21]. However, all the above work focuses on the migration technology or algorithm itself, or applying this technology into specific areas which does not involve a comprehensive performance analysis of the live migration performance of multiple virtual machines. Zhao et al. [22] evaluated the suspend/rusume migration performance of virtual machines rather than the live migration.

Differ from the above work, we study the live migration strategy of multiple virtual machines from experimental perspective, and investigate the impact of resource reservation methods on the migration efficiency. Additionally, other migration strategies such as parallel migration and workload-aware migration are also evaluated.

V. Conclusion and Future Work

Live migration of virtual machines is an efficient technology used to implement energy saving and load balancing in virtualized cloud computing data center. In this paper, we study the live migration efficiency of multiple virtual machines from experimental perspective and investigate different resource reservation methods in the live migration process as well as other complex migration strategies such as parallel migration and workload-aware migration. Experimental results show that: (1) Live migration of virtual machine brings some performance overheads. (2) The performance overheads of live migration is affected by memory size, CPU resource, and the workload types. (3) Resource reservation in target machine is necessary to avoid the migration failures. (4) The adequate system resources in the source machine can make more parallel number of migrations and can obtain better migration efficiency. (5) The workload-aware migration strategy can efficiently improve the performance of migrated workload. Based on the experimental discoveries, three optimization methods, optimization in the source machine, parallel migration of multiple virtual machines, and workload-aware migration strategy, are proposed to improve the migration efficiency.

Future work will include designing and implementing intelligent live migration mechanism to improve the live migration efficiency in the multiple virtual machines scenario and studying the migration strategies as an optimization problem using mathematical modeling methods.

Acknowledgment

This work is funded by the National 973 Basic Research Program of China under grant NO.2007CB310900 and National Natural Science Foundation of China under grant NO. 60970125.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] R. Buyya, C. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *The 10th IEEE international conference on high performance computing and communications*. Ieee, 2008, pp. 5–13.

[3] C. Waldspurger, "Memory resource management in VMware ESX server," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, p. 194, 2002.

[4] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003, p. 177.

[5] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid-Volume 00*, 2009, pp. 124–131.

[6] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, pp. 14–22, 2009.

[7] K. Ye, X. Jiang, D. Ye, and D. Huang, "Two Optimization Mechanisms to Improve the Isolation Property of Server Consolidation in Virtualized Multi-core Server," in *Proceedings of 12th IEEE International Conference on High Performance Computing and Communications*, 2010, pp. 281–288.

[8] C. Clark, K. Fraser, S. Hand, J. 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*, 2005, p. 286.

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

[10] 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*, 2008, pp. 99–106.

[11] 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*, 2009, pp. 101–110.

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

[13] 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*, 2009, pp. 51–60.

[14] K. Ye, J. Che, X. Jiang, J. Chen, and X. Li, "vTestkit: A Performance Benchmarking Framework for Virtualization Environments," in *Proceedings of fifth ChinaGrid Annual Conference*, 2010, pp. 130–136.

[15] 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 ACM/SPEC international conference on Performance engineering*, 2011, pp. 307–316.

[16] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schi "oberg, "Live wide-area migration of virtual machines including local persistent state," in *Proceedings of the 3rd international conference on Virtual execution environments*, 2007, pp. 169–179.

[17] W. Huang, Q. Gao, J. Liu, and D. Panda, "High performance virtual machine migration with RDMA over modern interconnects," in *Proceedings of the IEEE International Conference on Cluster Computing*, 2008, pp. 11–20.

[18] F. Travostino, P. Daspit, L. Gommans, C. Jog, C. De Laat, J. Mambretti, I. Monga, B. Van Oudenaarde, and S. Raghunath, "Seamless live migration of virtual machines over the MAN/WAN," *Future Generation Computer Systems*, vol. 22, no. 8, pp. 901–907, 2006.

[19] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and gray-box strategies for virtual machine migration," in *Proc. Networked Systems Design and Implementation*, 2007.

[20] K. Ye, D. Huang, X. Jiang, H. Chen, and S. Wu, "Virtual Machine Based Energy-Efficient Data Center Architecture for Cloud Computing: A Performance Perspective," in *Proceedings of 2010 IEEE/ACM International Conference on Green Computing and Communications*, 2010, pp. 171–178.

[21] X. Liao, L. Hu, and H. Jin, "Energy optimization schemes in cluster with virtual machines," *Cluster Computing*, vol. 13, no. 2, pp. 113–126, 2010.

[22] M. Zhao and R. Figueiredo, "Experimental study of virtual machine migration in support of reservation of cluster resources," in *Proceedings of the 3rd international workshop on Virtualization technology in distributed computing*, 2007, p. 5.