Regarding the challenges of Performance Analysis of Virtualized Systems

Steven Harris, sharris22@wustl.edu (A paper written under the guidance of Prof. Raj Jain)

Download

Abstract

Virtualization is an irreplaceable technique for efficient management of hardware resources in the field of information technology and computer science. With this phenomenal technology come challenges that hinder proper performance analysis of virtualized systems. This study focuses on virtualization implementations, various enterprise solutions, characterization of observations, concerns, pitfalls, and industry standard virtualization performance measurement tools. Our goal is to delineate the various intricacies of virtualization to help researchers, information technologists, and hobbyists realize areas of concern and improvement in their performance analysis of virtualized systems.

Keyword: Hardware Virtualization, Virtual Machine Monitor, Hypervisor, Performance Analysis, Benchmark, XEN, KVM, Wmware, Virtual Box, AMD-V, Intel VT, SPEC

Table of Contents:

1. Introduction

The transformative cost savings, operational efficiently, resiliency, and resource aggregation introduced by virtualization technologies in the enterprise environment has allowed these solutions to become ubiquitous across the information technology spectrum. As performance increases exponentially in computer hardware, particularly Central Processing Units (CPU), Random Access Memory (RAM), and Storage Area Networks (SAN), we see these devices becoming less cost prohibitive to a larger range of users.[1] With the adoption of this technology fueling new improvements to multicore processors, leading computer processor manufactures, such as Intel and AMD, have introduced their own unique extensions to processor architecture to assist with the demands for hardware virtualization.[2] In this paper, we will survey the topology of computer system virtualization. With the plethora of virtualization technology currently employed in the field, there is much variability in the solutions employed by users and organization. The understanding of the distinctions in virtualization implementations is paramount to proper modeling and performance analysis of these solutions. In light of this, in the section 2 we will delineate some of the most popular virtualization implementations and their differences, in section 3 we will discuss how the solutions are employed. In section 4 we will look at some of the challenges faced in performance analysis of virtualized computer systems and discuss some of the issues caused by the host/guest paradigm,interdependence, and utilization ambiguity. We will conclude our survey by looking at some of the current measurement tools in section 5 used to categorize and classify observations of the systems as they relate to performance analysis.

1.1Virtualization Overview

Virtualization is a means to stimulate some effect, condition, or response of a computer, network device, or program. More specifically, virtualization is a technique used to efficiently utilize systems resources.[2] For the sake of our discussion, we will be limiting our scope to the virtualization of Computer System. In this scope, virtualization implementations may be divided into four categories: Full Virtualization, Partial Virtualization, Paravirtualization, and Hardware virtualization.[4, 5] These virtualization paradigms are implemented by a

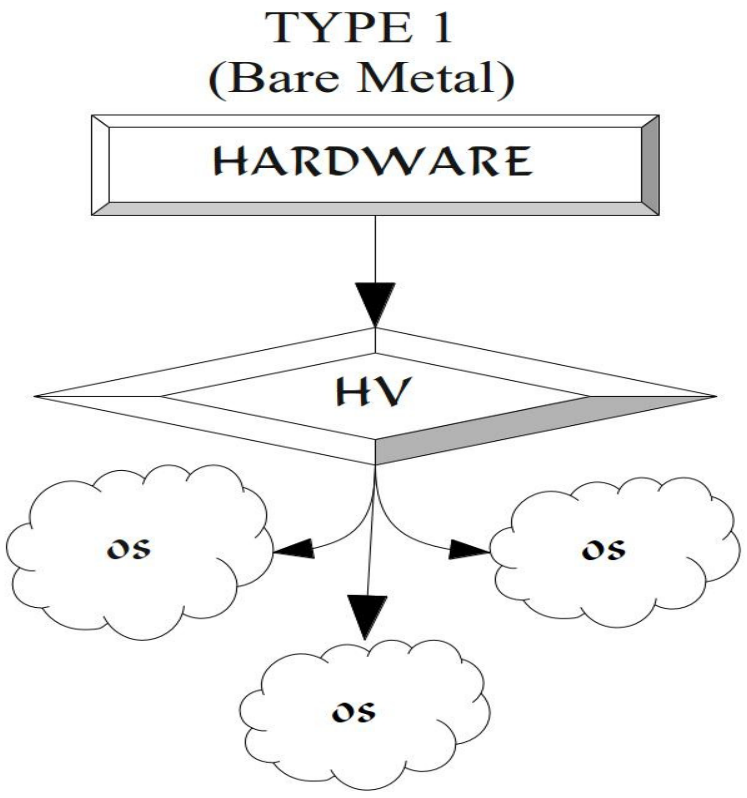

Figure 1: Type 1 Hypervisor

Virtual Machine Monitor (VMM), which is also known as a Hypervisor.[3] The Hypervisor is a software layer that provides Virtual Machine abstractions to the guest systems and is responsible for managing the execution of guest operating systems in the single physical host device.[2] In some instances, as we will see Section 2, the hypervisor abstraction presentation mimics physical hardware, which enables the guest operating systems and related applications to execute without modifying the guest operating system.[3] The Host is a term often used interchangeably for the hypervisor or the physical device on which it runs. Each instantiation of a virtual machine (VM) is defined as a guest that runs on the abstractions presented by the hypervisor. With respect to the hypervisor, there are two categories based on their hierarchal execution level in the computer system. [2]. Those which execute above the operating system stack are known as Type-1 hypervisors as shown in Figure 1. Those which run below the Operating System stack are known as Type-2 hypervisors as shown in Figure 2. The hypervisors provides virtual containers for each operating system to execute within. These containers are more commonly referred to as virtual machines. For all practical purposes, they are simply files containing operating system specific information that are executed in the context of the hypervisor.

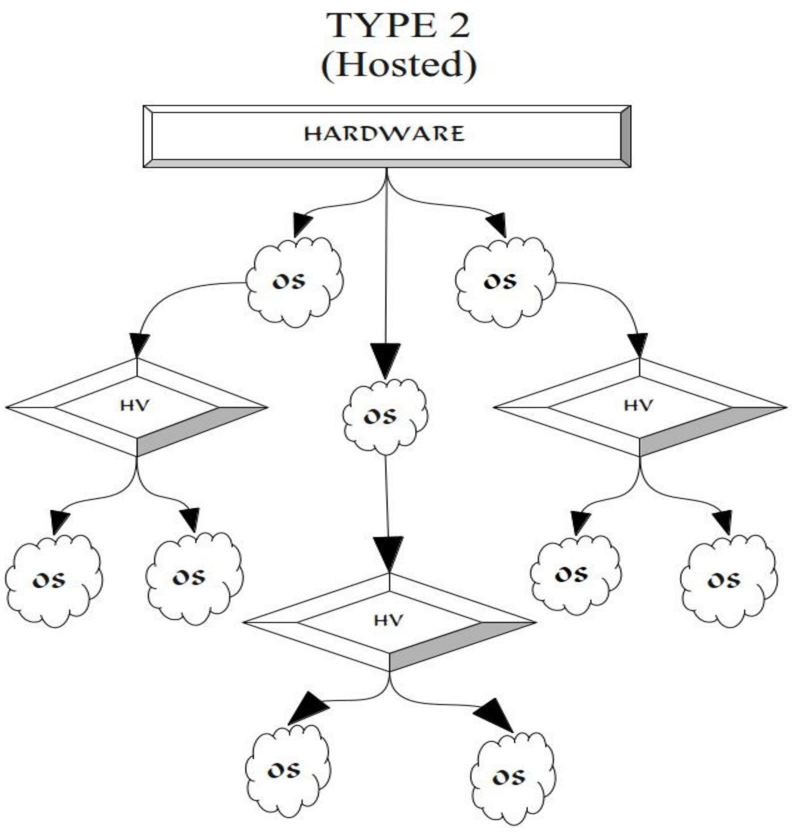

Figure 2: Type 2 Hypervisor

2.Virtualization Implementation

Depending on the requirements of the environment, several techniques can be implemented to suit the specification requirements. Depending on the complexity of the topology, one or more techniques may be required to accomplish the task. We will look into the three most common techniques. Although the implementations are divided into specific categories, the hypervisor implementations are not always completely diametrically polarized as some implementations may be created as operating system modules which blur the boundaries between type-1 and type-2 hypervisors.[6]

2.1Full Virtualization

This implementation of virtualization is often referred to as a Type 1 (Bare-metal) Hypervisor as it executes above the operating system layer. This approach allows implementers to run an unmodified guest operating system, thereby maintaining the existing resources and limiting disruptions due to migration or upgrading. This virtualization technique presents a complete virtual hardware platform seamless to the guest operating system based on the specification of the implementer. One advantage of full virtualization is that the physical resources of the system can be partitioned uniquely based upon the requirements of the guest operating system without any modifications to the guest. Certain operating systems such as Linux or BSD allow the modification of the kernel so this is not much of a concern in those environments. However, it is not possible to modify closed sourced operating systems such as Microsoft Windows. Since Microsoft windows is the prevalent technology in most network topologies, unmodified guest machines implementations are critical. [7] Furthermore, this technique is the most flexible of all virtualization types as most operating system will execute using this technique. Also of note is that some hypervisors employ binary translation to modify the operating system software in real time to replace instructions that penetrate the virtual machine with comparable sequences of instructions thereby giving the appearance of Full Virtualization. [8] Popular Type-2 Hypervisors include Vmware Server and Oracle VM VirtualBox.] Popular Type-1 Hypervisors include Citrix XenServer, VMware ESX/ESXi, Red Hat KVM, Xen, Oracle VM Server for SPARC, Oracle VM Server for x86 and Microsoft Hyper-V hypervisor.

2.2Partial Virtualization

Often referred to as address space virtualization, is a Type-2 (Hosted) Hypervisor. This technique attempts to partially simulate the required hardware environment for a specific guest machine operating system. Unfortunately, this technique is not robust enough to simulate the entire hardware platform required to run the complete operating system in a manner such as Full Virtualization. However, this technique maybe used to emulate enough of the underlying hardware to virtualize specific applications, which can then be used to share computer resources and services among computer users.

2.3Paravirtualization

Using a technique with employs features of both full and Partial virtualization, Paravirtualization attempts to encapsulate the virtual machine guest workloads from the operating system overhead. During this implementation, each guest operating system is modified to detect that it is executing in a virtual environment. The goal is to reduce the CPU time and resources spent performing operating system tasks that are complicated by the virtualization process by sending those requests to the hypervisor outside of the virtual guest for execution. The drawback in this case is that the guest operating system must be modified which is not feasible for closed source systems. This solution although generally implemented in Type-1 hypervisors may be used in type-2 Hypervisors as well. [9]

2.4Hardware Virtualization

This implementation takes the Type-1 full implementation a step further by enabling certain x86 processors such as AMD- V and Intel VT to employ virtualization within the internal x86 CPU architecture itself. This hardware assistance at the moment only supports CPU virtualization. However, later implementations will include Memory, Input-output (I/O), and network virtualization as well.[7]

3.Virtualization Solutions

The best solution is often the most robust solution. You will find in Table 1 a listing of the most current bare-metal virtualization solutions. There are many bare-metal virtualization solutions to fit the needs of most enterprise. However, the system that supports the most bare-metal host resources may not always be the best solution in terms of guest systems so it is important to pick the best solution for the environment. Furthermore, some implementations may be

	Vmware Vsphere	Microsoft Hyper-V	Citrix XenServer	Red Hat RHEV
HOST				
Max Logical CPU	160	320	160	160
Max Cores per CPU	Unlimited	Unlimited	Unlimited	Unlimited
Max RAM	2 TB	4 TB	1 TB	2 TB
GUEST				
Max Virtual CPU	64	64	16	160
Max Virtual Memory	1 TB	1 TB	128 TB	2 TB
Max Virtual Disk Size	2 TB	64 TB	2 TB	Unlimited

Table 1: Type 1 Hypervisors

based on cost more than performance. In those instances, it is important to review which services are required and the cost for that level of functionality. In most enterprise environments, the open-source solutions are not only cost effective but can be tailored to fit the most strenuous enterprise requirements.

4. Performance Analysis Challenges

For each virtualization design, virtualization implementations are generally bound to a particular virtualization solution to address some requirement in the environment. These virtualization configurations can have multiple factors that affect as well as asymptotically limit their performance. To further complicate this issue, in an environment with more than one virtual machine, resource contention among virtual machines essentially causes them to complete for the shared resources. It would be difficult to model a system where the variability in workload characteristics may be due to a number of factors such as the quantity of virtual machines, variations in host hardware, type of hypervisor, application versions, number of users, network congestion, storage configuration, and operating system differences. Many challenges in performance analysis of virtual environments are due to hypervisor implementation, guest operating system, guest resource interdependence, and utilization ambiguity. A correct perspective in terms of proper model construction may assist in performance analysis and modeling of virtual implementations and solutions. [10, 11]

4.1 Hypervisor Implementation

Although hypervisors are generally used to create, manages, and monitor virtual machines, not all hypervisors rely on a single node for this functionality. For instance, many researchers tend to use the Xen virtualization implementation as its open source nature allows easy modification and customization to specific project requirements as well as the flexibility to utilize either full or Paravirtualization. [11, 12, 13, 14, 15, 16] However, some implementations such as VMware may use more than one hypervisor for each requirement such as virtual machine partitioning, Infrastructure virtualization, system infrastructure services, management automation technologies, and virtual desktop infrastructure. Segmenting requirements based on the virtualization requirement can make it challenging to model performance when it may be based on the interdependence of systems.

4.2Guest Resource Interdependence

Performance analysis of virtualized guest can be described by the hardware abstraction platform that is presented to the guest operating system. Those being CPU, RAM, Storage, and Networking. In a non-virtualized system, these four factors can be isolated and analyzed based on certain constraints that take into account the workloads and interactions between the operating system and the factors. However, in a virtualized environment, there are more layers of complexity that results from certain resources increasing or decreasing performance that is dependent on the total utilization of the host system. That is, the competition for resource between guest systems may affect the performance of individual factors of the guest machine as they rely on the host system utilization.[17] In light of this, it is pivotal to take into account such interactions when attempting to model the performance of any guest machine factors or the analysis may be flawed.

4.3Guest Operating System

Although, the choice of hypervisor and guest resource interdependence may play a role in the performance analysis of the virtual machine, the results may be further compounded by the choice of guest operating system. In the enterprise environment, most servers are used to provide some service to clients. The services required by the clients may be Secure Shell (SSH), Secure File Transfer Protocol (sftp), Hyper Text transfer Protocol (HTTP), Radius authentication or countless other services. Depending on the operating system chosen for such services, there will be performance differences for each service that may vary by version, configuration, network connectivity, or client/server communication. Radius authentication for instance utilizes encryption that may have a different implementation and CPU utilization for Windows, UNIX, or Linux operating systems as well as additional overhead based on the configuration of the client authenticating. It is important to take operating system differences into account if performance and analysis models are made in a heterogeneous environment, which ultimately most wide area networks are in some sense.

4.4Utilization Ambiguity

Utilization ambiguity is often challenging to isolate as spikes in resources may be due to any number of factors. The most common are workloads which can be dependent on the number of applications running, number users, virtualization overhead, background processes, and configuration just to name a few. The forementioned issues may also lead to confounded results, as there may be interdependence between the resource utilizations or spawned application processes.
Failure to account for the variability in the utilization will often produce incorrect results in analysis and performance measurement.

4.5Proper Model Construction

Virtualization modeling is often a reflection of nonvirtual system modeling. In addition to coundfoundings, the hypervisor implementation, guest operating system, guest resource interdependence, and utilization ambiguity, it is important to realize that these factors may apply to more than one physical system that may further compound the analysis. In the field of virtualization, modeling it is important to divide the models into roughly three axes. Those being Single Guest on Single Host (SGSH), Multiple Guest on Single Host (MGSH), Multiple Guest across Multiple Hosts (MGMH).

Single Guest on Single Host: Observing performance under this scenario eliminates guest resource interdependence (resource contention). This method would be an ideal base line to classify the quantity of utilization consumed by a single VM given a particular workload or stress.

Multiple Guest on Single Host:: In this instance, guest resource interdependence may be a factor in the performance. The observations of performance based on stress or a particular workload will allow for the classification of resource consumption on the host system as well as the performance degradation of the individual VM's. Comparing and contrasting the results from SGSH in conjunction with this observation makes a compelling argument for the analysis of the utilization as well as performance.

Multiple Guest across Multiple Host: This is by far the most complicated analysis, as multiple hosts spanning multiple devices will add an additional layer of complexity given the intermediary system routes taken to connect the devices that may traverse switches, routers, wireless hubs, and Electromagnetic Field (EMF) interference. All of which can severely affect the overall interconnected performance of the systems. The conclusions found from this observation will be a product of the previous considerations contrasted with the observations seen across the network. The performance analysis of the individual systems will clarify some of the phenomenon observed across the network topology and make isolating issues more efficient.

5. Virtualization Measurement Tools

Many of the virtualization tools utilized for nonvirtual systems may be implemented in the performance testing of virtualized systems. However, the results of the performance analysis in some instances may be misleading as the results could either indicate the performance of the host system hypervisor to provided resources to the guest system as in CPU benchmarks or it may be due to the actual response time of the hosts physical hardware. In the latter case, if we looked at the read/write performance of a guest system on two host systems of identical hardware and configuration except for one host-utilizing standard Serial-Attached SCSI (SAS) drives versus Solid State Drives (SSD), we would find performance differences that go beyond the resource presentation of the host hypervisor. In terms of performance measurements, most enterprise level systems are utilized as servers that provide some service to clients. In light of this, we will be discussing measurement tools that reflect the specific genre of workloads.

5.1Benchmarks

There are a variety of virtualization benchmarks available specifically designed to compare and contrast virtualized systems [16]. Of the multitude of tests, several stand out as integral to server performance. Standard Performance Evaluation Corporation (SPEC) provides numerous benchmarks that are used for profiling systems. A few of the most important benchmarks are the SPECvirt_SC2010, CPU2006, SPECweb2005, SPECjAppServer2004, and SPECmail2008. This suite consists of workloads representative of industry standard survey reports on the demands in virtualization and server consolidation of production environments.

5.2 SPECvirt_SC2010:

SPECvirt_SC2010: A comprehensive standardized method for measuring the ability of a virtualized platform to manage server consolidation and comparing performance between environments by measurement of performance of the hardware, software, and application layers of the environment [18]. Instead of being a single benchmark, virt_Sc2010 is a combination of tests that builds on SPECweb, SPECjAppServer, and SPECmail2008.

5.3 SPECweb2005:

SPECweb2005: This workload is representative of infrastructure, file, and web servers. The workload is divided into two virtual machines consisting of four logical components: clients, primary client. Webserver and backend simulator. The clients send Hyper Text Transfer Protocol (HTTP) requests to the webserver and receive responses from the server. [18], the primary client administrates the benchmark process, with the webserver and backend simulator emulating the typical production environment.

5.4 SPECjAppServer2004:

The goal of this benchmark was to emulate the functionality of enterprise level java application usage by adhering to the following characteristics of: Industry- standardization, Simplicity, Neutrality, Openness, Complexity, Universality, Full distribution, Redundancy and Availability. Using these qualities, it stress the middle-tier rather than the end points of client or database server, maintains a comparable workload of real-world systems, captures intr-,extra-, and inter- enterprise processes, and factors in the quality of the system which it executes on.[19]

5.5 SPECmail2008:

SPECmail2008: This benchmark is designed to measure the performance and resiliency of enterprise level mail server workloads that range in excess of 1000+ users using standard SMTP and IMAP4 protocols. It also creates client workloads in excess for 40,000 users communicating using industry standard file types for office documents, rich media content, and encrypted communications implemented with SSL and TLS technologies.[19]

5.6 CPU2006:

CPU2006: This implementation is designed as a CPU- intensive suite of tools that focus on different types of computationally intensive performance metrics. The brunt of the workloads is directed at CPU intensive tasks usually caused by compilers, and CPU/RAM subsystem that are the primary resources in many server related functions. [19]

SPEC has a well-rounded set of tools for the performance analysis of any virtualized system. One of the drawbacks to precise measurement of performance is that many tools created by companies are vendor or Hypervisor specific. Therefore, there may be one set of tools that works well for Xen or KVM while not compatible with VMware or Hyper-V. However, SPEC benchmarks are compatible with most systems.

5.7 Network

In terms of network performance, there are tools that can measure the throughput of your devices as well as some configuration changes that can be made to slightly increase your performance. However, most tools in the network category are used to resolve network issues. In terms of performance increase, most performances gains will be made in upgraded hardware whether in network redundancy, clustering, or transitioning from cable to fiber interfaces, etc. Two open source tools to test the network virtualization performance are Netperf, Uperf developed by Sun Microsystems.[20, 21] CPU utilization is often time the most requested metric in terms of network performance. However, it can be challenging to measure this metric with great accuracy. Netperf and Uperf use several platform dependent measurement schemes to measure the CPU utilization. A closed source suite would also include httperf by Hewlett Packard.[22] These tools utilize network resources and provide workloads that would be comparable to those experienced in the enterprise environment.

5.8 Filesystems

Many performance related issues with the file system might be resolved by proper files system configuration. Similarly, to network performance, transitioning from Small Computer System Interfaces (SCSI)/Serial ATA (SATA) to Solid State Drives (SSD) has drastic performance increases over any configuration changes to the latter. There are a plethora of open-source tools used to test file system implementations in addition to may tools which built-in to specific file systems such as NTFS,EXT 3/4, and ZFS to name a few. Some of the tools worth mentioning are iozone, and bonnie++ which will take the file system through the paces and isolate performance bottlenecks within the file systems.[23, 24] These tend to be command line tools that eliminate much of the overhead that is often associated with graphical performance benchmarking tools.

6. Conclusion

As technology becomes more cost effective virtualization becomes more prevalent in all aspects of information technology. In an effort to improve the performance of virtualized guest systems as well as virtualization hosts, it is important to improve the precision and accuracy of performance measurement, analysis, and modeling. We have explored various types of virtualization implementations as well as virtualization solutions. We have also address some of the challenges which plague performance analysis and measurement of virtualization devices such as hypervisor implementation, guest resource interdependence, guest operating system selection, utilization ambiguity, and proper modeling concerns, in the hope that researchers will be more cognizant of the multi-faceted interactions and confoundings which occur in the virtualized environment. We have also touched on various virtualization measurement tools that will assist researchers in proper analysis of virtualized topologies. As virtualization research continues, it follows that the performance analysis will only become more complex and intertwined but with continued awareness of the interdependence of the systems on performance analysis, these systems can be properly modeled, simulated, and analysis.

Acronyms

- CPU - Central Processing Unit
- RAM - Random Access Memory
- SAN - Storage Area Networks
- VMM - Virtual Machine Monitor
- Bare-metal - Type-1 Hypervisor
- Hosted - Type-2 Hypervisor
- I/O - Input-output
- SSH - Secure Shell

- SFTP - Secure File Transfer Protocol
- HTML - Hyper Text transfer Protocol
- SGSH - Single Guest on Single Host
- MGSH - Multiple Guest on Single Host
- MGMH - Multiple Guest across Multiple Hosts
- EMF - Electromagnetic Field
- SAS - Serial-Attached SCSI
- SSD - Solid State Drives
- SPEC - Standard Performance Evaluation Corporation
- SCSI - Small Computer System Interfaces
- SATA - Serial ATA
- VM - virtual machine

References

1. Borkar, Shekhar, and Andrew A. Chien. "The future of Microprocessors."Communications of the ACM 54.5 (2011): 67-77.
2.]Kalady, Saidalavi, et al. "Implementation of a Purely Hardware- assistedVMM for x86 Architecture."Proceedings of the World Congress on Engineering. Vol. 1. 2009.
3. Rosenblum, Mendel; Garfinkel, Tal. "Virtual machine monitors: current technology and future trends," IEEE Computer, volume 38, issue 5, May, 2005.
4. Intel, "Intel Virtualization Technology," Intel Tech- nology Journal, Volume 10, Issue 3, August 2006.
5. POPEK, G. J., GOLDBERG, R. P., "Formal re-quirements for virtualizable third generation archi- tectures," ACM Communications,, July 1974
6. Graziano, Charles David. "A performance analysis of Xen and KVM hypervisors for hosting the Xen Worlds Project." (2011).
7. Vmware, "A Performance Comparison of Hypervisors" (2007)
8. Marshall, David. "Understanding Full Virtualization, Paravirtualization, and Hardware Assist." (2007).
9. http://virtualization-software.virtual-strategy.com/guide
10. J.P. Casazza, M. Greenfield, and K. Shi, Redefining Server Performance Characterization for Virtualization Benchmarking, Intel Technology Journal, 2006.
11. V. Makhija, B. Herndon, P. Smith, L. Roderick, E. Zamost, and J. Anderson, VMmark: A Scalable Benchmark for Virtualized Systems, Technical Report, VMware, 2006.
12. P. Apparao, R. Iyer, X. Zhang, D. Newell, and T. Adelmeyer, Characterization & Analysis of a Server Consolidation Benchmark, ACM/USENIX International Conference on Virtual Execution Environments (VEE), 2008.
13. P. Apparao, S. Makineni, and D. Newell, Characterization of Network Processing Overheads in Xen, IEEE International Workshop on Virtualization Technology in Distributed Computing (VTDC), 2006.
14. 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 Symposium on Operating Systems Principles (SOSP), 2003.
15. XenSource, A Performance Comparison of Commercial Hypervisors, http://www.xensource.com/Documents/hypervisor_performance_comparison%20_1_0_5_with_esx-data.pdf, 2007.

16. Deshane, Todd, et al. "Quantitative comparison of Xen and KVM."Xen Summit, Boston, MA, USA (2008): 1-2.
17. Tickoo, Omesh, et al. "Modeling virtual machine performance: challenges and approaches."ACM SIGMETRICS Performance Evaluation Review 37.3 (2010): 55-60.
18. SPEC to Develop Standard Methods of Comparing Virtualization Performance, Press Release, http://www.spec.org/specvirtualization/pressrelease.html, 2006.
19. SPEC, http://www.spec.org/
20. Netperf, http://netperf.org/
21. Uperf, http://uperf.org/
22. Httperf, http://www.hpl.hp.com/research/linux/httperf/
23. Iozone Filesystem Benchmark, http://www.iozone.org/
24. Bonnie++, http://www.coker.com.au/bonnie++/

Last Modified: April 24, 2013
This and other papers on latest advances in Performance Analysis and Modeling are available on line at http://www.cse.wustl.edu/~jain/cse567-13/index.html
Back to Raj Jain's Home Page