ABSTRACT

Virtualization refers to technologies designed to provide a layer of abstraction between computer hardware systems and the software running on them. By providing a logical view of computing resources, rather than a physical view, This virtual view of the resources is not restricted by the implementation, geographic location or the physical configuration of underlying resources. Commonly virtualized resources include computing power and data storage and recently network sharing resources.

This research compare two virtualized environments (Xen and Vmware), the comparison based on two criteria, the first measures the memory performance (RAM) and the second measures the processor performance (CPU) I offered same hardware specification to the both systems. And same size and structured of data , this data is part of ERP system that belong to one of Sudanese electricity companies.

The result that found show differences between two virtual systems, it is very small differences, further more the research appears that some case of large data XEN gives better performance result than VMWARE this due to better management provided by XEN.

المستخلص

الأنظمة الافتراضية تشير إلي تقنية صممت لتقدم طبقة بين الأجهزة و الأنظمة التي تعمل عليها وذلك عن طريق منظور منطقي أكثر من محسوس لموارد الحاسوب . هذا المنظور المنطقي غير مقيد بالتنفيذ أو الموقع الجغرافي أو إعدادات الأجهزة . في العموم الموارد الافتراضية تتضمن موارد الطاقة والأقراص الصلبة و في الآونة الأخيرة هناك موارد الشبكات.

تم في هذا البحث عمل مقارنة بين نظامين افتراضيين (Xen and Vmware) و تمت المقارنة علي محورين الاول هو مقارنة الاداء بالنسبة لذاكرة الوصول العشوائي (RAM) و الثاني هو مقارنة الاداء لوحدة المعالجة المركزية (CPU) . و وفرت مواصفات hardware متماثلة لكلا النظامين كما تم استخدام نفس البيانات و بنفس الحجم . هذه البيانات هي جزء من ERP system لأحدي شركات الكهرباء السودانية.

النائج التي حصلت عليه تظهر فروقات ضئيلة بين النظامين الافتراضيين، فروقات ضئيلة جدا ، و ابعد من ذلك اظهر البحث الي انه في بعض الحالات XEN تعطي اداء افضل وهذا ما ظهر في البيانات ذات الحجم الكبير

وذلك بسبب الادارة الجيدة التي توفرها ال XEN للذاكرة و المعالج

Acknowledgment

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah. All praises to Allah for the strengths and His blessing in completing this thesis. Special appreciation goes to my supervisor, Dr. TALAAT MOHI ELDEEN WAHBY for supervising this thesis. I must express my deep gratitude to my parents, uncles and aunts and love to all my family who have supported me and guided me to reach this stage.

Table of Contents

List of figures

List of Tables

1.1 Overview

Virtualization refers to technologies designed to provide a layer of abstraction between computer hardware systems and the software running on them. By providing a logical view of computing resources, rather than a physical view, virtualization solutions make it possible to do many of very useful things.

The virtual view of the resources is not restricted by the implementation, geographic location or the physical configuration of underlying resources. Commonly virtualized resources include computing power and data storage [1].

Virtualization can also help a company cut down on energy consumption, since there are fewer physical servers consuming power. That's especially important, given the trend toward green IT planning and implementation.

Virtualization is being used by a growing number of organizations to reduce power sumption and air conditioning needs and trim the building space and land requirements that have always been associated with server farm growth. Virtualization also provides high availability for critical applications, and streamlines application deployment and migrations. Virtualization can simplify IT operations and allow IT organizations to respond faster to changing business Demands.

The performance of virtualized system will not be as same as non-virtualized system this because the virtualized system doesn’t running on machine directly, it uses intermediate to make the order take place like system callas .

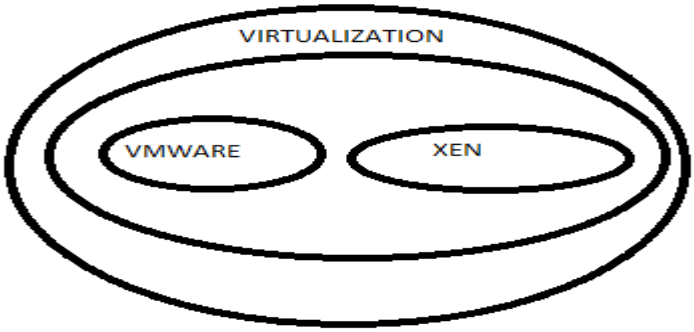

Figure 1.1: Scope of the research

1.2Scope of the research

Server-based computing (SBC) refering to the technology by which applications are implemented, controlled, supported and functioned on the server instead of the client. Upgrading hardware, application deployment, backing up of data and technical support are simplified in a server-based environment.

The focus of this research is to compare between two virtual machines performance . Beside illustration the advantages of virtualization in general and focus on (VMware and Xen virtual machines) as topic of research.

1.3Problem statement

The two case study on this research (VMWare ESXi and Xen hypervisor) belonging to type-1 or baremetal hypervisor, which makes it possible to run many instances of an operating systems or indeed different operating systems in parallel on a single machine (or host).

The criteria that measures is

- processor utilization
- total usage of memory
- Both processor utilization and total usage of memory

1.4Question of the research

- Which of these two virtual server is best in processor utilization and total usage of memory .
- Is open source virtual machines (XEN) performance do as same as commercial virtual machines (VMWARE) .
- Is open source virtual machines (XEN) good enough to take place instead of commercial(Vmware). .

1.5 Objectives of the research

- Use a technology that reduce the cost of Hardware and software on IT industry.
- Be familiar with open-source technologies to avoid getting commercial software license which is sometimes forbidden for our country for political reasons .
- State-of-the-art of the used technology (hardware Virtualization).

1.6 Research methodology

- Xen® hypervisor, the powerful open source industry standard for virtualization
- VMware server with client side to maintain operations.
- Hardware Utilization Monitoring Tools
- Database Postgresql Operating system Linux Ubuntu Server 12.0
- Do the experiments and analyze result.
- Gives recommendations.

1.7 Research organization

This research consist of five chapters which they are:

Chapter one: General overview Scope of the research, Problem statement, Question of the research, Objectives of the research, Research methodology, Research organization

Chapter two: This chapter is about Review talking about the various types of virtualization full hardware and Paravirtualization illustrate the differences and

advantages of types, the end of this chapter focused on the tools used (VMWare an Xen hypervisor) , Related work

Chapter three: this chapter about the describe the experiments and methodology, screen shots which shows technique used to make this work take place.

Chapter four: this chapter focus on the results and analyze these results.

Chapter five: conclusions, Recommendations and future work.

2.1 Review

The techniques of virtualization appears as need to developing robust time-sharing systems by IBM, main frames Time-sharing refers to the shared usage of computer resources among a large group of users, aiming to increase the efficiency of both the users and the expensive computer resources they share this concept took place in 60s to 70s of the past century [1] . This model represented a major breakthrough in computer technology: the cost of providing computing capability dropped considerably and it became possible for organizations, and even individuals, to use a computer without actually owning one. Similar reasons are driving virtualization for industry standard computing today: the capacity in a single server is so large that it is almost impossible for most workloads to effectively use it. The best way to improve resource utilization, and at the same time simplify data center management, is through virtualization [2].

The way virtualization working Multiple users share the same CPU, Memory, Hard Disk, and other hardware resources, pool of resources, but each have their own profile, separate from the other users on the system. Depending on the way the system is configured, the user may be able to install their own set of applications, and security is handled on a per user basis. Virtual machines hide the hardware specification from the OS and make virtual hardware OS working on it . When OS want execute a job VM take instances of all resources needed to make this job be done. Sharing these resources with other VM.

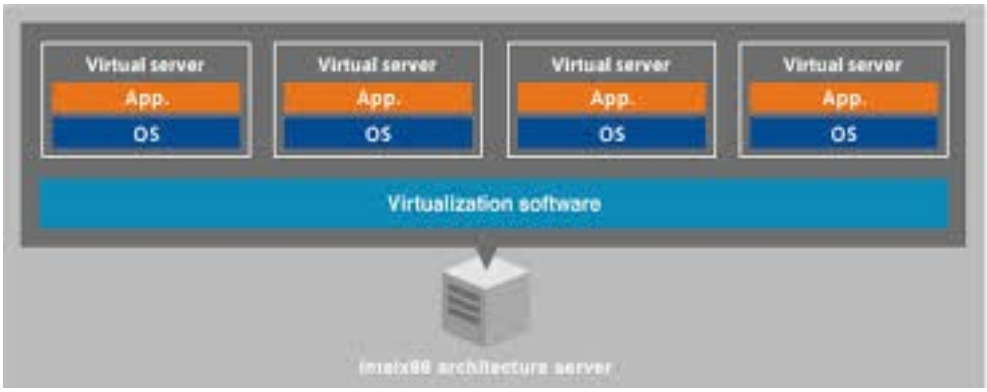

Figure 2.1: VIRTUAL SYSTEM

There are many types of virtualization. The most common approach is server virtualization here are major three types of server virtualization.

2.2 types of virtualization

Types of virtualization are three which is:

- **Full virtualization**:

Almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified Full virtualization requires that every salient feature of the hardware be reflected into one of several virtual machines – including the full instruction set, input/output operations, interrupts, memory access, and whatever other elements are used by the software that runs on the bare machine, and that is intended to run in a virtual machine[3].
full virtualization is a virtualization technique used to provide a certain kind of virtual machine environment, namely, one that is a complete simulation of the underlying hardware. Full virtualization requires that every salient feature of the hardware be reflected into one of several virtual machines – including the full instruction set, input/output operations, interrupts, memory access, and whatever other elements are used by the software that runs on the bare machine, and that is intended to run in a virtual machine. In such an environment, any software capable of execution on the raw

hardware can be run in the virtual machine and, in particular, any operating systems. The obvious test of virtualization is whether an operating system intended for stand-alone use can successfully run inside a virtual machine..

Full virtualization is possible only with the right combination of hardware and software elements.

A key challenge for full virtualization is the interception and simulation of privileged operations, such as I/O instructions. The effects of every operation performed within a given virtual machine must be kept within that virtual machine – virtual operations cannot be allowed to alter the state of any other virtual machine, the control program, or the hardware. Some machine instructions can be executed directly by the hardware, since their effects are entirely contained within the elements managed by the control program, such as memory locations and arithmetic registers. But other instructions that would "pierce the virtual machine" cannot be allowed to execute directly; they must instead be trapped and simulated. Such instructions either access or affect state information that is outside the virtual machine.

Full virtualization has proven highly successful for:

- sharing a computer system among multiple users;
- isolating users from each other (and from the control program);
- emulating new hardware to achieve improved reliability, security and productivity.

- **Hardware virtualization:**

Hardware virtualization refers to the creation of virtual (as opposed to concrete) versions of computers and operating systems.

Computer hardware virtualization is virtualization of computers or operating systems. It hides physical characteristics of computers platforms from users, instead showing another abstract computing platform. The software that control the virtualization used

to be called the" control program" at its origins. But nowadays the terms "Hypervisor" or "Virtual machine monitor" are preferred [4].

Hardware virtualization is an evolving technology that is gaining popularity in server platforms. The basic idea of the technology is to consolidate many small physical servers into one large physical server so that the processor can be used more effectively. The operating system running on a physical server gets converted into a distinct OS running inside the virtual machine.

The hypervisor controls the processor, memory and other components by allowing several different operating systems to run on the same machine without the need for a source code. The operating system running on the machine will appear to have its own processor, memory and other components.

Hardware virtualization has many advantages because controlling virtual machines is much easier than controlling a physical server.

The term hardware virtualization is also known as hardware-assisted virtualization.

Figure 2.2: HARDWARE VIRTUALIZATION

- **Paravirtualization**

The virtualization technique that presents a software interface to virtual machines that is similar, but not identical to that of the underlying hardware.

Paravirtualization is an enhancement of virtualization technology in which a guest OS is recompiled prior to installation inside a virtual machine. Paravirtualization allows for an interface to the virtual machine that can differ somewhat from that of the

underlying hardware. This capacity minimizes overhead and optimizes system performance by supporting the use of virtual machines that would be underutilized in conventional or full virtualization [5].

The main limitation of Paravirtualization is the fact that the guest OS must be tailored specifically to run on top of the virtual machine monitor (VMM), the host program that allows a single computer to support multiple, identical execution environments. However, Paravirtualization eliminates the need for the virtual machine to trap privileged instructions. Trapping, a means of handling unexpected or unallowable conditions, can be time-consuming and can adversely impact performance in systems that employ full virtualization.

Paravirtualization is an expansion of a technology that has existed for years in the IBM OS known as VM. Xen, an open-source software project, incorporates Paravirtualization.

Performance is the most well known advantage that Paravirtualization has, however with paravirtualized device drivers in a fully virtualized OS this advantage is actually getting smaller over time

Another major concern is performance measurement of applications. On a system with 3 virtual CPUs per physical CPU, you can easily end up with a "fudge factor" of 3, making performance measurement totally useless [6]

Figure 2.3: PARAVIRTUALIZATION

2.3 Research tools

2.3.1 VMware virtual machine

VMware ESXi is an operating system-independent hypervisor based on the VMkernel operating system interfacing with agents that run atop it. ESXi is the exclusive hypervisor for VMware vSphere 5.x licenses[7].

VMware ESX is an enterprise-level computer virtualization product offered by **VMware**, Inc. ESX is a component of **VMware's** larger offering, VMware Infrastructure, which adds management and reliability services to the core server product. **VMware** is replacing the original ESX with ESXi.

VMware ESX and **VMware** ESXi are Type 1 hypervisors that are **VMware's** enterprise software hypervisors for guest virtual servers that run directly on host server hardware without requiring an additional underlying operating system[8].

The basic server requires some form of persistent storage (typically an array of hard disk drives) that store the hypervisor and support files. A smaller footprint variant, ESXi, does away with the first requirement by permitting placement of the hypervisor on a dedicated compact storage device. Both variants support the services offered by **VMware** Infrastructure.

VMware describes an ESXi system as similar to a stateless compute node. State information can be uploaded from a saved configuration file. ESXI's VMkernel interfaces directly with **VMware** agents and approved third-party modules. Virtualization administrators can configure **VMware** ESXi through its console or the **VMware** vSphere Client and check VMware's Hardware Compatibility List for approved, supported hardware on which to install ESXi.

ESX licensees can choose to deploy ESXi instead of ESX on any given server. Before ESXi, **VMware** offered the ESX hypervisor, which comprised more parts, such as the console operating system (OS) and firewall. Remote command line interfaces and system management standards replace the service console functions. The hypervisor supports Auto Deploy and custom image creation, along with other tools that were not

included in ESX. **VMware** reports that ESXI's architecture occupies less than 150 MB of space -- 32 MB of on-disk space -- as compared to about 2 GB with ESX [8].

Architecture

VMware states that the ESX product runs on *bare* metal. In contrast to other **VMware** products, it does not run atop a third-party operating system, but instead includes its own kernel. A Linux kernel was started first, and is used to load a variety of specialized virtualization components, including **VMware's** VMkernel component. This previously booted Linux kernel then becomes the first running virtual machine and is called the service console. Thus, at normal run-time, the VMkernel is running on the bare computer and the Linux-based service console runs as the first virtual machine. As of version 4.1, **VMware** has dropped development of ESX and now focuses exclusively on ESXi, which is devoid of a Linux kernel.

The VMkernel itself, which **VMware** says is a microkernel, has three interfaces to the outside world:

- Hardware.
- Guest systems.
- Service console (Console OS).

Interface to hardware

The VMkernel handles CPU and memory directly, using scan-before-execution (SBE) to handle special or privileged CPU instructions and the SRAT (system resource allocation table) to track allocated memory.

Access to other hardware (such as network or storage devices) takes place using modules. At least some of the modules derive from modules used in the Linux kernel. To access these modules, an additional module called vmklinux implements the Linux module interface. According to the README file, "This module contains the Linux emulation layer used by the kernel

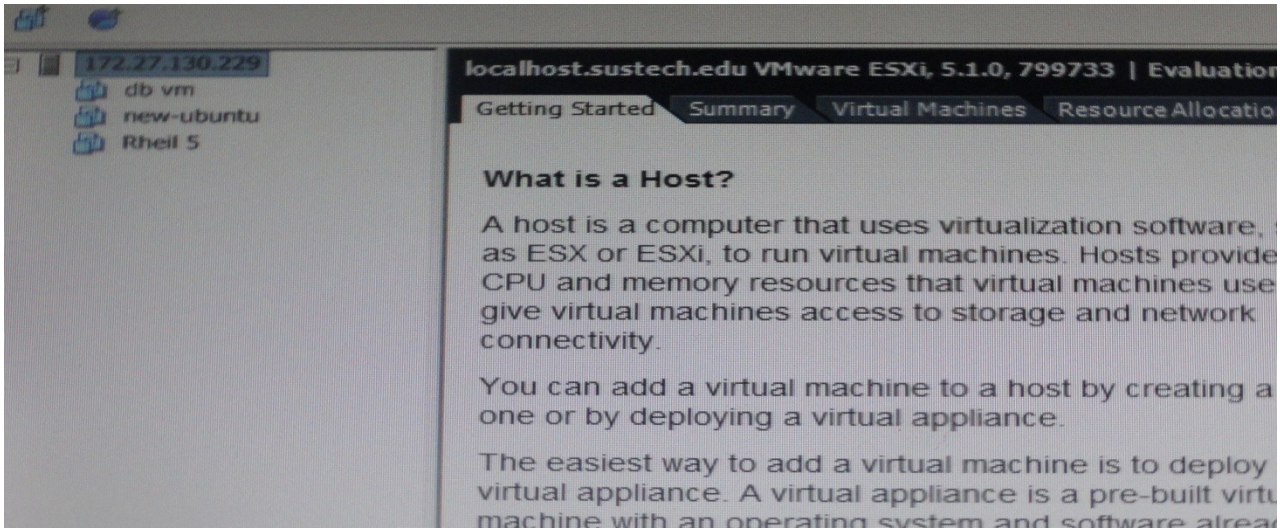

Figure 2.4 Vsphere

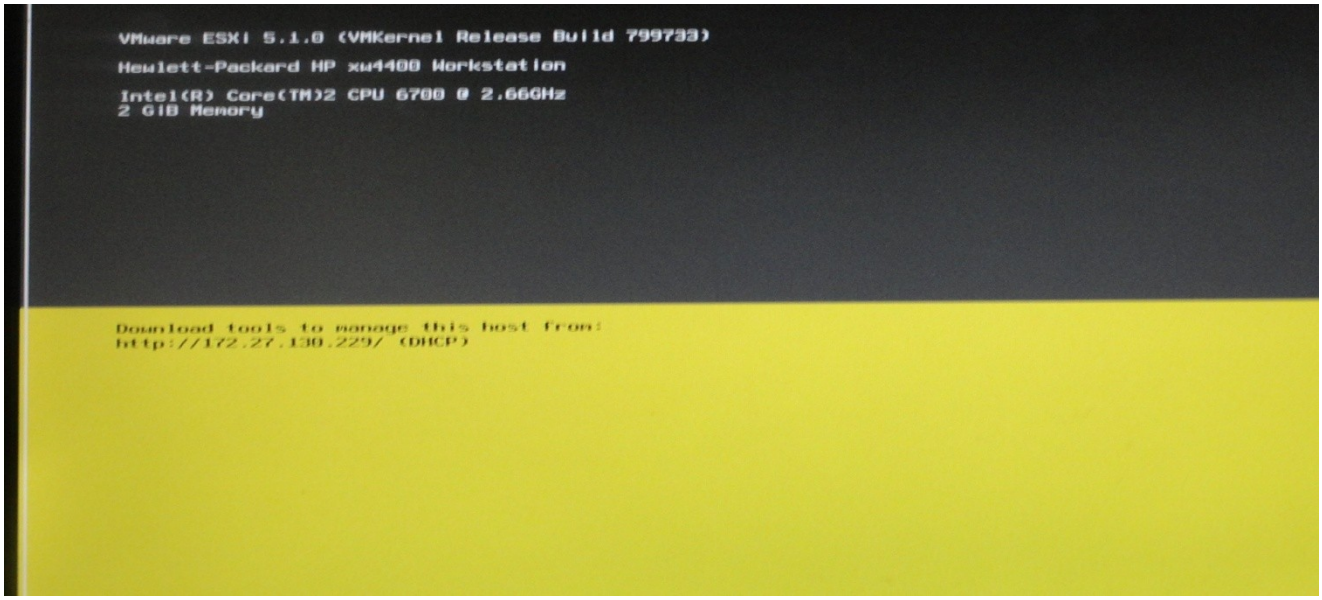

Figure 2.5 Vmware esxi

- ## Service console

In ESX (and not ESXi), the Service Console is a vestigial general purpose operating system most significantly used as bootstrap for the VMware kernel, VMkernel, and secondarily used as a management interface. Both of these Console Operating System functions are being deprecated from version 5.0, as VMware migrates exclusively to the 'embedded' ESXi model, current version being ESXi. The Service Console, for all

intents and purposes, is the operating system used to interact with VMware ESX and the virtual machines that run on the server.

- **VMware ESXi**

VMware ESXi is a smaller footprint version of ESX that does not include the ESX Service Console. It is available without the need to purchase a vCenter license as a free download from VMware with some features disabled.

VMware ESXi was originally a compact version of VMware ESX that allowed for a smaller 32 MB disk footprint on the Host. With a simple configuration console for mostly network configuration and remote based VMware Infrastructure Client Interface, this allows for more resources to be dedicated to the Guest environments.

There are two variations of ESXi, VMware ESXi Installable and VMware ESXi Embedded Edition. The same installation media will install to either one or the other of these installation modes depending on the size of the target media. It has the ability to upgrade to VMware Infrastructure 3 or VMware vSphere 4.0 ESXi.

Originally named VMware ESX Server ESXi edition, through several revisions the product finally became VMware ESXi 3. New editions then followed: ESXi 3.5, ESXi 4 and now ESXi 5.

To virtualizes Windows 8 or Windows Server 2012 as guest operating systems, the ESXi version must be 5.x or greater

2.3.2 XENHYPERVISOR

XenServer is a turnkey open source virtualization solution that enables out-of-the box virtualization and cloud computing. XenServer includes the Xen Project hypervisor, the enterprise ready XAPI tool stack and integrations for cloud, storage and networking solutions.

Xen is an open-source type-1 or baremetal hypervisor, which makes it possible to run many instances of an operating system or indeed different operating systems in parallel on a single machine (or host). Xen is the only type-1 hypervisor that is

available as open source. Xen is used as the basis for a number of different commercial and open source applications, such as: server virtualization, Infrastructure as a Service (IaaS), desktop virtualization, security applications, embedded and hardware appliances. Xen is powering the largest clouds in production today.

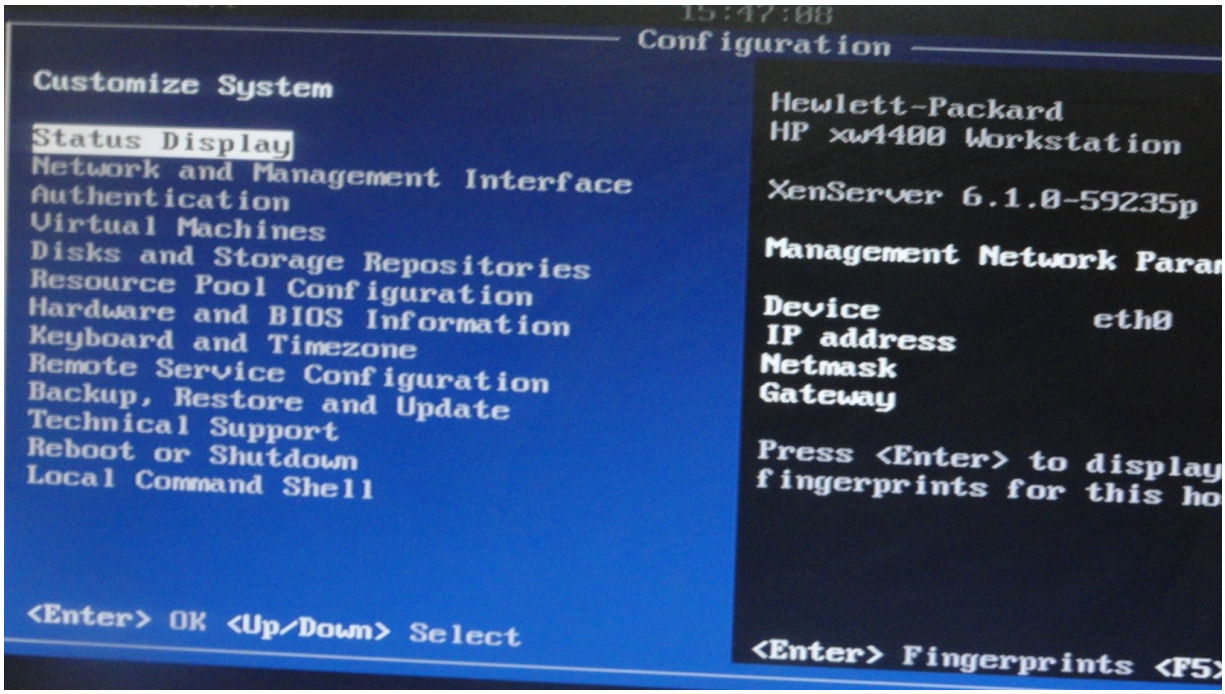

Figure 2.6: XEN Hypervisor

- **Core Platform**

XenServer is based on the Xen Project™ hypervisor. The Xen Project hypervisor is a bare metal virtualization platform used by XenServer to deliver near native application performance for x86 workloads in an Intel and AMD environment.

The Xen hypervisor is available as source distribution from Xen.org. However, you can get recent Xen binaries as packages from many Linux and Unix distributions, both open source and commercial. The Xen Cloud Platform ISOs also contain the Xen Hypervisor in binary form.

The Xen community delivers the Xen hypervisor as a source distribution, following the delivery model of the Linux kernel. Xen is released approximately once a year, with several update releases per year containing security fixes and critical bug fixes.

- ## Xen Cloud Platform

Xen Cloud Platform (or XCP) is a binary distribution of Xen that contains the Xen Hypervisor, a pre-packaged and configured Control Domain and the powerful Xen API tool stack and management API. XCP can be installed from either a single installable ISO.

XCP delivers an enterprise-ready, server virtualization and cloud computing platform with lots of additional management functionality compared Xen: with XCP you can manage pools of host systems, use advanced storage repositories, and take advantage of advanced performance monitoring capabilities.

- ## Xen Architecture

Xen hypervisor runs directly on the hardware and is responsible for handling CPU, Memory, and interrupts. It is the first program running after exiting the boot loader. On top of Xen run a number of virtual machines. A running instance of a virtual machine in Xen is called a domain or guest. A special domain, called domain 0 contains the drivers for all the devices in the system. Domain 0 also contains a control stack to manage virtual machine creation, destruction, and configuration.

The Control Domain (or Domain 0) is a specialized Virtual Machine that has special privileges like the capability to access the hardware directly, handles all access to the system's I/O functions and interacts with the other Virtual Machines. It also exposes a control interface to the outside world, through which the system is controlled. The Xen hypervisor is not usable without Domain 0, which is the first VM started by the system

Tool stack and Console: Domain 0 contains a control stack (also called Tool stack) that allows a user to manage virtual machine creation, destruction, and configuration. The tool stack exposes an interface that is either driven by a command line console, by a

graphical interface or by a cloud orchestration stack such as Open Stack or Cloud Stack [9].

2.4 Related Work:

In recent years, there have been a number of papers comparing the performance of different virtualization environments for x86 such as Xen, VMware and UML.
this paper present the results of running a variety of different misbehaving applications under three different virtualization environments VMware, Xen, and Solaris containers. These are each examples of a larger class of virtualization techniques namely full virtualization, Paravirtualization and generic operating systems with additional isolation layers. To test the isolation properties of these systems, we run six different stress tests - a fork bomb, a test that consumes a large amount of memory, a CPU intensive test, a test that runs 10 threads of I Ozone and two tests that send and receive a large amount of network I/O. Overall, we find that VMware protects the well-behaved virtual machines under all stress tests, but sometimes shows a greater performance degradation for the misbehaving VM. Xen protects the well-behaved virtual machines for all stress tests except the disk I/O intensive one. For Solaris containers, the well-behaved VMs suffer the same fate as the misbehaving one for all tests[10].
Hypervisors are widely used in cloud environments and their impact on application performance has been a topic of significant research and practical interest. We conduct experimental measurements of several benchmarks using Hadoop MapReduce to evaluate and compare the performance impact of three popular hypervisors: a commercial hypervisor, Xen, and KVM. We found that differences in the workload type (CPU or I/O intensive), workload size and VM placement yielded significant performance differences among the hypervisors. In our study, we used the three hypervisors to run several MapReduce benchmarks such as Word Count, TestDSFIO, and TeraSort and further validated our observed hypothesis using microbenchmarks.

We observed for CPU-bound benchmark, the performance difference between the three hypervisors was negligible; however, significant performance variations were seen for I/O-bound benchmarks. Moreover, adding more virtual machines on the same physical host degraded the performance on all three hypervisors, yet we observed different degradation trends amongst them. Concretely, the commercial hypervisor is 46% faster at TestDFSIO Write than KVM, but 49% slower in the TeraSort benchmark. In addition, increasing the workload size for TeraSort yielded completion times for CVM that were two times that of Xen and KVM. The performance differences shown between the hypervisors suggests that further analysis and consideration of hypervisors is needed in the future when deploying applications to cloud environments [11].

This paper present the design of a performance isolation benchmark that quantifies the degree to which a virtualization system limits the impact of a misbehaving virtual machine on other well-behaving virtual machines running on the same physical machine. test suite includes six different stress tests - a CPU intensive test, a memory intensive test, a disk intensive test, two network intensive tests (send and receive) and a fork bomb. We describe the design of our benchmark suite and present results of testing three flavors of virtualization systems –an example of full virtualization (VMware Workstation), an example of Paravirtualization (Xen) and two examples of operating system level virtualization (Solaris Containers and OpenVZ). We find that the full virtualization system offers complete isolation in all cases and that the paravirtualization system offers nearly the same benefits – no degradation in many cases with at most 1.7% degradation in the disk intensive test. The results for operating system level virtualization systems are varied – illustrating the complexity of achieving isolation of all resources in a tightly coupled system. Our results highlight the difference between these classes of virtualization systems as well as the importance of considering multiple categories of resource consumption when Evaluating the performance isolation properties of a virtualization system[12].

Virtualization has become a popular way to make more efficient use of server resources within both private data centers and public cloud platforms. While recent advances in CPU architectures and new virtualization techniques have reduced the performance cost of using virtualization, overheads still exist, particularly when multiple virtual machines are competing for resources. We have performed an extensive performance comparison under hardware-assisted virtualization settings considering four popular virtualization platforms, Hyper-V, KVM, vSphere and Xen, and find that the overheads incurred by each hypervisor can vary significantly depending on the type of application and the resources assigned to it. We also find dramatic differences in the performance isolation provided by different hypervisors. However, we find no single hypervisor always outperforms the others. This suggests that effectively managing hypervisor diversity in order to match applications to the best platform is an important [13].

3.1 Overview

In this chapter the experiment and environment are focused in detail. Also contain information carried out and discus the steps in the way this research come out.
The result of all experiments are documented and there are chart to highlight on the differences in executions.
I have two servers, the first for VMware esxi and the other for xen hypervisor. These server can be deal with by the tools. Each one has the specific tool VSPHERE this for VMware and CITRIX this for XEN system. These tools located on the same pc.

3.2 Experimental Environment

The experiment is based on the hardware and software features as follows:

- **Hardware**
  - **System info :**
    - MANFACTURER HEWLET-PACKARD
    - MODEL HP XW4400 WORKDTATION
  - **STORAGE**
    - 360 GB
  - **Processor**
    - TYPE INTEL® core ™2 CPU
    - SPEED 2.66 GH ˜2.7 GH
  - **NETWORK (SWITCHES, CAPLES).**
  - **RAM**
    - TYPE DDR
    - SIZE 4096 MB

- **Software**
    - VMWARE SERVER ESXI 5
    - VSPHERE (VMWARE CLIENT).
    - XEN HYPERVISOR 6.1
    - XEN CLIENT
    - WINDOWS 7.
    - LINUX UBUNTU SERVER 13.4
    - DATABASE POSTGRES 9.1
    - SCHEMA (from Electricity Company).
    - Pgadmin ||| (to access the postgres database)

3.3 INSTALL SYSTEM PROGRAMS

- INSTALL VMWARE ESXI

I install VMware esxi from bootable CD. Once system set up on the machine now just set the network configuration (static or dynamic Ip address) and password.

- INSTALL VSPHERE

Once the ESXI installed to communicate with server there is need to API program to access server resources. This API is VSPHERE provided by VMware. After installing Vsphere on windows 7 to maintain the server (this is entrance to administrates the server virtual machine create, delete, maintain etc).

Then create new machine here you can add or remove hardware like usb and cdroom. Then determine the disk space you need for this vm, determine the memory. After machine created power it on and attach it with physical cd room witch in the pc client side. This step let vm boot from the cd room witch containing the Linux UBUNTU edition and Linux installation is normal and take option as compatible with job this system made to.

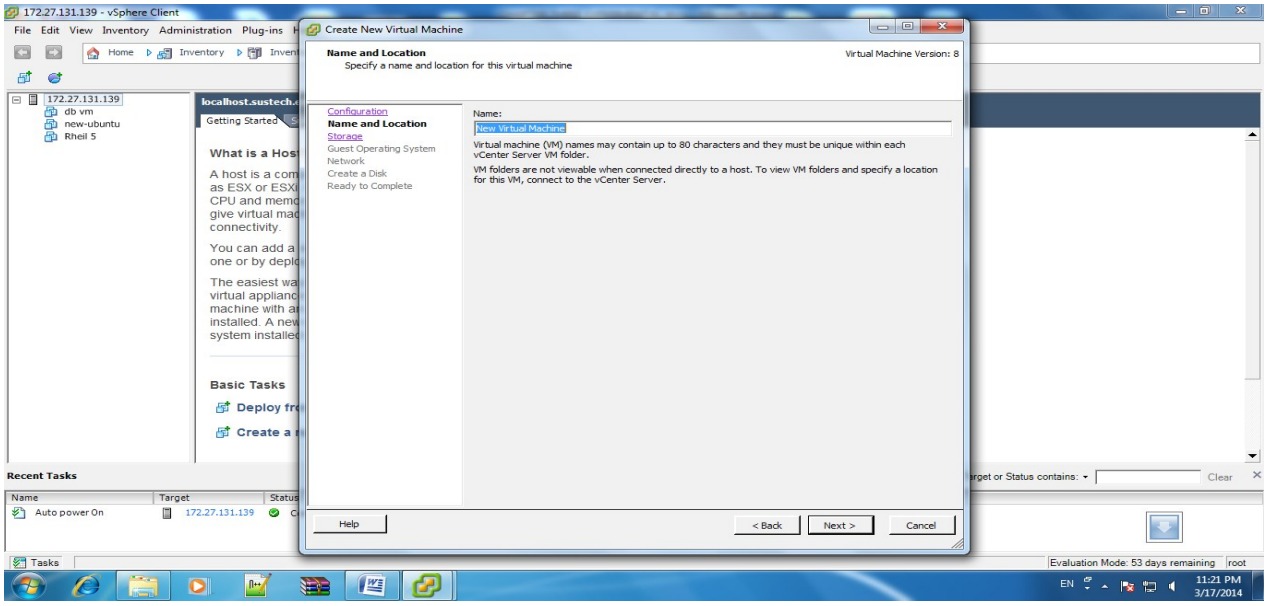

Figure 3.1: Vsphere create VM

- **INSTALL XEN HYPERVISOR**

I did same steps when installing VMWare server to install XEN HYPERVISOR first install the server witch provide the resources to users whom need to use this resources. Boot from the CD that containing the XEN HYPERVISOR source and install it in the computer. Here it similar to install VMWARE ESXI.

- **INSTALLING CITRIX**

Here I need to get inside the server to control and administrate server In the way to use server resources I need to install Citrix (Xen Api client side) which allow me to make new machine , delete, alter, start, and stop the machine. After this step creating new machine is possible.

Creating machine by using create new machine and set the memory size and hard disk space. Then choose the OS type and distribution. To install OS on the new virtual machine put the OS cd in the XEN HYPERVISOE server drive. After OS set up all the system now is working.

- **INSTALL POSTGRES**

When all system set up now I need to install POSTGRES. This step began earlier within installing UBUNTU when installation asking for the software you want this system include or the services this system should do like ftp server, POSTGRES, ssh, ..Etc. Check on POSTGRES and SSH.

Then login the system and create account (role name) and database. Import the schema into your database.

The command I used to install postgresql was

[elmahi@ubuntu /]# apt-get install postgresql

Then create username and database and give all privileges to user

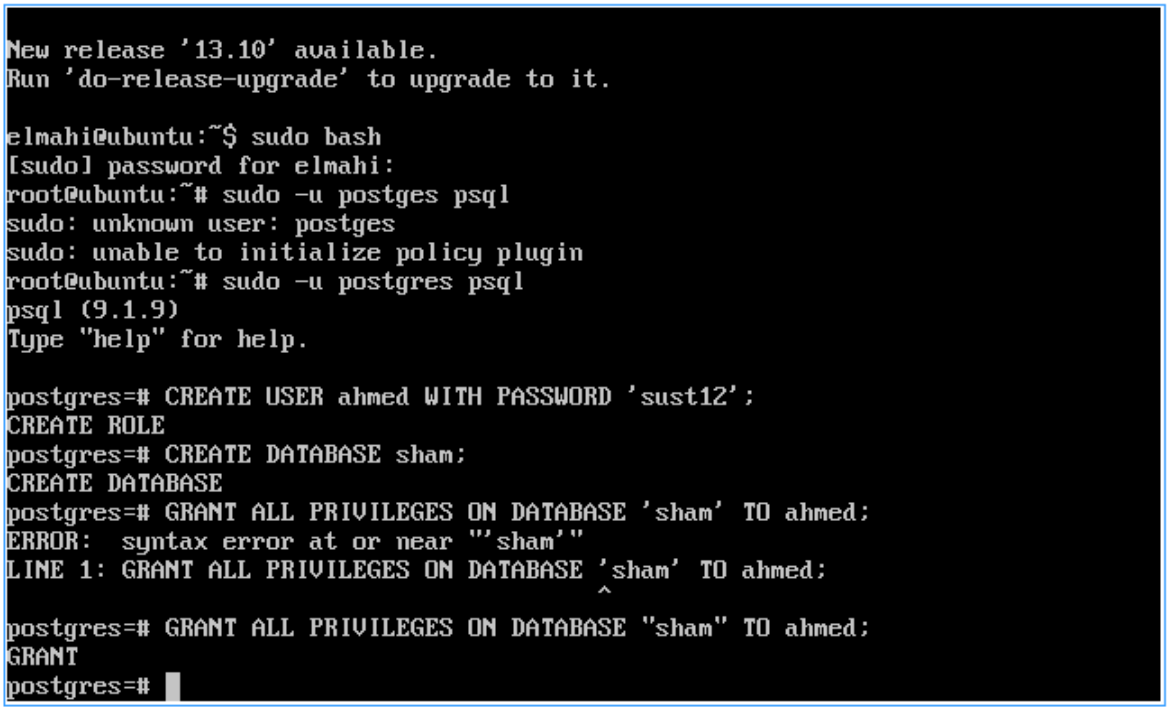

Figure 3.2: install postgres

3.4 DATABASE MANAGEMENT SYSTEM (DBMS)

I used postgresql 9.1 as database management system because the data format used in these experiments is ARP. Postgresql 9.1 open source object-oriented system. I used it to store the schema and save data that used in experiments.

3. 5 EXPERIMENT METHODOLOGY

The database source that used to do this research it belong to ERP system of Ministry of water resources and electricity. it is Postgres database containing field hold all information about the employees (period, account bank statement, moves ant etc) beside information about asset that belong to company.

3.6 SCHEMA

Here I illustrate the schema of the whole database

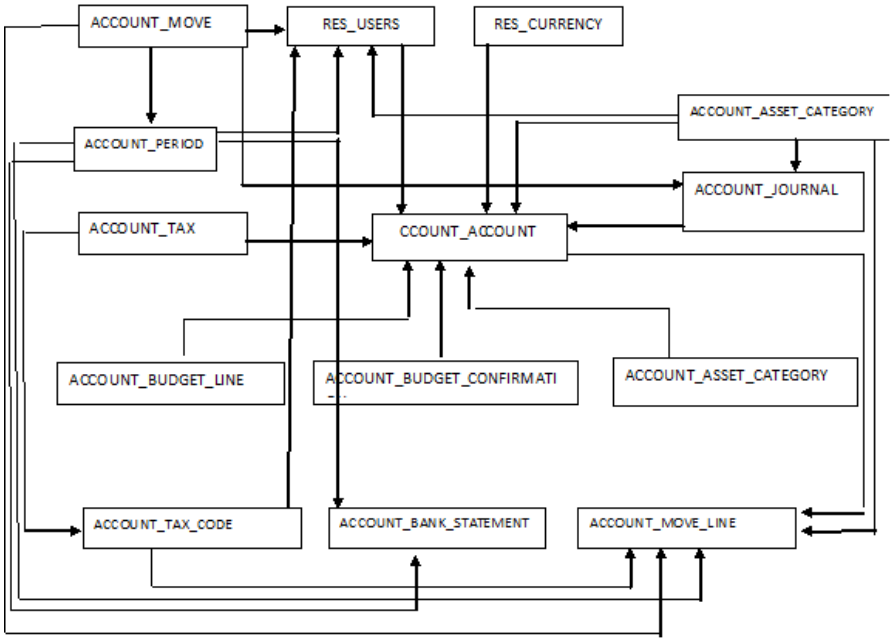

Figure 3.3: schema

3.7 TABLES SIZES

I choose five tables to do experiment, every experiment I did it six times on every table. I choose the tables according to size, began with smaller size and take bigger size due to formula this formula approximately:

NO	TABLE NAME	SIZE
1	ACCOUNT_ANALYTIC_ACCOUNT	335
2	PRODUCT_PRODUCT	3862
3	ACCOUNT_MOVE_RECONCILE	57435
4	ACCOUNT_BUDGET_LINE	94458
5	ACCOUNT_MOVE_LINE	1144310

Table 3.1 sizes of chosen table

- **CONNECTING TO DATABASE USING PGADMIN |||**

To connect database first attach to server. This step need special configuration , authentication process in the way to make security system valid.

In the beginning identify the server name and ip address then username and password Ithose I make

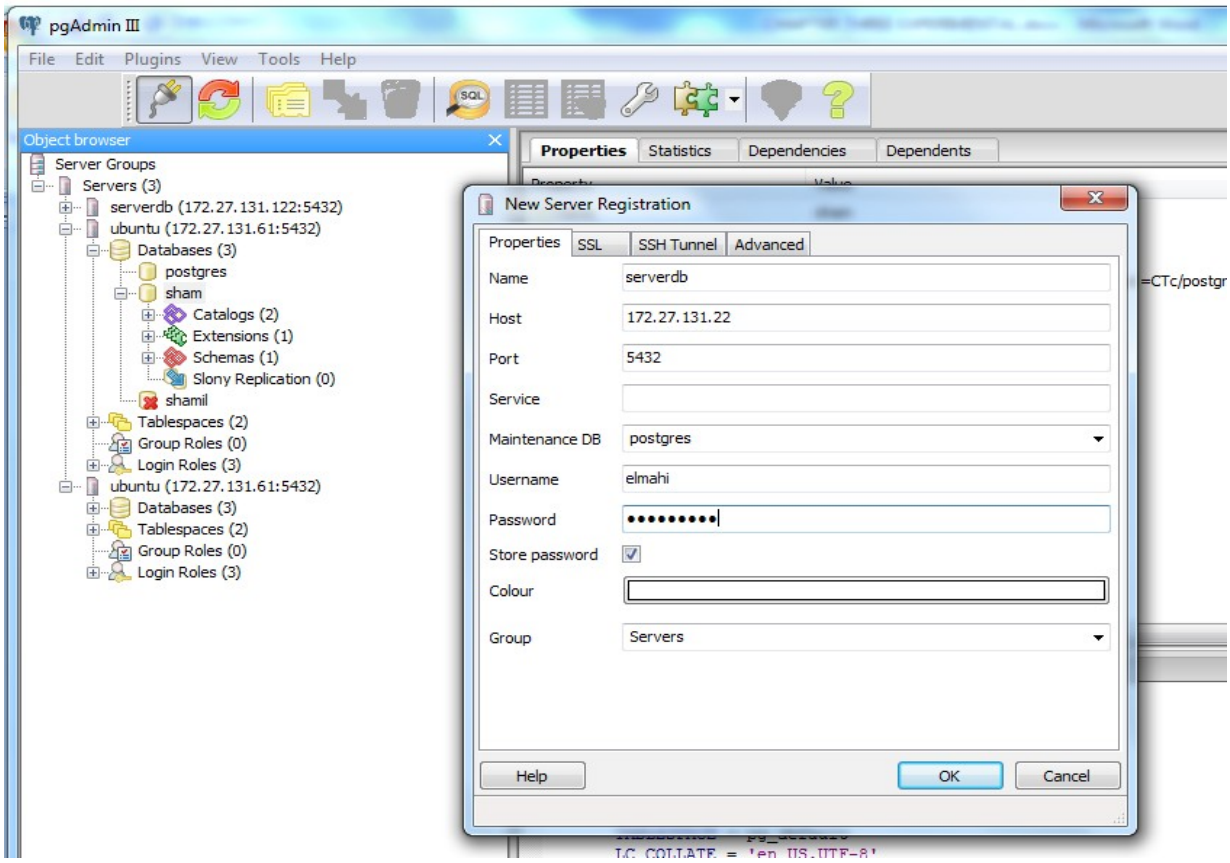

Figure 3.4: PG ADMIN

First there are configurations to allow specific host or network(sub) to access this database If I don't configure the system to accept connection I will face authentication refused problem and I solved it in the following step

[elmahi@ubuntu /]# nano /etc/postgresql/9.1/main/pg-hba.conf

In database administration there is ip v4 local connections I put the network that I used to connect to this server 172.27.130.0./23.

[elmahi@ubuntu /]# nano /etc/postgresql/9.1/main/postgresql.conf

In connection and administration there listen_addresses= 'localhost' here set local host to '*' this means listen to all as figure (3.7) shown.

Now the system ready to connect database with pgadmin (Api).. All databases now appear on pgadmin screen I choosed the database that I worked on, all tables as shown in figure (3.8)

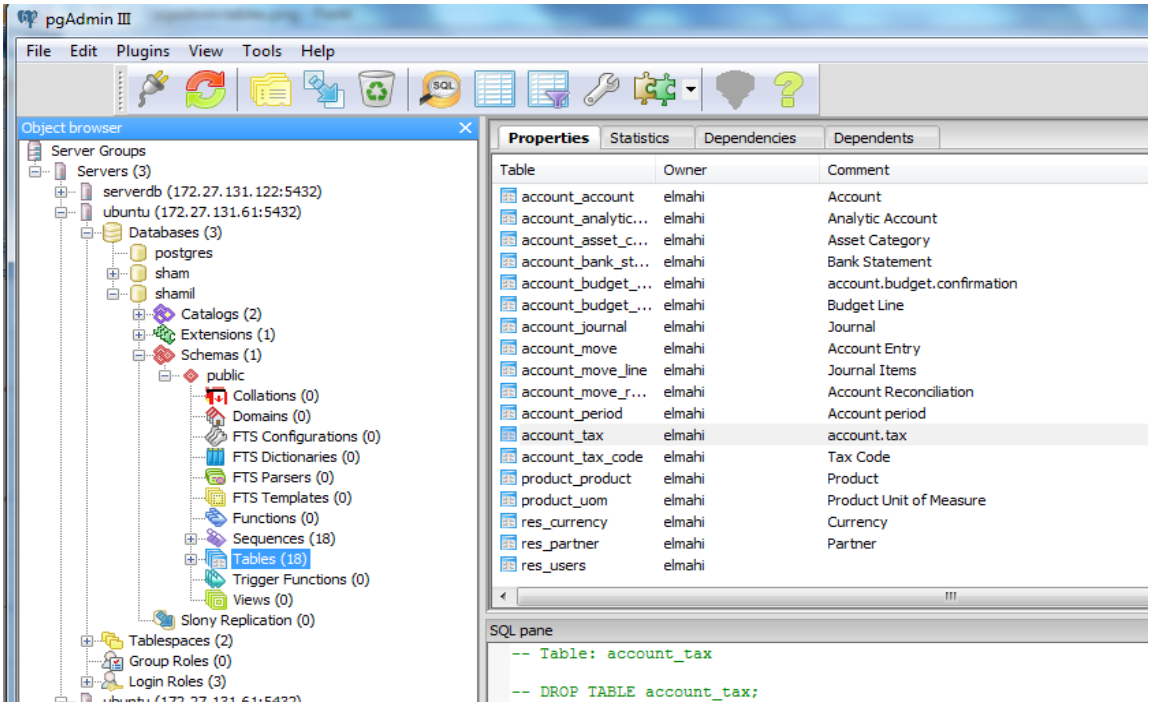

Figure3.5: TABLES IN PG ADMIN

3.8 EXPERIMENT METHODOLOGY

I had use tow different of query to measures memory usage and the other to measures CPU utilization.

3.8.1 Memory Usage Queries

First I test memory usage criteria and I did this by these queries which retrieve huge data from data base

- Select *from ACCOUNT_ANALYTIC_ACCOUNT .
- Select *from PRODUCT_PRODUCT .
- Select *from ACCOUNT_MOVE_RECONCILE .
- Select *from ACCOUNT_BUDGET_LINE .
- Select *from ACCOUNT_MOVE_LINE .

3.8.2 CPU Utilization Queries

This queries aims to measures performance of CPU . so it complicated to put processor under pressures.

- SELECT * FROM public. ACCOUNT_ANALYTIC_ACCOUNT, public.res_users

  WHERE ACCOUNT_ANALYTIC_ACCOUNT = res_users.id;

- SELECT * FROM public.product_product, public.res_users

  WHERE product_product.id = res_users.id ;

- SELECT * FROM public.account_move_reconcile, public.res_users

  WHERE account_move_reconcile.create_uid = res_users.create_uid;

- SELECT * FROM public.account_account, public.account_budget_confirmation

  WHERE account_account.create_uid = account_budget_confirmation.create_uid;

- SELECT * FROM public.account_move, public.account_move_line

  WHERE account_move.id = account_move_line.id;

- **Experiment To Measures Memory Usage**

In this experiment I execute query and register the usage of memory consumed by the execution query to accomplish the job and gives the result. I did this six times on every table and take the medium of every guery .

- **Experiment To Measures CPU Utilization**

In this experiment I execute query and register the usage of CPU occupy by the query to accomplish the job and gives the result. I took the result by percentage of CPU usage I did this six times on every table and take the medium of every query usage.

4.1 Over view

As I mentioned before this experiment aims to clarify the advantages of using to virtual machines. And I focus on CPU usage; I mean percentage usage that the processors need to accomplish the job assigned to it. And total of time taken to accomplish the job. This measured in mille second.
I used the same query to take readings in both experiment six times.

4.2 Memory Usage

This part of experiment to test the time taken by two systems(vmware and xen hyperviser) to finish the job assigned to them in the way to measure the efficiency of them. I did this by run the query and waiting until the job finish. There is tool to measure this time.

4.2.1 VMWARE Memory Usage

TABLES	VMWARE Memory Usage						
ACCOUNT_ANALYTIC_ACCOUNT	NO	1	2	3	4	5	6
	RUN TIME	110	111	112	112	111	114
	MEDIUM	111.67					
PRODUCT_PRODUCT	NO	1	2	3	4	5	6
	RUNTIME	943	883	942	882	882	883.
	MEDIUM	902.5					
ACCOUNT_MOVE_RECONCILE	NO	1	2	3	4	5	6
	RUNTIME	5348	5503	5345	5387	5398	5385
	MEDIUM	5394.3					
ACCOUNT_BUDGET_LINE	NO	1	2	3	4	5	6
	RUNTIME	11014	10938	11173	10856	10871	10871
	MEDIUM	10953.83					
ACCOUNT_MOVE_LIN	NO	1	2	3	4	5	6
	RUN TIME	502424.	504328	617027	709226	674986	716822
	MEDIUM	620802.17					

Table 4.1: VMWARE Memory Usage Result

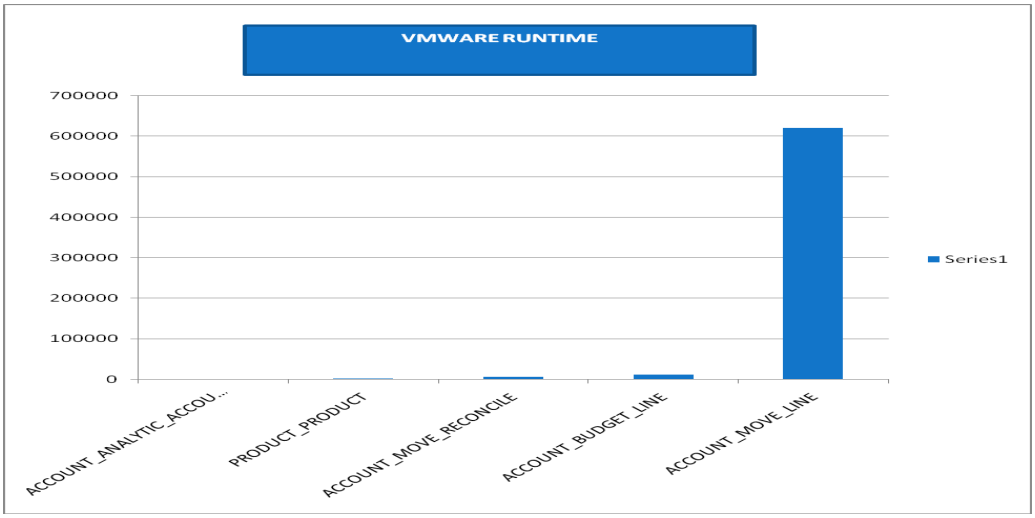

Figure 4.1 VMWARE Memory Usage Result Graph

4.2.2 XEN HYPERVISOR Memory Usage

This table shows result of XEN HYPERVISOR experiment Memory Usage

TABLES	XEN HYPERVISOR Memory Usage						
ACCOUNT_ANALYTIC_ACCOUNT	NO	1	2	3	4	5	6
	RUN TIME	218	171	172	187	171	288
	MEDIUM	201.17					
PRODUCT_PRODUCT	NO	1	2	3	4	5	6
	RUNTIME	1373	1373	1373	1373	1373	882
	MEDIUM	1291.17					
ACCOUNT_MOVE_RECONCILE	NO	1	2	3	4	5	6
	RUNTIME	8299	8299	8299	8315	8300	8315
	MEDIUM	8304.5					
ACCOUNT_BUDGET_LINE	NO	1	2	3	4	5	6
	RUNTIME	16848	16852	11125	10966	10839	10826
	MEDIUM	12909.33					
ACCOUNT_MOVE_LIN	NO	1	2	3	4	5	6
	RUNTIME	560589	540280	576802	569906	642307	498673
	MEDIUM	564759.5					

Table 4.2: XEN HYPERVISOR Memory Usage

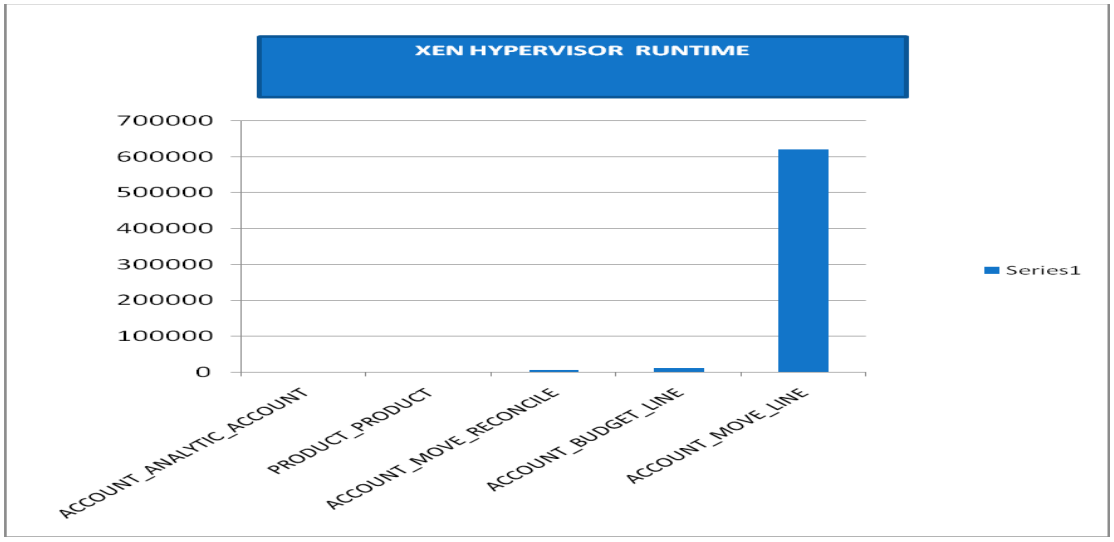

Figure 4.2:XEN HYPERVISOR Memory Usage

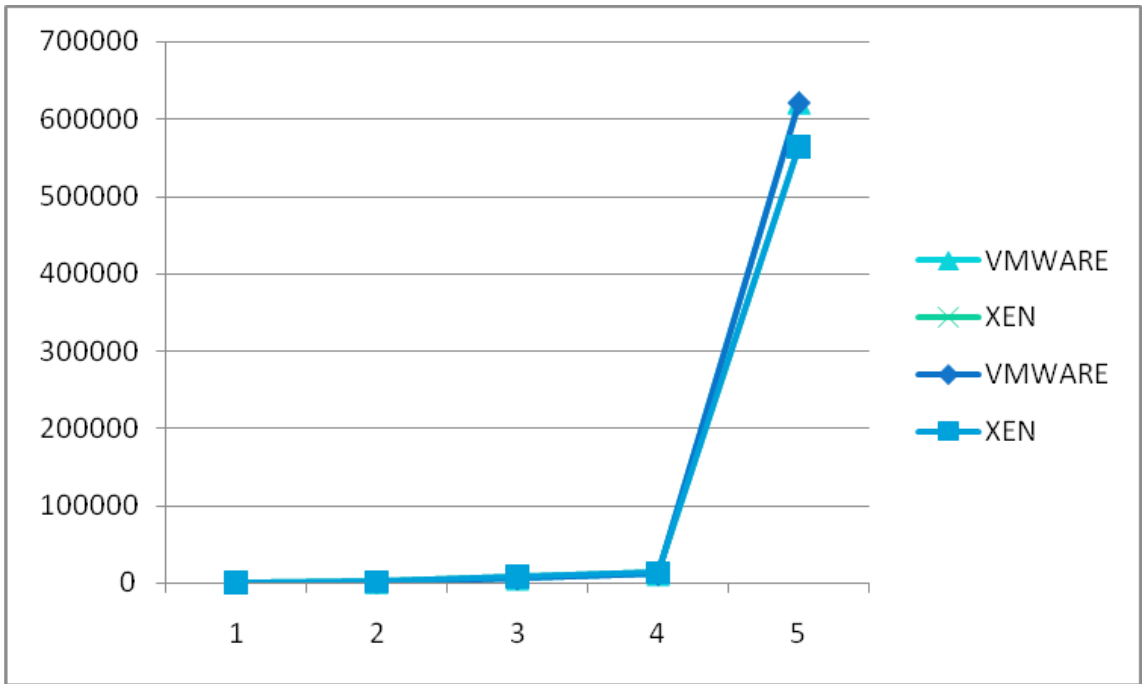

Figure 4.3: Memory Usage VMWARE v XEN

4.2.3 MEMORY USAGE ANALYSIS

As we see in the figure Figure(4.3) VMware systems finish the job in short period of time than Xen hypervisor . the latency in finishing job all mostly belonging to instruction set that used by the Xen and VMware virtualized system theses tables taken . Differences in time finishing the job is negligible. There needs to do this experiment again uses large database schema and uses another methodology of prior well testing virtual machine environment.

4.3 CPU UTILIZATION

4.3.1 VMWARE PROCESSOR UTILIZATION IN MHZ

This table shows experiment result of VMWARE PROCESSOR USAGE IN MHZ

TABLES	VMWARE PROCESSOR UTILIZATION IN MHZ						
ACCOUNT_ANALYTIC_ACCOUNT	NO	1	2	3	4	5	6
	RUN TIME/ms	0.72	0.72%	0.69%	0.73%	0.73 %	076 %
	MEDIUM	0.725%					
PRODUCT_PRODUCT	NO	1	2	3	4	5	6
	RUNTIME/ms	1.2%	1. 2%	1.11%	1.02%	1.01 %	1.23 %
	MEDIUM	1.23%					
ACCOUNT_MOVE_RECONCILE	NO	1	2	3	4	5	6
	RUNTIME/ms	5.14%	6.75%	6.33%	4.84%	5.37 %	7.11 %
	MEDIUM	5.92%					
ACCOUNT_BUDGET_LINE	NO	1	2	3	4	5	6
	RUNTIME/ms	7.68%	7.12%	7.11%	7.26%	7.62 %	7.42 %
	MEDIUM	7.37%					
ACCOUNT_MOVE_LIN	NO	1	2	3	4	5	6
	RUN TIME/ms	11.89	11.64%	12.48%	11.87 %	12.09 %	10.87 %
	MEDIUM	11.80%					

Table 4.3: VMWARE PROCESSOR UTILIZATION IN MHZ

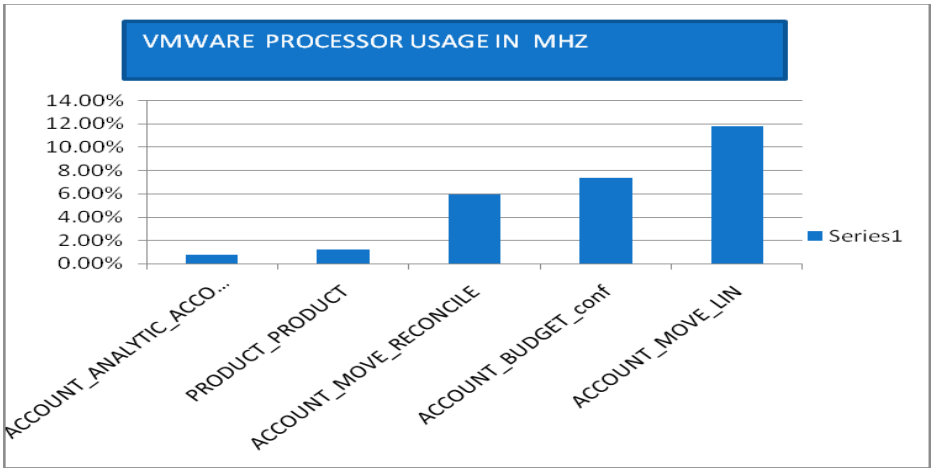

Figure 4.4: VMWARE PROCESSOR UTILIZATION IN MHZ

4.3.2 XEN HYPERVISOR PROCESSOR UTILIZATION

This table shows experiment result of XEN HYPERVISOR processor utilization inMHZ

TABLES	XEN HYPERVISOR PROCESSOR UTILIZATION IN MHZ						
ACCOUNT_ANALYTIC_ACCOUNT	NO	1	2	3	4	5	6
	RUN TIME/ms	0.33%	0.32%	0.32%	0.34%	0.34 %	0.37 %
	MEDIUM	0.29%					
PRODUCT_PRODUCT	NO	1	2	3	4	5	6
	RUNTIME/ms	0.5%	0.49%	0.47%	0.49%	0.46 %	0.51 %
	MEDIUM	0.47%					
ACCOUNT_MOVE_RECONCILE	NO	1	2	3	4	5	6
	RUNTIME/ms	2.35%	2.69%	2.74%	2.78%	2.69 %	2.72 %
	MEDIUM	2.66%					
ACCOUNT_BUDGET_LINE	NO	1	2	3	4	5	6
	RUNTIME/ms	6.34%	6.28%	5.33%	5.08%	5.62 %	4.49 %
	MEDIUM	5.52%					
ACCOUNT_MOVE_LIN	NO	1	2	3	4	5	6
	RUN TIME/ms	7.94%	9.66%	8.78%	7.18%	7.83 %	7.68 %
	MEDIUM	8.18%					

Table 4.4: XENPROCESSOR UTILIZATION IN MHZ

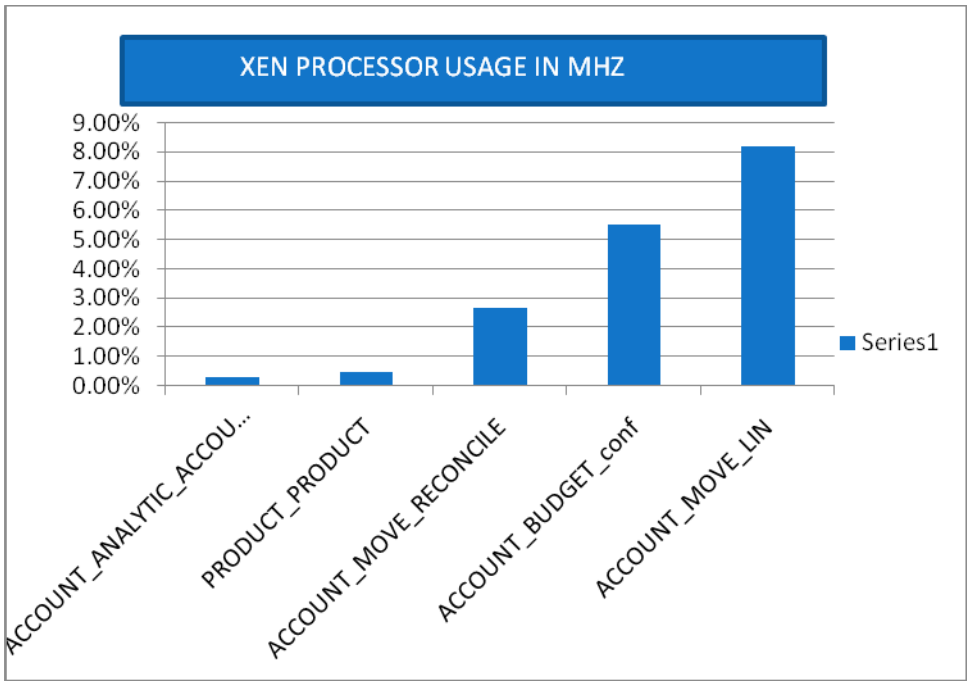

Figure4.5: XEN PROCESSOR UTILIZATION GRAPH

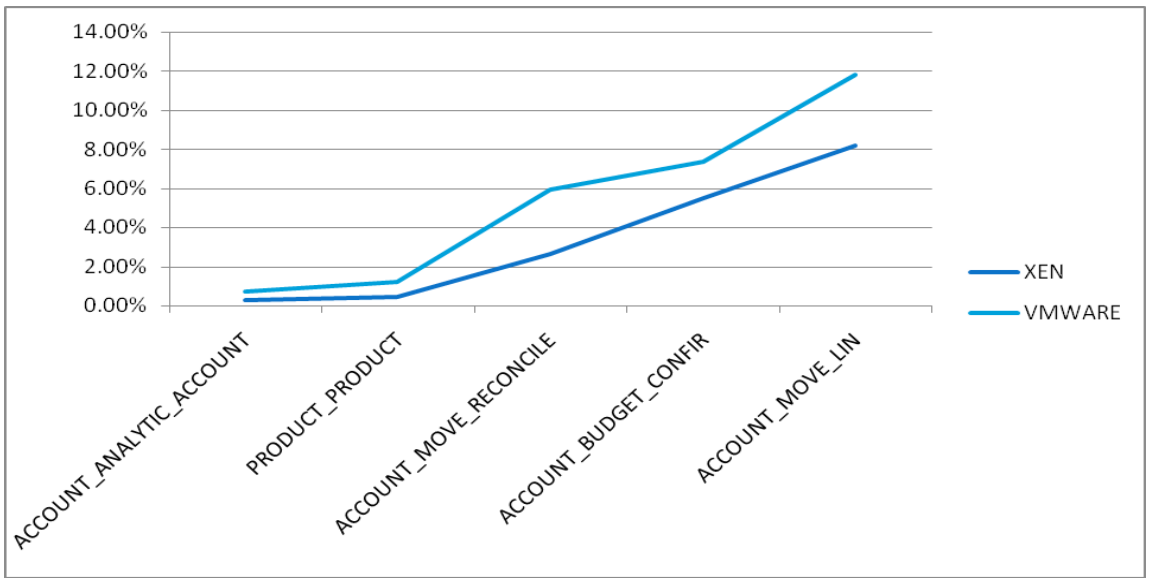

Figure 4.6: PROCESSOR USAGE VMWARE v XEN

4.3.3 PROCESSOR UTILIZATION ANALYSIS

This measures shows that XEN is best in performance . it use less cpu effort than VMWARE . specially in the query that retrieve little bytes of data.
And the query that retrieve huge amount of data also xen do better performance.

Here in this figure shows that when VMware in the idle terms the processor usage reads over 0% of usage, means while the Xen processor usage goes to 0% when there is no activity, this shown in the figure (4.5) .

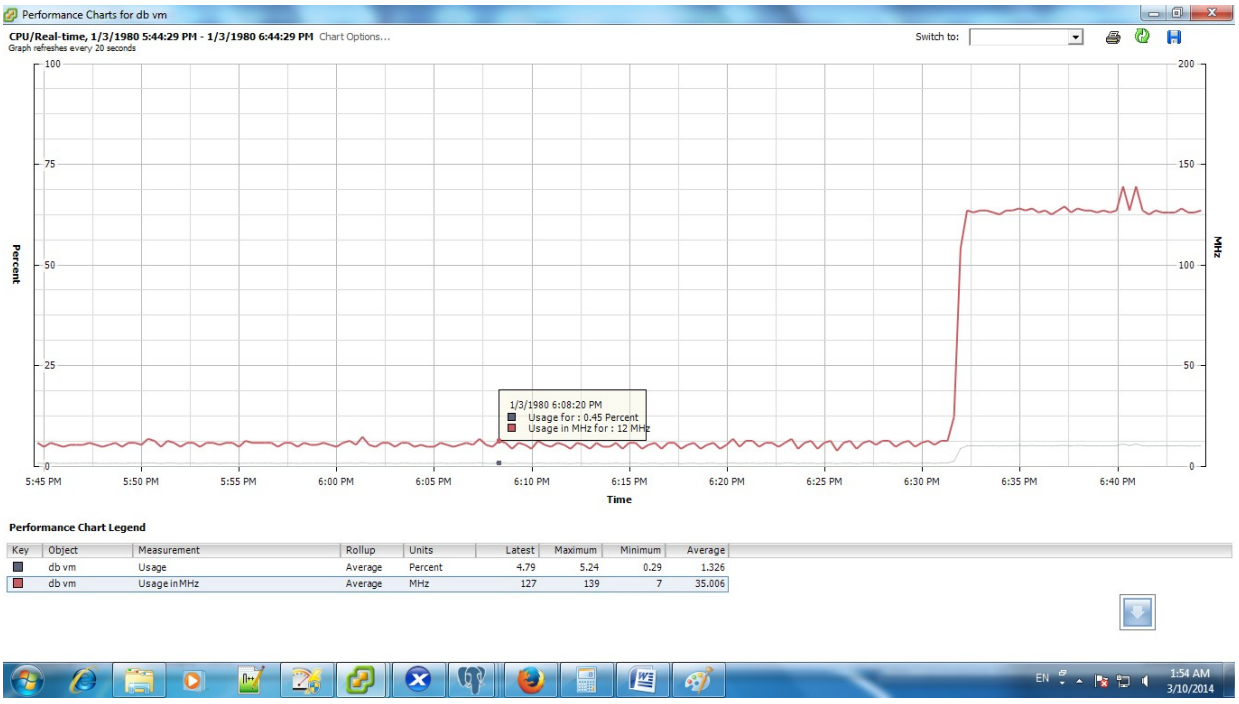

Figure 4.7: VMWARE PROCESSOR IN IDLE TERM

5.1 CONCLUSION

This research aims to spot light of open source virtual machine XEN hypervisor Versace commercial VMware to see if this open source virtual machine is good enough to do the jobs as same as commercial.

the comparison done on two criteria, first comparing memory usage between two virtual machines VMware systems finish the job in short period time than Xen hypervisor

The second criteria CPU usage, every process needs part of CPU to make the process finish. By running same process on tow virtual machines I take result in the usage by percentage .

This research shows that this open source virtual machine did the job as same as commercial virtual machine and sometimes it is best in performance.

5.2 FUTURE WORK

Future work under consideration includes more subjective tests covering a wider area of criteria and following up the newest version of virtual machines. Besides do these test on bigger size of database schema. The future tests will also use tow kind of guest operating systems 64 and 32 bit, test the latency and network traffic utilization.

REFERENCES

[1] Upper Canada the chartered institute for it
http://home.bcsuppercanada.ca/app/Publish/pubEngine.php?st=34&dom=Editorial

[2] http://docs.oracle.com/cd/E26996_01/E18549/html/VMUSG1010.html.

[3] http://www.dcseurope.info/section.php?section=29#ixzz32WZEKd00

[4] http://www.scribd.com/doc/112245294/Virtual-Machines-High-impact-Strategies-What-You-Need-to-Know-Definitions-Adoptions-Impact-Benefits-Maturity-Vendors

[5] http://searchservervirtualization.techtarget.com/definition/paravirtualization

[6] Eli: bare-metal performance for i/o virtualization
http://www.cs.columbia.edu/search?cx=017181709448284692436%3Acjy5gsheid4&cof=FORID%3A11&ie=UTF-8&q=eli%3A+bare-metal+performance+for+i%2Fo+virtualization&sa=GO:

[7] www.packtpub.com›NewsCenter
Cookbookshttp://www.cio.com/article/40701/Virtualization_Definition_and_Solutions

[8] community.spiceworks.com/product/33810-vmware-esx-server

[9] http://xenserver.org/component/easyblog/blogger/listings/jamesbu.html?Itemid=179

[*10*] Todd Deshane, Demetrios Dimatos, Gary Hamilton, Madhujith Hapuarachchi, Wenjin Hu, Michael McCabe, Jeanna Neefe Matthews, Clarkson University "Performance Isolation of a Misbehaving Virtual Machine with Xen, VMware and Solaris Containers"

[11] Jack Li, Qingyang Wang, Deepal Jayasinghe, Junhee Park, Tao Zhu, Calton Pu "School of Computer Science at the College of Computing Georgia Institute of Technology Atlanta, Georgia 30332–0250" "Performance Overhead Among Three Hypervisors: An Experimental Study using Hadoop Benchmarks"

[12] Jeanna Neefe Matthews, Wenjin Hu, Madhujith Hapuarachchi, Todd Deshane, Demetrios Dimatos, Gary Hamilton, Michael McCabe, James Owens Clarkson University "Quantifying the Performance Isolation Properties of Virtualization Systems"

[13] Jinho Hwang "The George Washington University jinho10@gwu.edu ", Sai Zeng and Frederick y wu "IBM T. J. Watson Research Center {saizeng, fywu}@us.ibm.com" , "Timothy Wood "The George Washington University timwood@gwu.edu" "A Component-Based Performance Comparison of Four Hypervisors"