

A PERFORMANCE COMPARISONS OF DIFFERENT VIRTUAL MACHINES IN HYBRID CLOUD SYSTEM

Dheeraj K. Rathore¹, Dr. Vibhakar Pathak²

¹Computer Science department, Poornima university Jaipur, India

²Prof. Department of CS/IT Arya College of Engg. and IT, Jaipur, India

Abstract

The virtualization of IT infrastructure enables the consolidation and pooling of IT resources so that they can be shared over diverse applications. With the help of virtualization logical abstraction of physical resource is possible which is very lucrative for the IT business. Thus we can save the capital of IT business. Virtualization is very important for cloud computing because the delivery of services is simplified by providing a platform, infrastructure and services for optimizing complex IT resources in a feasible manner, which makes cloud computing more cost effective. This paper presents a comparative analysis of different virtual machine of Ubuntu and WinXP's performance in the CPU Time, Memory usage, percent CPU usage in the KVM hypervisor virtualization. Our comparison was carried using libvirt tool kit commands result and sysbench benchmark tool. Our goals are that how the different virtual machine of single host affects the performance and make informed decision about choice of best virtual machine assign to the cloud guest.

Key words: Virtualization, Virtual Machine, KVM (Kernel-based virtual machine),

Introduction

Virtualization has changed the analogy of businesses concepts and their applications, first by enabling much greater consolidation within private data centers, and more recently as a driving technology behind cloud computing. Whether it is used with private, public and hybrid resources. Virtualization simplifies management, speeds up deployment, and improves resource efficiency and best use of hardware. Virtualization enables new features such as performance management, flexibility in storage and reliability services to be applied without need to change in application or host. While the overheads of the virtualization layer still prevent it from being used in performance critical domains such as high performance computing. Virtualization has become the norm for running a growing set of applications.

Virtualization has been used recently by the ease with which Infrastructure as a service (IaaS) cloud platforms can rent users virtual machines (VMs). Many enterprises appreciate the rapid deployment and flexible scalability of these clouds. Virtualization consolidates the workload of physical machines and a single machine can multiplexed to many

users. Thus virtualization improves the efficiency of infrastructure, data storage and services by allowing the small instances of physical machine [1].

The hypervisor is a software layer which manages the virtualization platform between a Virtual machine (VM) and the physical resources it makes use of [2]. We use Kernel-based Virtual Machine for our initial set of experiment consist of Ubuntu14.04 AMD64 on the Host Machine. All the Virtual Machines (Domain) are installed by One by One with the help Virtual Machine Monitor (VMM). We used Ubuntu10, 11, 12, 14.04 and Window XP for the guest Operating Systems. Now guest Operating System run on the host machine. The hardware system was a Lenovo system with a 3.30 GHz x 4 Intel Core i32120, 4 GB of RAM, 223.6 GB hard drive. The results were evaluated by Libvirt Command using *virsh* command.

Our results are evaluated by three scenarios first is only one Ubuntu Virtual Machine (VM) is running at a time second is combination of VM of Ubuntu run and last is combination of Window XP and Ubuntu run simultaneously. Finally we compare the all results and give the resource provisioning.

A. Background and Related work

Virtualization refers to abstraction of physical resource (RAM, Secondary disk etc) and Software resources (Servers applications). Virtualization provides interfaces for applications and operating systems. Hardware resources can be used among several operating systems with the help of virtualization. It is possible to virtualized complete machines or just parts of the machine by using virtualization. Virtualization is generally used in mainframe system [3]. There are three different approaches to achieve virtualization which are: Full Virtualization, Para Virtualization and Hardware level virtualization. Full virtualization makes virtual machines (VMs) very flexible. The host OS emulates the complete hardware package which are visible for the virtualized operating systems. Guest operating system assumes that it has all the controls of hardware and software resource to it. For example emulate x86 architecture to run Windows XP on a Macintosh with PPC architecture. Para virtualization implements various instances of virtual guest operating system. The vir-

tual machines use the virtualized image of operating system. This kind of virtualization is also known as Single Kernel Image (SKI) or container-based virtualization. Virtual server operating system is virtualized in this approach, which allows the hypervisor and the guest OS to collaborate on achieving the fastest, most robust performance. Hardware level virtualization is used in high performance and isolation for server technology. In hardware level virtualization virtual machine monitor directly run on the hardware that controls the access of the guest operating system to the hardware.

A. Hypervisor

There are various hypervisors are available for achieving the virtualization. Hypervisor is software layer between hardware and Virtual machine (VM) which enables virtualization. It is responsible to creating the virtual environment on which the guest virtual machines operate. It managed the guest systems and makes sure resources are allocated to the guests as it requires [4]. There are two types of hypervisor: Type I and Type II. Type I is lowest level of hypervisor which directly run on the host machine. All the resource management of Virtual Machine is done by hypervisor. This hypervisor is generally used in server virtualization. Type II hypervisor run on full host operating system to operate. This means that is working on the top of the host operating system. It requires fewer driver/hardware to interface with the host operating system. So this hypervisor has less issue to operating this. This hypervisor is used in Java Virtual Machine (JVM) to used application portability.

We use Kernel-based Virtual Machine (KVM) hypervisor to implement our project.-based virtual Machine (KVM) is used for experimentation. Kernel-based Virtual Machine is a kernel module for the Linux operating that allows for full virtualization in X86architecture. First introduced in 2006, and subsequently accepted into the Linux kernel tree for version 2.6.20. A hypervisor is consist of several software components generally having to write a scheduler, memory management, I/O-stack for a new hypervisor, as well as drivers for the architecture on the host machine. KVM unlike other hypervisors such as Xen, has focused the implementation on the guest handling of the virtualization and other resource management component. Since Linux has developed into a secure and stable operating system, as well as having some of the most important features for a hypervisor. When the KVM is installed to the machine it runs as a kernel module and work as a device on /dev/kvm. The device or module is work interface between guest and hardware. Every guest is initiated in the user space tool and supported by the QEMU. KVM manage the lower part of Virtual Machine such as controlling the guest when host migrate in hardware. Each guest processor is run its own thread that is spawned from the user space tool which is schedule by the

hypervisor. The memory management of guest is done by user space tool, which maps the memory from the guest physical memory to the host virtual memory. Input/Output are also managed by the user space tools. KVM is never directly used, in most cases of Linux operating systems the kernel module is loaded and present by default.

B. Commands of Libvirt

We use Libvirt toolkit of communicate the hardware to Guest operating system performance. Libvirt is a toolkit which interacts with the virtualization capabilities of Linux and other Operating Systems. The goal of Libvirt is to provide a common and stable layer sufficient to securely manage domains on a node. With the Libvirt node can be managed in remotely connection. Libvirt should provide all APIs needed to do the management, such as: provision, create, modify, monitor, control, migrate and stop the domains - within the limits of the support of the hypervisor for those operations. Libvirt also provide the monitoring management of the Virtual machines and Libvirt's API implements the management policies, checking the domain states, resource consumption of node. We use following commands of Libvirt with keyword *virsh* for our experimentation.

Table No. 1 List of commands used for experimentation.

Command's Key-word	Description
Node (Host Commands)	
Nodeinfo	Return basic information of node
node cpu-stats – percent	Return the CPU usage time in percentage
node memstats	Return the memory used by node
Domain Commands	
Dominfo	Return basic information of Domain
cpu-stats	Return the CPU usage time by Domain
Vcpuinfo	Virtual Machine and host machine CPU information.
Monitoring Commands	
list –all	Return the all domain name and thre Id with current state (running or shut of)
Reboot	Run the reboot command in Guest domain

C. Benchmarking

This tool is a modular, cross-platform and multi-threaded benchmark tool for evaluating OS parameters that are important for a system running a database under intensive load. Sysbench tool evaluate the file I/O performance, scheduler performance, memory allocation and transfer speed, POSIX threads implementation performance and database server performance. In Ubuntu14.04 sysbench is installed by following commands.

apt-get install sysbench

After install this tool We analyzed the CPU performance of Ubuntu14.04 LTS.

Methodology

The methodologies for our performance comparison of Virtual Machines depend on the workload taken by VM on to host (Ubuntu14.04). The components include CPU Time, CPU frequency, Memory (RAM, Buffer, and Cache), VCPU Time. Each component has different virtualization requirements that need to be tested with different workloads. We follow this with a set of more general workloads representative of higher-level applications. Thus how we assume it as a Hybrid Cloud System.

When a Virtual Machine created, it assign to specific number of Virtual CPUs (VCPU). VCPU ensure that how many cores this VM can use on to the host. But VCPU does not guarantee that physical machine is dedicated to VM. It represents a flexible assignment of physical machine to the VM. It is totally based on the scheduling weights of the different Virtual Machine. We implemented that a VMs are assign to single VCPU or maximum two VCPU in our system.

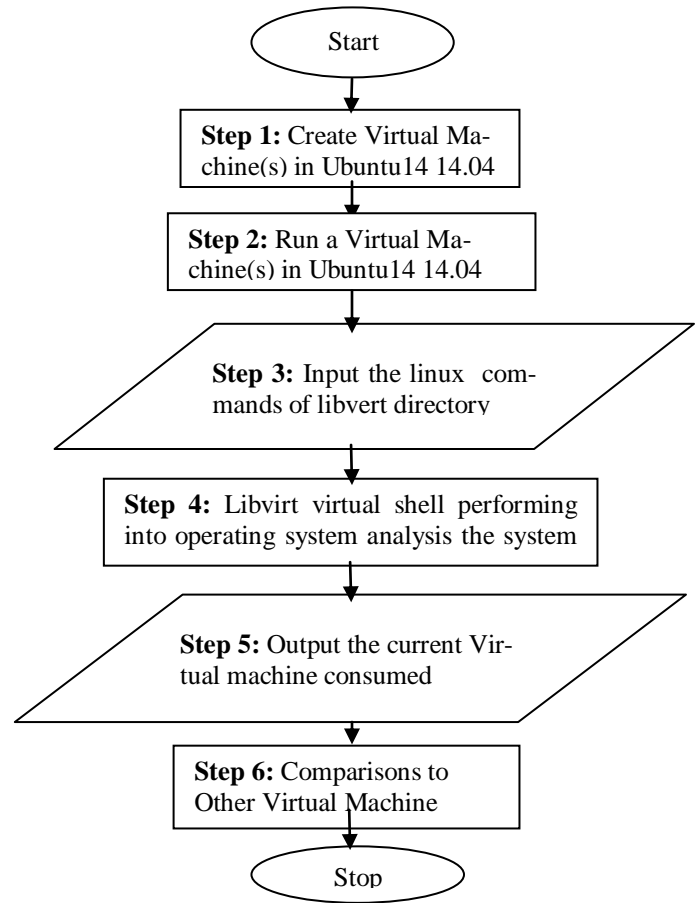
We divided our system into three different-2 scenarios so the analogue to the hybrid Cloud is full fill. In scenario 1 only the single Virtual Machine is run at a time. In this scenario only single virtual machine runs at time. After stable the virtual machine result is evaluated from every virtual machine. The parameters of results are CPU time and memory of Host machine (Ubuntu14.04). After that evaluates the result of virtual machine VCPU time, CPU Time, VCPU information. This scenario has four VM of Ubuntu flavor Ubuntu10, Ubuntu11, Ubuntu12, Ubuntu14.04.

In second scenario Combination of Ubuntu's flavor Virtual Machine runs at time. After stable the virtual machine result is evaluate for every virtual machine. The parameters of results are CPU time and memory of Host machine (Ubuntu14.04). After that evaluates the result of virtual machine VCPU time, CPU Time, VCPU information. This scenario has three Ubuntu's flavor Ubuntu10+Ubuntu11, Ubuntu11+Ubuntu14, Ubuntu10+Ubuntu14.

In third scenario Combination of Ubuntu's flavor and WinXP's virtual machine runs at time. The parameters of results are CPU time and memory of Host machine (Ubuntu14.04). After that evaluates the result of virtual machine VCPU time, CPU Time, VCPU information. This scenario has two Virtual Machine WindowXP and WindowXP+Ubuntu10.

After the implementation of Virtual Machines the Libirt tool kit is used for gaining the work load of the host (Ubuntu14.04 LTS) and various Guest. The workload is evaluat-

ed in parameters of CPU Time, memory and VCPU Time. When two Virtual Machines are running at a same time the Libvirt Commands is also able evaluate the workload of the individual virtual Machine. Finally we check the benchmarking of the system by sysbench tool. A general flow chart of our project is as following.



The goal of our system setup is to provide complete fairness in all the aspects that can affect the system performance.

Hardware setting: For a fair comparison we run all host in same Host machine. Host Machine has processor of Intel core i3-2-2120 CPU @ 3.30 x 4, its RAM memory is 3.7 GB, Operating System type is 64-bits and Secondary Memory is 223.6 GB. Ubuntu14.04 LTS Operating System is installed in the Host Machine. Thus we established our Host Machine for the Project. In the host we installed the KVM/QEMU hypervisor and also installed Libvirt and sysbench benchmarking tool.

Guest operating Systems: In the guest operating system we implemented the Virtual Machine by Ubuntu10, Ubuntu11, Ubuntu12, Ubuntu14.04 LTS and Window XP.

Result and Discussion

This work has three scenarios so result show according to that.

Scenarios 1: Results are taken of single virtual machine.

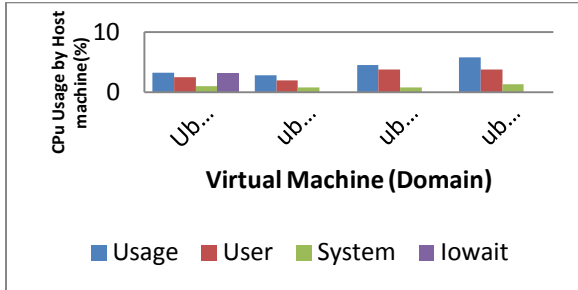


Figure 1: percent CPU usage by the host machine while different-2 guest OS are running

Figure 1 shows the comparative analysis of Host Machine (Ubuntu14.04) with the parameter CPU use. The Usage of CPU in Processing the Virtual machine Ubuntu14.04 consumes the maximum CPU Usage 5.8 %. So it clear that ubuntu14.04 consume the maximum CPU for processing the virtual machine. Ubuntu14.04 also consumes 3.8 % of CPU in User processing. This is the maximum in the all virtual machine. The only virtual machine which use the I/O wait is ubuntu10. While testing in memory comparison Ubuntu14.04 has least free memory which is 165252KiB. Ubuntu14 use the maximum memory and the ubuntu10 use the minimum memory of the host machine.

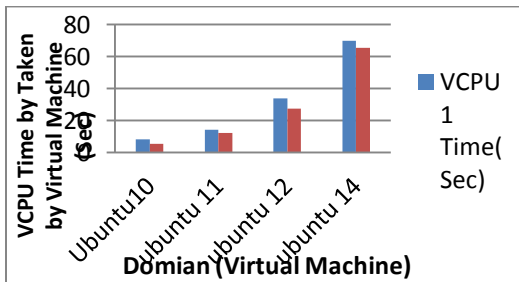


Figure 2 VCPU Time usage by Virtual Machines

Figure 2 shows the VCPU time of different-2 Virtual machine. This figure shows the both CPU core of the system VCPU time. Figureshow that the ubuntu14.04 takes the maximum VCPU time in both cores. The VCPU time of both core are 69.7 sec and 65.3 sec. as expected ubuntu10use. This figure show both part of VCPU it show the VCPU1 and VCPU2 time in sec. Ubuntu10 consume the minimum time of VCPU which is 8.2 sec for VCPU1 and 5.3 sec for VCPU2.

Scenario 2: Results are taken of Combination of Virtual Machine.

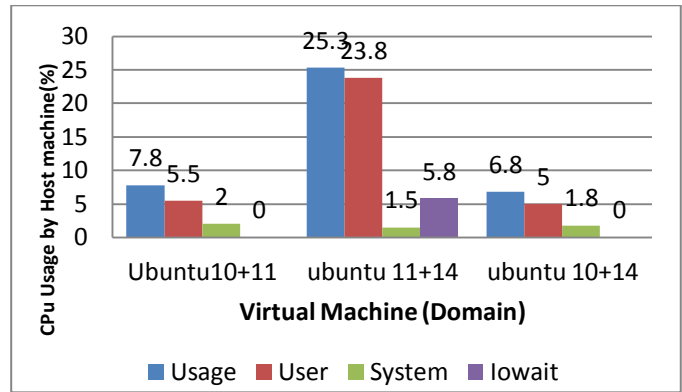


Figure 3: percent CPU usage by the host machine while different-2 guest OS are running in Combinations

Figure 3 shows the comparative analysis of Host Machine (Ubuntu14.04) with the parameter CPU usage while the combination of Ubuntu10 + Ubuntu11, Ubuntu11 + Ubuntu14, Ubuntu10 + Ubuntu14. It shows that usage time is taken by Ubuntu11+Ubuntu14 is 25.8 % which is highest in the all virtual machine with respect to other virtual machine. So it clear that Ubuntu11+Ubuntu14.04 consume the maximum CPU for processing the virtual machine. Ubuntu11+14 also use the I/O wait 5.8 %. So it clear that in stable state the combination of Ubuntu11+14 takes the highest CPU use percentage.

Ubuntu11+11 have least free memory (132650KiB) so it uses the memory maximum. After analyze the buffer memory the virtual machine again the ubuntu11+14 uses minimum buffer memory in all virtual machine which is (18708 Kib). After that analyzed the cache memory again the ubuntu10+11 uses the minimum cache memory which is (581608 Kib). Ubuntu11+14 have the maximum free memory (242716 Kib). Thus we can say that ubuntu11+14 use the maximum memory and the ubuntu10+11 use the minimum memory of the host machine

Scenario 3: Results are taken of Combination Window XP and Ubuntu10

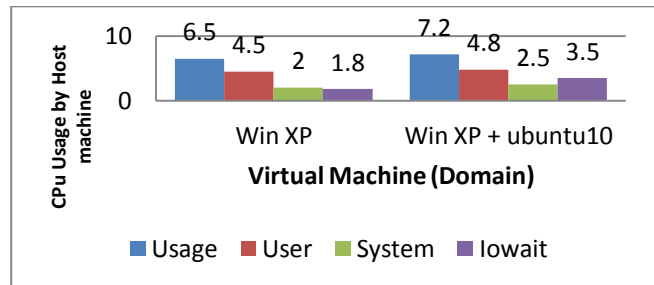


Figure 4: percent CPU usage by the host machine while different-2 guest OS are running in Combinations

Figure 4 is comparative analysis of Host Machine (Ubuntu14.04) with the parameter CPU usage while the window XP runs and combination of Window XP + ubuntu10 run simultaneously. It shows that usage time is taken by WinXP+ubuntu10 is 7.2 % which is greater than single XP System. The user time of WinXP+ubuntu10 is 4.8 %. I/O wait of WinXP+ubuntu10 is 3.5% which is greater than single XP virtual machine. Thus it can be said that combination of WinXP+ubuntu10 takes more system CPU percent.

A. Discussion

Our experimental results paint a complicated picture about the relative performance of different virtual Machines and its combination. Clearly, that every virtual Machine has different applications and workload on to it. So it is helpful to administrator choose the best virtual Machine according to user need. These results are very beneficial for the infrastructure management in hybrid cloud. After the checking by the sysbench it gives 95 % performance in CPU event.

Quantitative comparison of xen and kvm. Xen "Submit, June 2008.

REFERENCES

- [1.] Vijayaraghavan Soundararajan and Kinshuk Govil, "Challenges in building scalable virtualized datacenter management," SIGOPS Oper. Syst. Rev., vol. 44, no. 4, pp. 95–102, Dec. 2010.
- [2.] Timothy Wood, Ludmila Cherkasova, Kivanc Ozonat, and Prashant Shenoy, "Profiling and modeling resource usage of virtualized applications," in Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, New York, NY, USA, 2008, Middleware '08, pp. 366–387, Springer-Verlag New York, Inc.
- [3.] Stuart Devenish, Ingo Dimmer, Rafael Folco, Mark Roy, Stephane Saleur, Oliver Stadler, and Naoya Takizawa, "Ibm powervm virtualization introduction and configuration," Redbooks, 1999.
- [4.] VMware, "Understanding full virtualization, paravirtualization, and hardware assist," VMware White Paper, 2007.
- [5.] P. Apparao, R. Iyer, and D. Newell, Towards Modeling & Analysis of Consolidated CMP Servers, Workshop on the Design, Analysis, and Simulation of Chip Multi-Processors (dasCMP), 2007.
- [6.] Bao Rong Chang, Hsiu-Fen Tsai, "Evaluation of Virtual Machine Performance and Virtualized Consolidation Ratio in Cloud Computing System" Journal of Information Hiding and Multimedia Signal Processing 2013 ISSN 2073-4212 Ubiquitous International vol no 4 page no 3
- [7.] Todd Deshane, Zachary Shepherd, Jeanna N. Matthews, Muli Ben-Yehuda, Amit Shah, and Balaji Rao,