



HAL
open science

Evaluate the Performance and Scalability of Image Deployment in Virtual Data Center

Kejiang Ye, Xiaohong Jiang, Qinming He, Xing Li, Jianhai Chen

► **To cite this version:**

Kejiang Ye, Xiaohong Jiang, Qinming He, Xing Li, Jianhai Chen. Evaluate the Performance and Scalability of Image Deployment in Virtual Data Center. IFIP International Conference on Network and Parallel Computing (NPC), Sep 2010, Zhengzhou, China. pp.390-401, 10.1007/978-3-642-15672-4_32 . hal-01054970

HAL Id: hal-01054970

<https://inria.hal.science/hal-01054970v1>

Submitted on 11 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Evaluate the Performance and Scalability of Image Deployment in Virtual Data Center*

Kejiang Ye, Xiaohong Jiang, Qinming He, Xing Li, Jianhai Chen

College of Computer Science, Zhejiang University,
Zheda Rd. 38, Hangzhou 310027, China
{yekejiang, jiangxh, hqm, lix, chenjh919}@zju.edu.cn

Abstract. Virtualization technology plays an important role in modern data center, as it creates an opportunity to improve resource utilization, reduce energy costs, and ease server management. However, virtual machine deployment issues arise when allocating virtual machines into single or multiple physical servers. In this paper, we explore the performance and scalability issues for virtual machine deployment in a virtualized data center. We first evaluate the image scalability when allocating multiple VMs per physical server using four typical servers in data center. Then we investigate how the overall efficiency will be affected when deploying M virtual machines into N physical machines with different deployment strategies. Experimental results show that: (i) There is a resource bottleneck when deploying single type virtual machine server into single physical server, except for composite workloads. (ii) More physical machines do not always benefit for some specific applications to support a fixed number of virtual machines. (iii) MPI and network communication overheads affect the deployment efficiency seriously.

Key words: Virtual Machine, Scalability, Image Deployment, Server Consolidation

1 Introduction

Virtualization as a critical technology in modern data center has been receiving much attention as it creates an approach to improve resource utilization, reduce costs, and ease server management. The emergence of mature virtualization solutions such as VMware [1], Xen [2], KVM [3], and OpenVZ [4] have spurred this growth. Virtualization provides an abstraction of hardware resources enabling multiple instantiations of operating systems (OS) to run simultaneously on a single physical machine. By sharing the underlying physical resources, virtualization could achieve significantly higher system utilization compared to traditional physical machines.

* 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.

One of the challenges in realizing such resource sharing is how to efficiently deploy virtual machines into physical server computers. Specifically, two questions should be answered in the deployment process: (i) How many virtual machines can be allocated into one physical machine while still maintaining good performance? (ii) If M virtual machines are required to be deployed, how many physical machines are needed to support them?

Since the introduction of virtual machine monitor(VMM), the performance overheads introduced by virtualization is widely evaluated. Many researchers put their focus on the performance overheads of several pivotal system components in virtualization environments, e.g., CPU, memory, disk and network etc [5, 6]. Some researchers focus the performance overheads in specific scenarios, such as server consolidation [7, 8] and HPC environment [9]. However, to the best of our knowledge, none of them have quantified the deployment efficiency for typical server applications and HPC applications under different deployment strategies.

To evaluate the performance of different deployment strategies, we design two sets of experiments. We first investigate the deployment efficiency of M-1 scenario that allocating M virtual machines into one physical machine. We will investigate the best M selection. We choose four representative server applications in modern data center and quantify the server scalability. Then we evaluate the deployment efficiency of M-N scenario that allocating M virtual machines into N physical machines. To quantify the overall performance, we choose a standard HPC benchmark HPCC that running in parallel manner with MPI communication.

Our experimental results indicate that there is a resource bottleneck when deploying single type virtual machine server into single physical machine and consolidation of different workloads together is recommended. It is suitable to deploy several virtual machines running composite workloads into one physical machine. For computing intensive applications that don't run in parallel manner and processor communication intensive applications which are insensitive with the deployment strategies, adding more physical machines is useless. For the parallel applications with no MPI communication, more physical machines can achieve better performance. While for the computing intensive workloads that have high MPI and network communication traffic, it is not suitable to deploy the virtual machines across multiple physical machines due to the significant communication overheads between physical machines.

The rest of this paper is organized as follows. We describe the background of Xen virtualization technology and our experiment motivation in Section 2. And then in Section 3, we present our experimental methodology to solve the problems presented in Section 2. In Section 4, we discuss and analyze the experimental results. The related work is introduced in Section 5 and we conclude with the future work in Section 6.

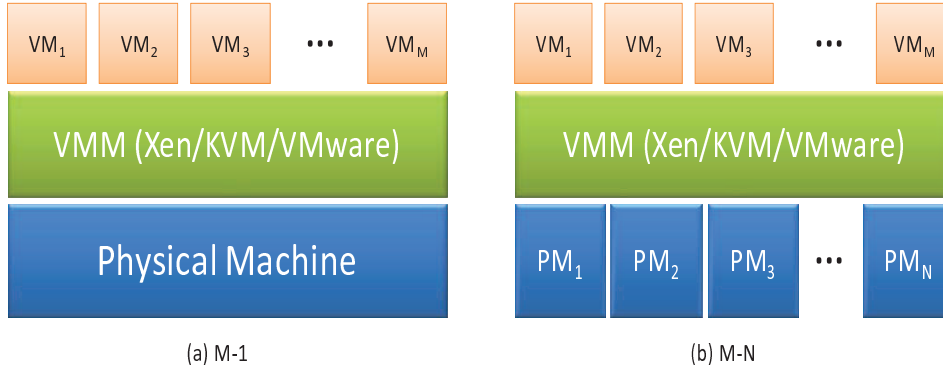


Fig. 1. Two experimental scenarios: (a) M-1 means allocating M virtual machines into one physical machine; (b) M-N means allocating M virtual machines into N physical machines.

2 Background and Motivation

Previous studies have shown that the resource utilization is very low in data center most of the time [10]. We also realize that the maximum resource utilization is much higher than the average one. Virtualization can be used to solve the utilization problem as it holds the benefits of running multiple virtual machine instances simultaneously while sharing the underlying physical resources. However, there is a lack of proposal for selecting the image deployment strategies. It is essential to investigate the most appropriate deployment strategy to achieve best resource utilization while maintaining good QoS. If we allocate too many or too few virtual machines into the physical machines, the physical machines may either be grossly under-utilized or overloaded leading poor application-level QoS.

Fig. 1 indicates our two experimental scenarios. In the M-1 scenario, we measure the scalability that adding the virtual machine into the physical machine one by one to investigate the appropriate virtual machine number hosting in one physical machine. While in the M-N scenario, we will evaluate the efficiency that allocating M virtual machines into N physical machines and investigate the best N .

2.1 Xen Virtual Machine Monitor

Xen [2] is a popular open-source x86 virtual machine monitor (VMM) that allows multiple instantiation operating systems (OSs) running concurrently on a single physical machine. Xen supports both full virtualization and para-virtualization scheme. In the para-virtualization scheme, the guests run a modified operating system using a special hypercall. Para-virtualization avoids frequent traps into the hypervisor, and achieves high performance. Starting from version 3.0, Xen implement the full virtualization through hardware-assisted virtualization (HVM)

technology. Both Intel and AMD have developed their own solutions named Intel VT and AMD-V to support Xen. Xen uses this hardware-assisted technology to support unmodified guest operating systems to run within virtual machines.

2.2 Requirements for M-1 Performance Evaluation

Generally, one physical machine will support several virtual machines running simultaneously to share the same physical resources in modern data center. However deploying too many virtual machines will cause the degradation of application QoS. To deploy appropriate number of virtual machines into the physical platform is a challenge. Besides, different workloads have different demands on system resources. For example, web server consumes a lot of network bandwidth and CPU resource while file server consumes large amounts of disk I/O resource but less CPU time. It is necessary to study the representative server workloads used in data center.

2.3 Requirements for M-N Performance Evaluation

The benefits of virtualization such as flexible resource management, high reliability, live migration play an important role in the high performance computing (HPC) and parallel computing community. However there is a virtual machine deployment issue that arranging appropriate physical machines to support above virtual machine workloads, which is a challenge. Arranging all the virtual machines into one physical machine will reduce the communication overhead between the virtual machines but there is a resource bottleneck which will seriously affect the application performance. In the otherwise, M virtual machines arranged into M physical machines can obtain the most abundant system resources that each physical machine hosts only one virtual machine, but the communication overheads between physical machines are very huge. What's more, there is a huge loss of electricity energy. For example, if one parallel job needs 16 threads running simultaneously, we will create 16 virtual machines to support it with MPI communication between each virtual machine. We need to find a optimal deployment that maximizes the use of system resource while maintaining overall good performance. Is one physical machine enough or more physical machines are needed to support the 16 virtual machines, we will investigate this deployment issue below.

3 Experimental Methodology

3.1 Experimental Configuration

The M-1 experimental evaluations are performed on the Dell 2900 PowerEdge server, with 2 Quad-core 64-bit Xeon processors at 1.86 GHz, while M-N experimental evaluations are performed on four Dell OPTIPLEX 755, with Intel Core2 Quad CPU at 2.4GHz. We use Ubuntu 8.10 with kernel version 2.6.27 in

domain 0, and the version of Xen hypervisor is the 3.3.1. Each virtual machine is installed with Ubuntu 8.10 as the guest OS with 4 VCPUs and 256MB memory size. We choose MPICH 2.1.0.8 as our MPI environment.

3.2 Workloads and Benchmarks

To investigate the deployment efficiency in M-1 and M-N scenarios, we design two sets of experiments. In the M-1 scenario, we measure the scalability that answers how many virtual machines can be allocated into one physical machine while maintaining good performance?

We choose four typical server workloads in modern data center running in virtual machine in our experiments. They are java server, file server, database server, and web server. Java performance is important in many modern multi-tiered applications. We choose SPECjbb2005 [11] as the benchmark for java server evaluation. File server performance is also a typical metric indicating the ability to serve I/O requests from clients. We use IOzone benchmark [12] to measure the file system I/O performance. Database is needed to support running transactional workloads in many modern applications, and it is resource intensive and exercises most system components, especially CPU and storage. We choose Sysbench [13] as our database benchmark. The back-end database used in the experiment is Mysql. Web server is pervasive in modern data centers and is a representative workload for consolidation. We use the WebBench, a simple tool for benchmarking WWW or proxy servers. Apache is used as the back-end server.

In the M-N scenario, we try to answer the question how many physical machines are needed to support M virtual machines? We use the HPC benchmark suite [14] for our study that is commonly used for HPC measurements. The HPC suite is a comprehensive set of synthetic benchmarks designed to profile the performance of several aspects of a cluster. The testing applications used in our study are listed as follows:

- HPL: measures the floating point rate of execution for solving a linear system of equations.
- DGEMM: measures the floating point rate of execution of double precision real matrix-matrix multiplication.
- FFT: measures the floating point rate of execution of double precision complex one-dimensional Discrete Fourier Transform (DDF).
- PTRANS: measures the transfer rate for large arrays of data from multiprocessor’s memory.
- STREAM: measures the sustainable memory bandwidth (in GB/s) and the corresponding computing rate for simple vector kernel.
- RandomAccess: measures the rate of integer random updates of memory (GUPS).
- Latency & Bandwidth: measures latency and bandwidth of a number of simultaneous communication patterns.

There are three running modes: **single** means that a single processor runs the benchmark, **star** means all the processors run separate independent copies

of the benchmark with no communication, **mpi** means all processing elements running the benchmark in parallel using explicit data communications.

In our experiments, three problem sizes were evaluated, they are 1000MB, 2000MB, 3000MB. The block and grid sizes used are common-block: 80, 100, 120; grid: 2*2, 1*4, 4*1.

4 Experimental Results and Analysis

In this section, we describe the measurement results of deployment performance and scalability in M-1 and M-N scenarios respectively. In order to ensure the data precision, each of the showed experiment results was 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.

4.1 Deployment in M-1 Scenario

In this experiment, we investigate the deployment performance that allocating M virtual machines into one physical machine. M scales from 1 to 8 here. Fig. 2 shows the deployment results of java server, file server, database server, and web server. It is obvious that more virtual machines deployed into one physical machine incurs more performance degradation, especially for file server and web server, since they are I/O intensive workloads and the serious I/O resource contention becomes a bottleneck.

Fig. 2(a) shows the deployment efficiency of java server with a 65.95% performance degradation when deploying 8 VMs into the physical machine compared to the case that deploying 1 VM into the physical machine. There is a nearly linear decline in the performance.

From Fig. 2(b), we find that there is a significant decline in file I/O performance. It is because that the file server is I/O intensive. When more virtual machines are added into the physical machine, the disk I/O resource becomes the performance bottleneck and seriously affects the performance. We note that when deploying more than 5 virtual machines into the physical machine, the decline trend becomes flat and maintains at a stable I/O bandwidth with about 30000 Kb/s. The reason is that when more than 5 file servers are deployed into the physical machine, not only the I/O bandwidth but also the other system resources like CPU become the performance bottleneck.

When in the database scalability testing (Fig. 2(c)), the physical machine can support 6 virtual machines well with the execution time less than 86 seconds. However, when more virtual machines are deployed into the physical machine there is a huge increase in the execution time due to the resource bottleneck like CPU, memory, and disk I/O.

The web server testing (Fig. 2(d)) shows the similar phenomenon with file server because of the similar system resource demands. The significant decline of web server performance is caused by intense competition for the network I/O and CPU resource.

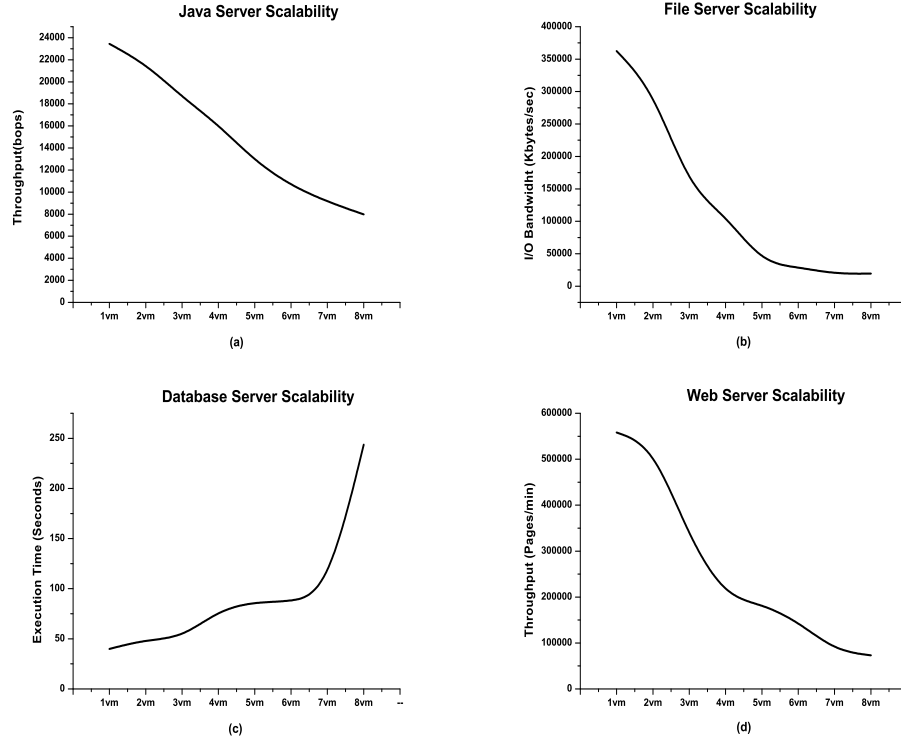


Fig. 2. The Deployment Performance and Scalability of Java Server, File Server, Database Server and Web Server in M-1 Scenario that Allocating M Virtual Machines into One Physical Machine (M-1 Scenario).

It is obvious from the above experiments, when the virtual machine number scales from 1 to 8, all the four server workloads have a sharp performance degradation due to heavy demand on the same kind resource which becomes the performance bottleneck. When more virtual machines are added, other system resources will become the performance bottleneck too. So in this M-1 scenario, M depends on user's QoS requirements. It is not suitable to deploy more than two same servers in the physical machine due to the QoS consideration. One alternative approach is to consolidate different workloads running simultaneously. What's more, we find an interesting phenomenon that one physical machine can host 6 database servers running simultaneously well based on our experimental results. It is because that database is a kind of composite workloads which consumes CPU, memory and I/O resource at the same time.

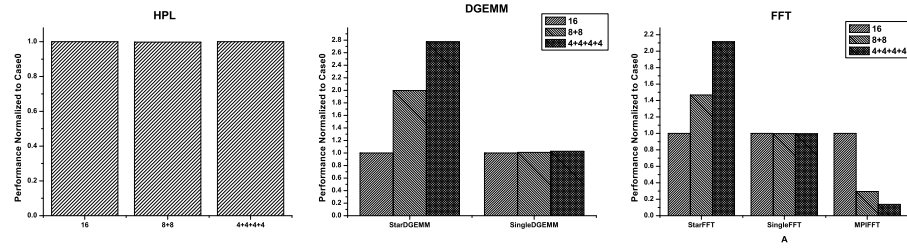


Fig. 3. The Computing Performance in the Case that Deploying M Virtual Machines into N Physical Machines (M-N Scenario).

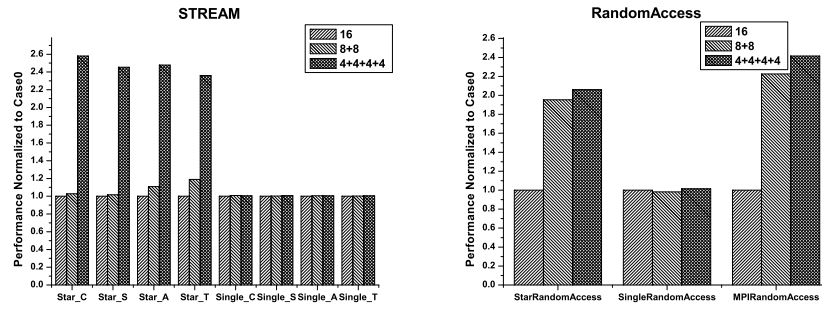


Fig. 4. The Memory Performance in the Case that Deploying M Virtual Machines into N Physical Machines (M-N Scenario).

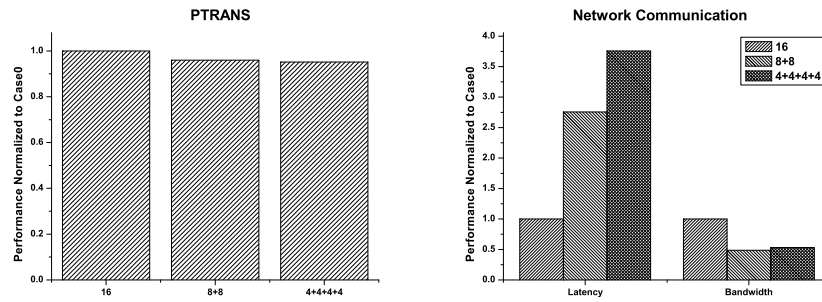


Fig. 5. The Communication Performance in the Case that Deploying M Virtual Machines into N Physical Machines (M-N Scenario).

4.2 Deployment in M-N Scenario

In this experiment, we will study the efficiency of deploying M virtual machines into N physical machines with HPCC benchmark. We create 16 virtual machines and test three different deployments: 1) allocate 16 VMs into one physical machine; 2) allocate 16 VMs into two physical machines with 8 VMs per physical machine; 3) allocate 16 VMs into four physical machines with 4 VMs per physical machine. We investigate the deployment efficiency from computing performance, memory performance and communication performance with diverse benchmarks. The results of these experiments are presented in Fig. 3 to 5.

Fig. 3 shows the computing performance with different deployment strategies in M-N scenario. HPL, DGEMM, and FFT measure the floating computing capability focusing different aspects. From Fig. 3 we note that the HPL performance has no apparent changes when 16 virtual machines are deployed into one, two or four physical machines. Obviously, this phenomenon indicates that more physical machines will not always improve the overall performance for some applications. In the **single** mode testing, DGEMM and FFT both have no significant changes in all the three deployment strategies. While in the **star** mode, the case that involving more physical machines achieves better performance. It is because in the **single** mode a single processor is used to run the benchmark, while in the **star** mode all the processors run separate independent copies of the benchmark with no communication which maximum utilize the physical resource. Furthermore, in the **mpi** mode testing of FFT, we observe the case that deploying 16 virtual machines into 4 physical machines obtains worse performance than the case that deploying 16 virtual machines into only one physical machine. The reason is that when involving more physical machines, the MPI communication overhead becomes a bottleneck and affects the performance.

Fig. 4 shows the result of memory performance with different deployment strategies using STREAM and RandomAccess benchmarks. Similar to the computing performance, in the **single** mode testing, both STREAM and RandomAccess have no significant changes in the three deployment cases. While in the **star** mode testing, the copy, scale, add, and triad performance of RandomAccess gain an performance improvement when more physical machines are used. RandomAccess has the similar result with STREAM in the **star** mode. While in the **mpi** mode, we find an opposite conclusion with FFT that the RandomAccess obtains better performance when using more physical machines. It is because that the RandomAccess performs much local processor communication that improves the performance while causes little network communication overhead.

Fig. 5 shows the communication performance including processor communication and network communication. PTRANS measures the communication performance of multiprocessor while Latency and Bandwidth refer to the network communication performance. From Fig. 5 we find that the data transfer rate is slightly affected by the deployment strategies due to the slightly local memory transfer overheads. While network latency and bandwidth are seriously affected by the deployment strategies with a 275.74% increase in latency and a 46.89% decrease in the bandwidth compared to the case that deploying 16 VMs

into one physical machine. This experiment demonstrates that there exists huge network communication overhead when deploying 16 virtual machines into more physical machines.

There is a compromise between communication overhead and system resources. For some specific server workloads such as HPL and PTRANS, more physical machines provide no obvious performance improvement. Generally, more computing resource will improve the running efficiency for the parallel workload running in parallel without MPI communication. Furthermore, for the communication intensive workloads that perform much MPI communication or network communication, it is better to deploy the virtual machines into fewer physical machines due to the heavy communication overheads between physical machines.

5 Related Work

A lot of research has been done to evaluate the performance in virtualization environments [2, 5–7, 15, 16]. Barham et al. [2] gave a comprehensive introduction to the Xen hypervisor and made a thorough performance evaluation of Xen against VMware ESX Server, UML, Connectix’s Virtual PC and Plex86 with SPEC CPU2000, OSDB, dbench and SPEC Web2005. Clark et al. [15] reproduced the results from [2] with almost identical hardware, and compared Xen with native Linux on a less powerful PC. Padala et al. [7] measured the performance of server consolidation. Che et al. [6] studied an initial comparison of Xen and KVM with Linpack, LMbench and IOzone. Recently, Deshane [5] presented an independent work about performance comparison of Xen and KVM at Xen Summit, which measured the overall performance, performance isolation and scalability of Xen and KVM. Hai Jin et al. [16] presented a VSCBenchmark to evaluate the dynamic performance of virtualization systems in server consolidation. However, all of the above work does not consider the deployment issues.

With respect to the resource and image management issues in virtualization environments, Garbacki et al. [17] solved the virtual resource management issue by introducing linear programming approach. Yamasaki et al. [18] presented a model-based resource selection policy to achieve fast virtual cluster installation on heterogeneous grid environment. Tan et al. [19] indicated that the choice of storage solution and access protocol would affect the deployment efficiency. Some management system and tools were also developed for flexible deployment and management of virtual machines [10, 20, 21]. Our work differ from the previous work, focusing the evaluation the performance and scalability of image deployment with different deployment strategies.

6 Conclusion and Future Work

In this paper, we have investigated the performance and scalability with different virtual machine deployment strategies. We summarize two typical virtual machine deployment scenarios M-1 and M-N to explore the deployment issue.

We first investigate the deployment efficiency of M-1 scenario that allocating M virtual machines into one physical machine. We choose four representative server applications in modern data center and quantify the scalability. Then we evaluate the efficiency of M-N scenario that allocating M virtual machines into N physical machines. To quantify the overall performance, we choose the standard HPC benchmark HPCC that running in parallel manner.

Our experimental results indicate: 1) There is a resource bottleneck when deploying single type virtual machine server into single physical machine and consolidation of different workloads together is recommended. It is suitable to deploy several virtual machines running composite workloads into one physical machine, such as database server which consumes CPU, memory and I/O resources at the same time. 2) For computing intensive applications that don't run in parallel manner and processor communication intensive applications which are insensitive with the deployment strategies, more physical machines are useless. 3) For the parallel applications with no MPI communication, more physical machines can achieve better performance. 4) For the computing intensive workloads that have high MPI and network communication traffic, it is not suitable to deploy the virtual machines across multiple physical machines due to the significant communication overheads between physical machines.

This is only the beginning of performance evaluation of deployment issues. Various other measurement and optimization algorithm will need to be explored in the future.

References

1. Waldspurger, C.A.: Memory Resource Management in VMware Esx Server. SIGOPS Oper. Syst. Rev. 36(SI), 181–194 (2002)
2. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the Art of Virtualization. SIGOPS Oper. Syst. Rev. 37(5), 164–177 (2003)
3. Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A.: kvm: the Linux virtual machine monitor. In: Linux Symposium. (2007)
4. OpenVZ: Server Virtualization Open Source Project, <http://openvz.org>
5. Deshane, T., Shepherd, Z., Matthews, J., Ben-Yehuda, M., Shah, A., Rao, B.: Quantitative comparison of Xen and KVM. Xen Summit, Boston, MA, USA (2008)
6. Che, J., He, Q., Gao, Q., Huang, D.: Performance Measuring and Comparing of Virtual Machine Monitors. In: IEEE/IFIP International Conference on Embedded and Ubiquitous Computing. (2008)
7. Padala, P., Zhu, X., Wang, Z., Singhal, S., Shin, K.: Performance evaluation of virtualization technologies for server consolidation. Technical Report, HP Laboratories (2007)
8. Apparao, P., Iyer, R., Zhang, X., Newell, D., Adelmeyer, T.: Characterization & analysis of a server consolidation benchmark. In: Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, 21–30 (2008)
9. Tikotekar, A., Vallée, G., Naughton, T., Ong, H., Engelmann, C., Scott, S.L.: An analysis of hpc benchmarks in virtual machine environments. In: Proceedings of

- 3rd Workshop on Virtualization in High-Performance Cluster and Grid Computing, 63–71 (2008)
10. Padala, P., Shin, K., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., Salem, K.: Adaptive control of virtualized resources in utility computing environments. *SIGOPS Oper. Syst. Rev.* 41(3) (2007)
 11. Standard Performance Evaluation Corporation, SPECjbb, <http://www.spec.org/jbb2005>
 12. Iozone Filesystem Benchmark, <http://www.iozone.org>
 13. Sysbench Benchmarking Tool, <http://sysbench.sourceforge.net>
 14. HPC Challenge Benchmark, <http://icl.cs.utk.edu/hpcc>
 15. Clark, B., Deshane, T., Dow, E., Evanchik, S., Finlayson, M., Herne, J., Matthews, J.: Xen and the art of repeated research. In: *USENIX annual Technical Conference*, 135–144 (2004)
 16. Jin, H., Cao, W., Yuan, P., Xie, X.: VSCBenchmark: benchmark for dynamic server performance of virtualization technology. In: *Proceedings of the 1st international forum on Next-generation multicore/manycore technologies* (2008)
 17. Garbacki, P., Naik, V.: Efficient resource virtualization and sharing strategies for heterogeneous Grid environments. In: *10th IFIP/IEEE International Symposium on Integrated Network Management*, 40–49 (2007)
 18. Yamasaki, S., Maruyama, N., Matsuoka, S.: Model-based resource selection for efficient virtual cluster deployment. In: *Proceedings of the 3rd international workshop on Virtualization technology in distributed computing* (2007)
 19. Tan, T., Simmonds, R., Arlt, B., Arlitt, M., Walker, B.: Image management in a virtualized data center. *ACM SIGMETRICS Performance Evaluation Review* 36(2), 4–9 (2008)
 20. Vallée, G., Naughton, T., Scott, S.: System management software for virtual environments. In: *Proceedings of the 4th international conference on Computing frontiers*, ACM (2007)
 21. McNett, M., Gupta, D., Vahdat, A., Voelker, G.: Usher: An extensible framework for managing clusters of virtual machines. In: *Proceedings of the 21st Large Installation System Administration Conference* (2007)