بسم الله الرحمن الرحيم

Sudan University of Science and Technology

College of Graduate Studies

College of Computer science and Information Technology

A comparison Study between Virtualized and non-virtualized Environments

Case study: Xen® hypervisor

دراسة مقارنة بين البيئات الإفتراضية وغير الإفتراضية

Xen® hypervisor:دراسة حالة

By: Mohammed Siddig Ahmed.

Supervisor: Dr.Talaat wahby.

May 2014.

Acknowledgment

I would like to thank my supervisor, Dr. Talaat wahby

For the advices he has provided throughout the research time.

I would also like to thank My Family and My Colleagues.

ABSTRACT

Hardware Virtualization has become an important technology nowadays because it provides excellent technical features for the present as well as the future through the provision of significant advantages for instance saving the Hardware expenses.

The research aims to compare Virtualized Environments to non-virtualized environments, and the Xen Technology which is chosen in this research is one of the leading technologies in the hardware virtualization field. The comparison based on two pivots, the first measures the Memory performance (RAM) and the second measures the processor performance, to obtain results that reflect the real difference between the two environments performances that used the same hardware specifications and the same size and structure of data, this data is part of ERP system that belong to one of Sudanese companies.

The results that we have acquired reflect the substantial convergence and very small differences in performance between the Virtualized Environments and Non-visualized Environments, and beyond that the research found that in case of large data the virtualized Environments give better performance results than non-virtual environments that is due to tuning and better management provided by virtualization tools.

المستخلص

تعتبر تقنية الـ Hardware Virtualization من التقنية المهمة في الاونة الاخيرة واصبحت تخطو بخطوات ثابتة لرسم ملامح التقنية في الحاضر والمستقبل بما توفره من مزايا كبيرة وبما توفره من اموال كانت تنفق لشراء الـHardware.

تم في هذا البحث عقد مقارنة بين Virtualized Environments و Non-visualized Environments وقد استخدمنا احدي التقنيات الرائدة في هذا المجال وهي ال Xen Technology . وترتكز المقارنة علي محورين الاول هو مقارنة الاداء بالنسبة لزاكرة الوصول العشوائي (RAM) والثاني هو مقارنة الاداء لوحدة المعالجة المركزية (CPU) , وللحصول علي نتائج تعكس الفرق بين البيئتين بصورة حقيقية تم استخدام مواصفات Hardware متماثلة لكلا البيئتين كما تم استخدام نفس البيانات وبنفس الحجم لاجراء التجارب وتمثل البيانات المستخدمة في التجارب حصيلة العمل لمدة ثلاثة سنوات بنظام ادارة المؤسسة لاحدي الشركات السودانية.

النتائج التي تحصلنا عليها تعكس عن تقارب كبير وعن فروقات ضئلة جدا في الاداء بين Virtualized Environmentsو Non-visualized Environments وابعد من ذلك توصل البحث الي انه في بعض الحالات تعطي ال Virtualized Environments نتائج افضل في الاداء وهذا ماظهر في البيانات ذات الحجم الكبير وهذا بسبب الادارة الجيدة التي توفرها ادوات الـ Virtualization للذاكرة وللمعالج.

Table of Contents

List of Figures

CHAPTER ONE

INTRODUCTION

1. Introduction

1.1. Overview

The concept of virtualization is generally believed to have its origins in the mainframe days in the late 1960s and early 1970s, when IBM invested a lot of time and effort in developing robust time-sharing solutions. 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 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.

Data centers today use virtualization techniques to make abstraction of the physical hardware, create large aggregated pools of logical resources consisting of CPUs, memory, disks, file storage, applications, networking, and offer those resources to users or customers in the form of agile, scalable, consolidated virtual machines. Even though the technology and use cases have evolved, the core meaning of virtualization remains the same: to enable a computing environment to run multiple independent systems at the same time.

Virtualization is commonly applied to physical hardware resources by combining multiple physical resources into shared pools from which users receive virtual resources. With virtualization, you can make one physical resource look like multiple virtual resources. Virtual resources can have functions or features that are not available in their underlying physical resources.

Virtualization provides the following benefits:

- Consolidation to reduce hardware cost:
    - Virtualization enables you to efficiently access and manage resources to reduce operations and systems management costs while maintaining needed capacity.
    - Virtualization enables you to have a single server function as multiple virtual servers.
- Optimization of workloads:
    - Virtualization enables you to respond dynamically to the application needs of its users.
    - Virtualization can increase the use of existing resources by enabling dynamic sharing of resource pools.
- IT flexibility and responsiveness:
    - Virtualization enables you to have a single, consolidated view of, and easy access to, all available resources in the network, regardless of location.

- Virtualization enables you to reduce the management of your environment by providing emulation for compatibility, improved interoperability, and transparent change windows.

1.2. Scope of the Research

Virtualization is very large information technology area, this research is focused on Hardware virtualization excluding the Network and connectivity issues and types of virtualization, such as software virtualization. The research covers only the full-virtualization (see figure 1).

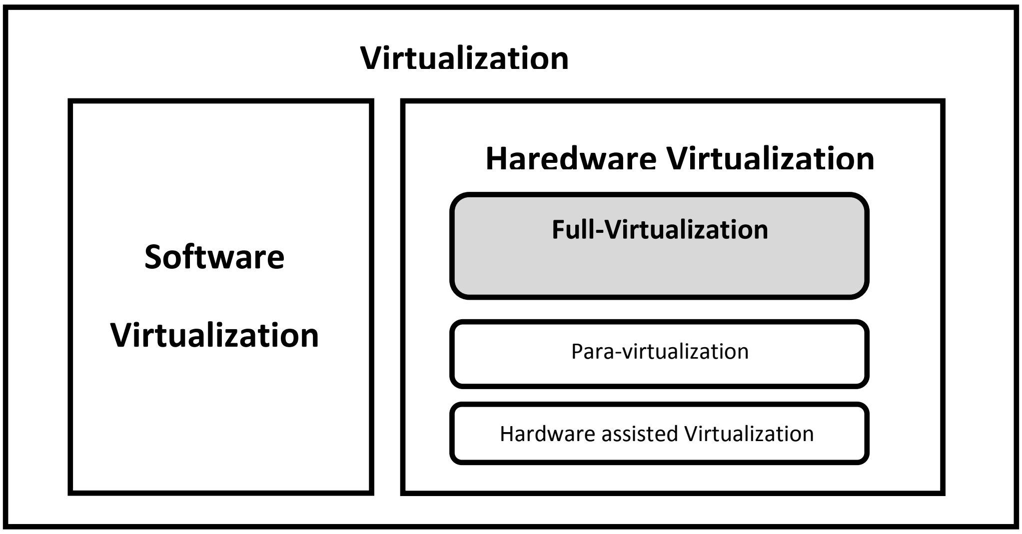


Figure 1.1: Scope of the Researches

1.3. Problem statement

The research compares the computer resource utilization in case of virtualized and none virtualized environment to prove that virtual environment is quite similar or close to the non-virtualized environment., these similarities or comparison can be summarized into the following : -

1.3.1. Memory usage.
1.3.2.CPU utilization.
1.3.3.Both of Memory Usage and CPU utilization at same time.

1.4. Questions of the research

The task of this research is to answer the following questions:

- Is the performance of Virtual environment is quite similar or close to the non-virtualized environment?
- Can we replace non-virtualized environment with Virtual environment without problems?
- Is the open-source virtualization technologies matured enough?

1.5. Objectives of the research

The main objective of this research is to prove that the Virtual environment performance is quite similar to the non-virtualized environment; in addition research aims to prove the reliability of virtualization technology.
Furthermore the Research tries to achieve the following objectives:-

- Use a technology that reduces the cost of Hardware and software in IT industry.
- Be familiar with open-source technologies to avoid getting commercial software licenses which are sometimes forbidden to be used in our country for some political reasons.
- State-of-the-art of the used technology (hardware Virtualization).
- Improve the legacy systems performance.

1.6. Research methodology

The methodology used in the research is summarized in the following steps:

1- Study related work.
2- Literature review.
3- Prepare experiment environment.
4- Conduct the experiment and extract the results.
5- Analyze the experiment results.
6- Research recommendations and future works.

1.7. Research organization

The research consists of six chapters; the first chapter is an introduction chapter which overview the research then discusses the scope of the research problem, problem statement, objectives, questions and methodology of the research. Then Chapter two overviews the literature and related works. Chapter three describes the research experiment in details, the description include the hardware and software used on the experiment in addition to environment of experiment. Chapter four displays the results of experiment and discusses these results. Chapter six is the last chapter, it displays the conclusion, recommendations and the future work.

CHAPTER TWO

Literature Review and Related Works

Section One: Literature review

2. Literature review and Related Works

2.1. Section One: Literature review

2.1.1. Introduction

This section reviews the concept of hardware Virtualization as general, in section 2.1 the chapter define Hardware Virtualization and then define the types of virtualization in details, after that the section display the main benefits of hardware Virtualization. section 2.3 reviews some virtualization tools which dominant the market, section 2.4 address the Xen technology in some details , it define the Xen and then displays its features and architecture, on the last of the section there is overview of XenServer™ and XenCenter™.

2.1.2. Virtualization

Definition

Computer hardware virtualizations the ability to run multiple virtual machines on a single piece of hardware, The hardware runs software which enables you to install multiple operating systems which are able to run simultaneously and independently, in their own secure environment, with minimal reduction in performance. Each virtual machine has its own virtual CPU, network interfaces, storage and operating system (Figure 1).

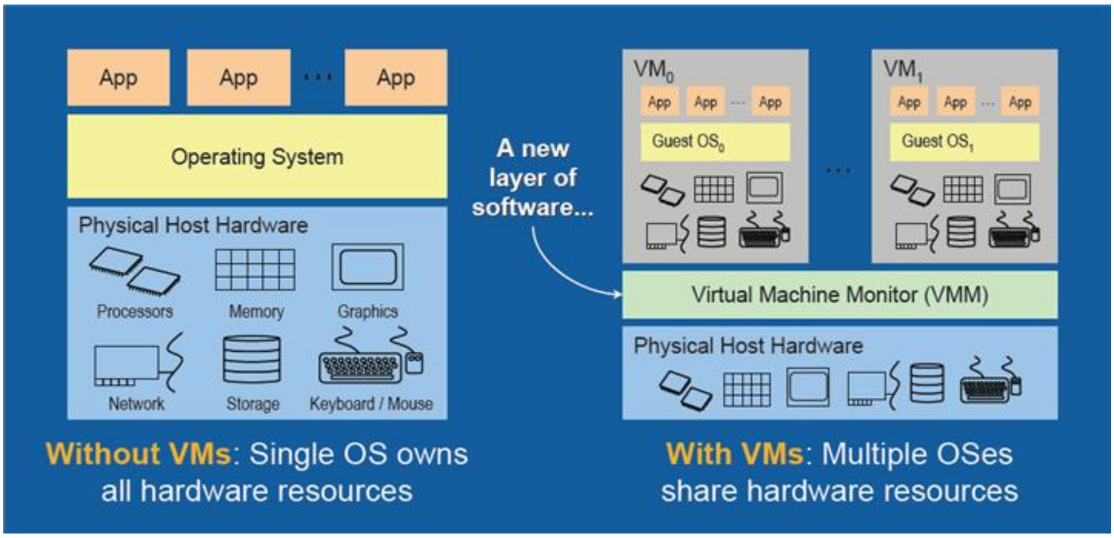


Figure 2.1 Non Virtual Machine VS.VM environment.

Types of virtualization

Various levels of hardware virtualization exist that perform various levels of abstraction:

I. **Full virtualization:-**

The guest OS is unaware that it is being virtualized. The hypervisor will handle all OS-to-hardware requests on demand and may cache the results for future use. In this instance, the virtualized OS is completely isolated from the hardware layer by the hypervisor. This provides the highest level of security and flexibility as a broader range of operating systems can be virtualized. full hardware virtualization, is the most popular and wide spread virtualization in the marketplace right now

A hypervisor, also known as a virtual machine manager/monitor (VMM), is computer hardware platform virtualization software that allows several operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and resources to itself. Instead, the hypervisor is controlling the host processor and resources, distributing what is needed to each operating system in turn and ensuring that the guest operating systems/virtual machines are unable to disrupt each other.

- **Hypervisor Classifications**

Hypervisors are classified into two types:

Bare Metal/Native Hypervisors— Software systems that run directly on the host's software as a hardware control and guest operating system monitor. A guest operating system thus runs on another level above the hypervisor. This is the classic implementation of virtual machine architectures.

Embedded/Host Hypervisors— Software applications that run within a conventional operating system environment. Considering the hypervisor layer being a distinct software layer, guest operating systems thus run at the third level above the hardware

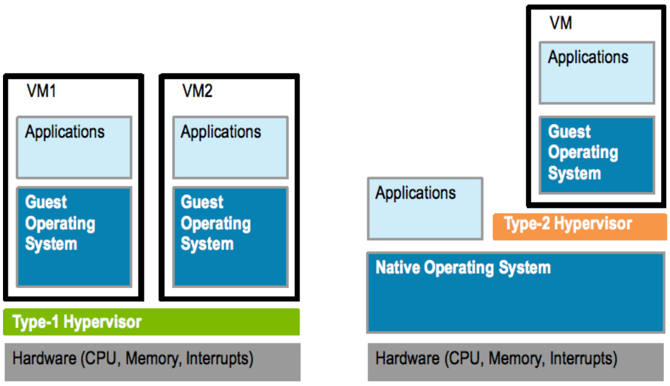


Figure 2.2. Comparisons between hypervisor design techniques

II. **Hardware assisted**:-

Hardware vendors have seen value in virtualization and have tailored their devices to enhance performance or functionality. This is most evident in the AMD-V and Intel Virtualization Technology processor enhancements. In the case of the AMD and Intel processors, specific CPU calls are not translated by the hypervisor, and are sent directly to the CPU. This reduces the hypervisor load and increases performance by removing the translation time from operating system calls.

- **AMD virtualization (AMD-V)**

  AMD-V (AMD virtualization) is a set of hardware extensions for the X86 processor architecture. Advanced Micro Dynamics (AMD) designed the extensions to perform repetitive tasks normally performed by software and improve resource use and virtual machine (VM) performance.

  Early virtualization efforts relied on software emulation to replace hardware functionality. But software emulation can be a slow and inefficient process. Because many virtualization tasks were handled through software, VM behavior and resource control were often poor, resulting in unacceptable VM performance on the server.

  Processors lacked the internal microcode to handle intensive virtualization tasks in hardware. Both Intel Corp. and AMD addressed this problem by creating processor extensions that could offload the repetitive and inefficient work from the software. By handling these tasks through processor extensions, traps and emulation of virtualization, tasks through the operating system were essentially eliminated, vastly improving VM performance on the physical server.

  AMD Virtualization (AMD-V) technology was first announced in 2004 and added to AMD's Pacifica 64-bit x86 processor designs. By 2006, AMD's Athlon 64 X2 and Athlon 64 FX processors appeared with AMD-V technology, and today, the technology is available on Turion 64 X2, second- and third-generation Opteron, Phenom and Phenom II processors.

- **Intel VT (Virtualization Technology)**

  Intel VT includes a series of extensions for hardware virtualization. The Intel VT-x extensions are probably the best recognized extensions, adding migration, priority and memory handling capabilities to a wide range of Intel processors. By comparison, the VT-d extensions add virtualization support to Intel chipsets that can assign specific I/O devices to specific virtual machines (VM) s, while the VT-c extensions bring better virtualization support to I/O devices such as network switches, in addition to Intel VT-i which is Intel Virtualization Technology for Itanium Processors.

III. **Paravirtualization (PV):-**

The guest OS needs to be engineered in such a way that it knows that it is virtualized. The kernel of the operating system is adjusted to replace instructions that cannot be virtualized with methods that interact directly with the hypervisor. Value for paravirtualized environments comes in the form of lower overhead and optimized operations. Paravirtualization is typically seen in Linux environments with the Xen kernels included, although it is more and more common to find Full Virtualization vendors, including some paravirtualization drivers, in their latest products.

2.1.2.1. Benefits of virtualization

- **Reduction of Complexity:** Infrastructure costs are reduced by removing the need for physical hardware, and networking. Instead of having a large number of physical computers, all networked together, consuming power and administration costs, fewer computers can be used to achieve the same goal. Administration and physical setup is less time consuming and costly.
- **Isolation**: Virtual machines cannot access the resources of other virtual machines. If one virtual machine performs poorly, or crashes, it does not affect any other virtual machine.
- **Platform Uniformity:** In a virtualized environment, a broad, heterogeneous array of hardware components is distilled into a uniform set of virtual devices presented to each guest operating system. This reduces the impact across the IT organization: from support, to documentation, to tools engineering.
- **Legacy Support:** With traditional operating system installations, when the hardware vendor replaces a component of a system, the operating system vendor is required to make a corresponding change to enable the new hardware (for example, an Ethernet card). As an operating system ages, the operating system vendor may no longer provide hardware enabling updates. In a virtualized operating system, the hardware remains constant for as long as the virtual environment is in place, regardless of any changes occurring in the real hardware, including full replacement.

2.1.3. Virtualization Technologies

2.1.3.1. Proxmox

Proxmox is a free, open source server virtualization product with a unique twist: It provides two virtualization solutions. It provides a full virtualization solution with Kernel-based Virtual Machine (KVM) and a container-based solution, OpenVZ.

2.1.3.2. VMware

VMware is the global leader in virtualization solutions from the desktop to the datacenter and on up to the cloud. VMware was founded in 1998 to bring the concept of virtualization to industry standard x86 systems. The concept of virtualization had been known since the 60ies mainframe world but for a long time no one was able to “crack the code” for bringing this concept to x86 systems.

2.1.3.3. Xen

Xen is capable of virtualizing x64 servers. Xen follows the Open Source (Free) licensing model and is specifically meant for Enterprise use and I choose this technology to implement experiments of this research , you can find more details about Xen technology on (figure 2.4).

2.1.4. Xen™ Technology

Xen is an open-source type-1 or bare-metal 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.

2.1.4.1. Xen key features

- Small footprint and interface (is around 1MB in size). Because Xen uses a microkernel design, with a small memory footprint and limited interface to the guest, it is more robust and secure than other hypervisors.
- Operating system agnostic: Most installations run with Linux as the main control stack (aka "domain 0"). But a number of other operating systems can be used instead, including NetBSD and OpenSolaris.
- Driver Isolation: Xen has the capability to allow the main device driver for a system to run inside of a virtual machine. If the driver crashes, or is compromised, the VM containing the driver can be rebooted and the driver restarted without affecting the rest of the system.

2.1.4.2. Xen Architecture

Below is a diagram of the Xen architecture (Figure 2.3). The 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 bootloader. There is a number of virtual machines run On top of Xen. 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.

- **Guest types**: Xen can run fully virtualized (HVM) guests, or paravirtualized (PV) guests.
- **Domain 0**: Xen has a special domain called domain 0 which contains drivers for the hardware, as well as the toolstack to control VMs.

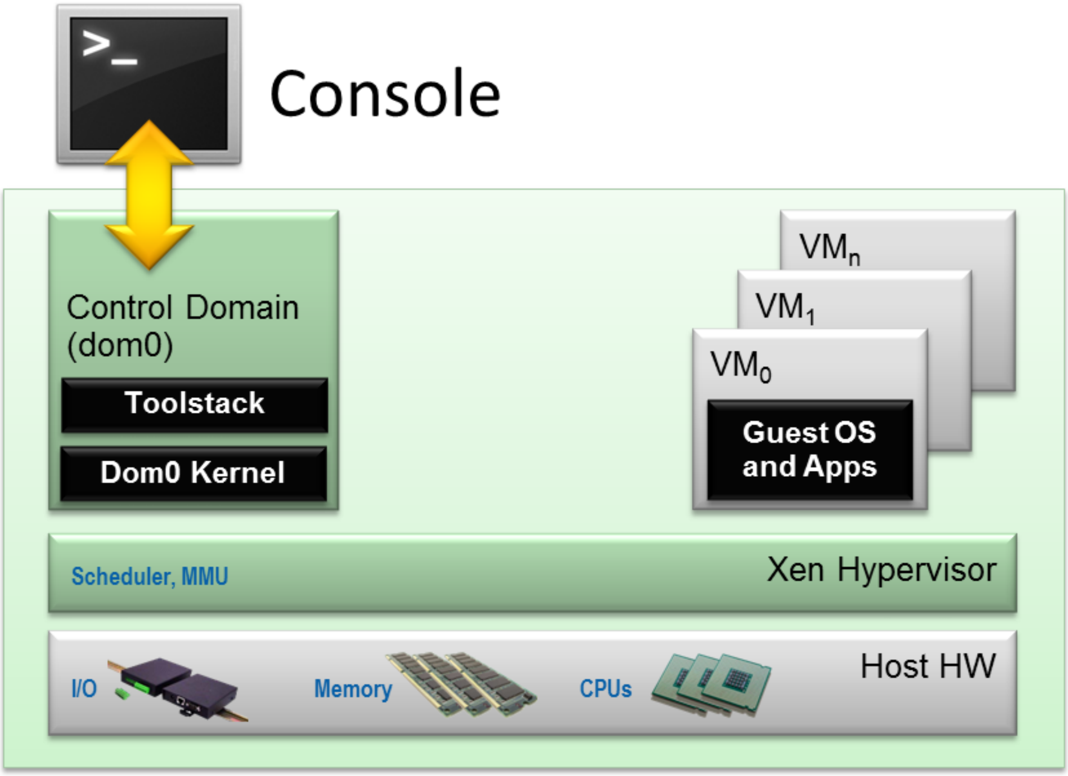


Figure 2.3 Xen architecture

- **The Xen Hypervisor**

  The Xen Hypervisor is an exceptionally lean (<150,000 lines of code) software layer that runs directly on the hardware and is responsible for managing CPU, memory, and interrupts. It is the first program running after the bootloader exits. The hypervisor itself has no knowledge of I/O functions such as networking and storage.

- **Guest Domains/Virtual Machines:**

  Guest Domains/Virtual Machines are virtualized environments, each running their own operating system and applications. Xen supports two different virtualization modes: Paravirtualization (PV) and Hardware-assisted or Full Virtualization (HVM). Both guest types can be used at the same time on a single Xen system. It is also possible to use techniques used for Paravirtualization in an HVM guest: essentially creating a continuum between PV and HVM. This approach is called PV on HVM.

Xen guests are totally isolated from the hardware: in other words, they have no privilege to access hardware or I/O functionality. Thus, they are also called unprivileged domain (or DomU).

- **Toolstack and Console:**

  Domain 0 contains a control stack (also called Toolstack) that allows a user to manage virtual machine creation, destruction, and configuration. The toolstack exposes an interface that is either driven by a command line console, by a graphical interface or by a cloud orchestration stack such as OpenStack or CloudStack.

2.1.4.3. XenServer

XenServer is a server virtualization platform that offers near bare-metal virtualization performance for virtualized server and client operating systems. XenServer uses the Xen hypervisor to virtualize each server, on which it is installed, enabling each to host multiple Virtual Machines simultaneously with guaranteed performance. XenServer allows you to combine multiple Xen-enabled servers into a powerful Resource Pool, using industry-standard shared storage architectures and leveraging resource clustering technology created by XenSource. In doing so, XenServer extends the basic single-server notion of virtualization to enable seamless virtualization of multiple servers as a Resource Pool, whose storage, memory, CPU and networking resources can be dynamically controlled to deliver optimal performance, increased resiliency and availability, and maximum utilization of data center resources. XenServer allows IT managers to create multiple clusters of Resource Pools, and to manage them and their resources from a single point of control, reducing complexity and cost, and dramatically simplifying the adoption and utility of a virtualized data center environment. With XenServer, a rack of servers can become a highly available compute cluster that protects key application workloads, leverages industry standard storage architectures, and offers no-downtime maintenance by allowing Virtual Machines to be moved while they are running between machines in the cluster. XenServer extends the most powerful abstraction: virtualization across servers, storage and networking to enable users to realize the full potential of a dynamic, responsive, efficient data center environment for Windows and Linux workloads.

- The XenServer product family
  - **XenExpress™** supports a single XenServer Host with dual sockets (or multiple XenServer Hosts, one at a time), up to 4GB physical RAM, hosting up to 4 concurrent VMs.

- **XenServer™** supports multiple simultaneous XenServer Hosts with up to 128GB physical RAM and no limit on the number of concurrent VMs except the amount of available RAM.
- **XenEnterprise™** supports multiple simultaneous XenServer Hosts with up to 128GB physical RAM and no limit on the number of concurrent VMs except the amount of available RAM. It also offers the following additional features:
    - Clustering of XenServer Hosts into Resource Pools.
    - Support for NFS and iSCSI shared Storage Repositories.
    - live relocation (XenMotion) of VMs within the same Resource Pool
    - support for specifying VLAN trunk ports in virtual bridges on the XenServer Host
    - additional Quality of Service (QoS) control for VMs

- **XenServer elements**
    - The Xen hypervisor.
    - Installers for both the XenServer Host and for XenCenter.
    - XenCenter, a Windows client application. From XenCenter you can manage XenServer Hosts, Resource Pools, and shared storage, and deploy, manage, and monitor VMs
    - The XE command line interface (CLI), for both Windows and Linux systems

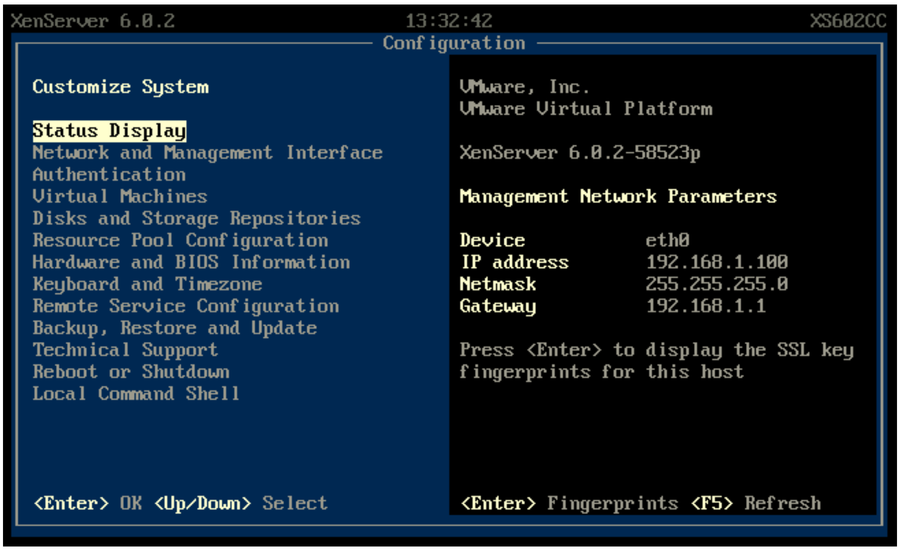


Figure 2.4 XenServer Screenshot.

XenCenter

XenCenter is the Windows-native graphical user interface for managing XenServer.it provides the following Features:

- Full virtual machine installation, configuration and lifecycle management.
- Access to VM consoles: VNC for installation-time, Xvnc for graphical displays on Linux, and Remote Desktop for Windows.
- Access to VM consoles: VNC for installation-time, Xvnc for graphical displays on Linux, and Remote Desktop for Windows.
- Remote storage configuration, including NetApp and DellEqualLogic StorageLink, and HBA (Fiber Channel and hardware iSCSI) support.
- Host networking management, including VLANs and internal networks, bonded and dedicated NICs.
- XenSearch: searching, sorting, filtering, and grouping, using tags, folders and custom fields.
- Dynamic memory management.
- Complete resource pool management.
- High availability configuration.
- Active Directory integration, for user authentication and revocation.
- Role-based access control
- VM snapshot management.
- Full memory snapshots and VM rollback.
- Disaster Recovery.
- Workload Reports, giving performance views over time and across the datacenter.
- Automated power management, to turn off idle servers.
- Performance metrics display.
- Long term metrics gathering and analysis.

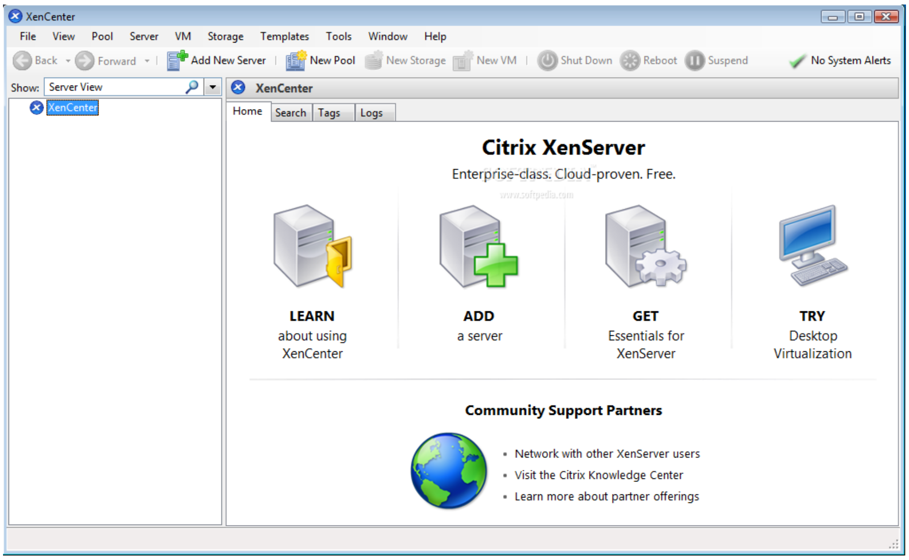


Figure 2.5 XenCenter Screenshot.

Section Two: Related Work

2.2. Section Two: Related Work

2.2.1 Introduction

This section reviews number of papers that related to the research topic , it display the main contributions of these papers and the difficulties which faced each papers, in addition to future works which suggested by these papers.

2.2.2 Papers Review

A Proposed Virtualization Technique to Enhance IT Services [1] paper talked about benefits that provides by virtualization in addition to disadvantages and challenges which are facing the virtualization.

The paper also proposes an effective and flexible distributed scheme with three phases; the scheme achieves reducing the workload of a data center and continual service improvement.

The proposed solutions are architected from the ground up to enable organizations to bring the benefits of virtualization to the entire enterprise platform, including server, storage and networking systems. The virtualization solutions Infrastructure not only enables organizations to build an entire virtual data center that matches the capabilities available in the physical world, but it goes one step further by offering the flexibility and control that is physically cost-prohibitive or impossible.

- There is proposed an Implemented plan which is easy to manage and scalable for future growth, resulting in reduced administrative burden.
- The proposed solution is an easier, faster and consistent centralized server deployment and protection of Virtual Machine.
- The paper discussed how System Virtualization Networking can help resolve network configuration challenges introduced by server virtualization.
- The paper has discussed management capabilities for virtualization Techniques.
- The paper has discussed how virtualization can help automate network configuration in virtualized environments, and how it enhances network control by making the network aware of virtual machines.
- With a whole suite of complementary virtualization and management services such as virtual VMware VMotion, VMware DRS, VMware HA and VMware Consolidated Backup, VMware Infrastructure is the only offering that delivers a complete solution rather than piecemeal approach for customers to build a data center in a virtual environment.

The paper defined the virtualization as follow:

"Virtualization is approach of making one resource look like many or many resources look like one. For example, Virtual Machines have long been used to make a single computer look like more than one (and Virtual Storage was used to divide up the real storage in a computer for each of the virtual machines)".

Hardware Virtualization Support in INTEL, AMD AND IBM Power Processors[2] paper aims to find out the advantages and limitations of current virtualization techniques, analyze their cost and performance and also depict which forthcoming hardware virtualization techniques will able to provide efficient solutions for multiprocessor operating systems. This is done by making a methodical literature survey and statistical analysis of the benchmark reports.

The paper presents the current aspects of hardware virtualization which will help the IT managers of the large organizations to take effective decision while choosing server with virtualization support.

The paper defined the virtualization as follow:

"Virtualization is a framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others".

The paper mentioned that Virtualization may provide new solution for some future challenges. Extension of existing operating systems to present the abstraction of multiple servers is required for implementation of virtualization at other levels of the software stack. Language level virtualization technologies may be introduced by the companies to provide language run-times that interpret and translate binaries compiled for abstract architectures enable portability.

Diagnosing Performance Overheads in the Xen Virtual Machine Environment[4] paper present Xenoprof, a system-wide statistical profiling toolkit implemented for the Xen virtual machine environment. The toolkit enables coordinated profiling of multiple VMs in a system to obtain the distribution of hardware events such as clock cycles and cache and TLB misses.

The paper evaluated network virtualization overheads in the Xen environment using different workloads and under different Xen configurations. The Xenoprof based profiling toolkit helped to analyze performance problems observed in the different Xen configurations. The paper identified key areas of network performance overheads in Xen, which should be the focus of further optimization efforts in. also the paper demonstrated the use of the profiling toolkit for resolving real performance bugs encountered in Xen.

Lastly the paper said that: "with the growing popularity of virtual machines for real world applications, profiling tools such as Xenoprof will be valuable in assisting in debugging performance problems in these environments".

Virtualization of Hardware – Introduction and Survey[5] paper introduced into the field of hardware virtualization, to discuss the main issues involved, and to provide a survey of important projects in this field. The paper identified three basic approaches of hardware virtualization: temporal partitioning virtualized execution and virtual machine. The paper discussed the programming models, execution architectures and tools for these approaches. Further, the paper presents a survey of the most important projects for each virtualization style.

Temporal partitioning applied to netlists and operation graphs. In the meantime, the main focus there has shifted to compilation from HLLs. the main role for temporal partitioning in embedded systems, where saving chip area and thus cost is an important optimization goal.

Virtualized execution comes in two flavors: approaches that compile to architectures with a fixed-size basic hardware element and approaches that map the atomic operator to variable-sized hardware elements at runtime. While the variable-sized approaches lead to potentially higher device utilization, the algorithmic problems involved are hard and their solutions time-consuming. The **virtual machine** approach is the newest one.

The paper defined the virtualization as follow:

"Hardware virtualization means that an application executes on virtualized hardware as opposed to physical hardware".

Pre-Virtualization: Slashing the Cost of Virtualization [3] paper talked about para-virtualization and its enormous cost and limitations in some features which comes with pure virtualization such as: stable and well-defined platform interfaces, single binaries for kernel and device drivers, and vendor independence.

The paper argues that the above limitations are not inevitable and present pre-**virtualization**, which preserves the benefits of full virtualization without sacrificing the performance benefits of para-virtualization.

The paper demonstrated that pre-virtualization technique achieves nearly the same performance as established para-virtualization approaches. However, it does so at significantly reduced engineering effort, and without requiring intimate knowledge of the guest OS. This not only reduces the cost of virtualizing a system, it also maintains confidence in the correctness of the virtualized OS, and makes it easy to keep the virtualized guest in sync with on-going development.

The paper authors also promise for applying techniques similar to pre-virtualization to the task of translating operating system source code between CPU architectures.

2.2.3 Papers Summary

#	Paper title	Author	Date	Paper contribution
1	A Proposed Virtualization Technique to Enhance IT Services	Dr Nashaat el-Khameesy and Hossam Abdel Rahman Mohamed - Sadat Academ	Nov. 2012	The paper proposes an effective and flexible distributed Virtualization scheme with three phases to Enhance IT Services
2	Hardware Virtualization Support in INTEL, AMD AND IBM Power Processors	Kamanashis Biswas and Md. Ashraful Islam , Daffodil International University.	2009	find out the advantages and limitations of current virtualization techniques
3	Diagnosing Performance Overheads in the Xen Virtual Machine Environment	Aravind Menon and other,HP Labs, Palo Alto.	2005	The paper present statistical profiling toolkit implemented for the Xen virtual machine environment.
4	Virtualization of Hardware – Introduction and Survey	Christian Plessl and Marco Platzner, Swiss Federal Institute of Technology (ETH).	2004	The paper provides a survey of important projects in Virtualization field.
5	Pre-Virtualization: Slashing the Cost of Virtualization	Joshua LeVasseur ,University of Karlsruhe, Germany , and others	Nov. 2005	Suggest Pre-Virtualization technique instead of para-Virtualization

Table 2.1 Papers Summary

CHAPTER THREE

Experiments Description

3. Experiments Description

3.1. Introduction

This chapter describe the experiments in details, it display the two sites of experiments , server site and client site and then display the hardware specifications and all of the technologies and tools that used into servers and clintes.in section three the chapter display data that used in the experiment, the last section discuss the experiment methodology.

3.2. Environment Setup

3.2.1. Servers side

Servers are the main elements on these experiments, there are two servers used, one of them is native server and another used virtualization technology.

3.2.1.1 Native server

In The Native server the operating system installed directly over the hardware.

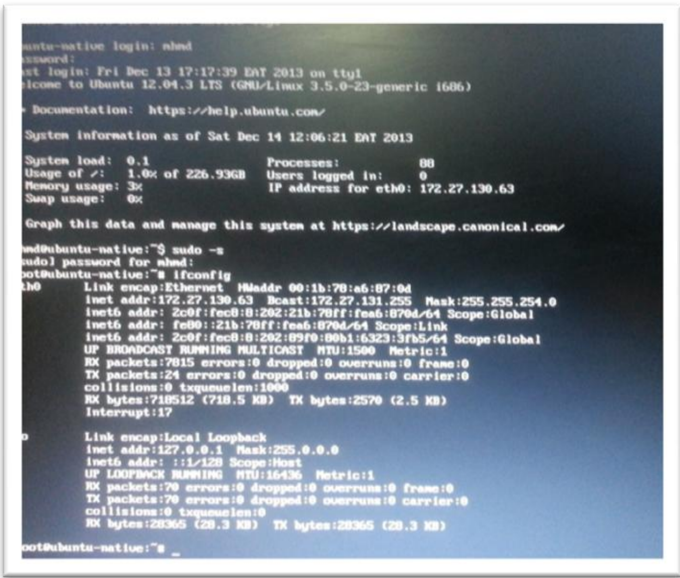


Figure 3.1 Native Server Screenshot

3.2.1.2 Virtual server

In the Virtual server hardware virtualization is used, we installed Xen Hypervisor and then create virtual machine and we installed the operating system on this virtual machine.

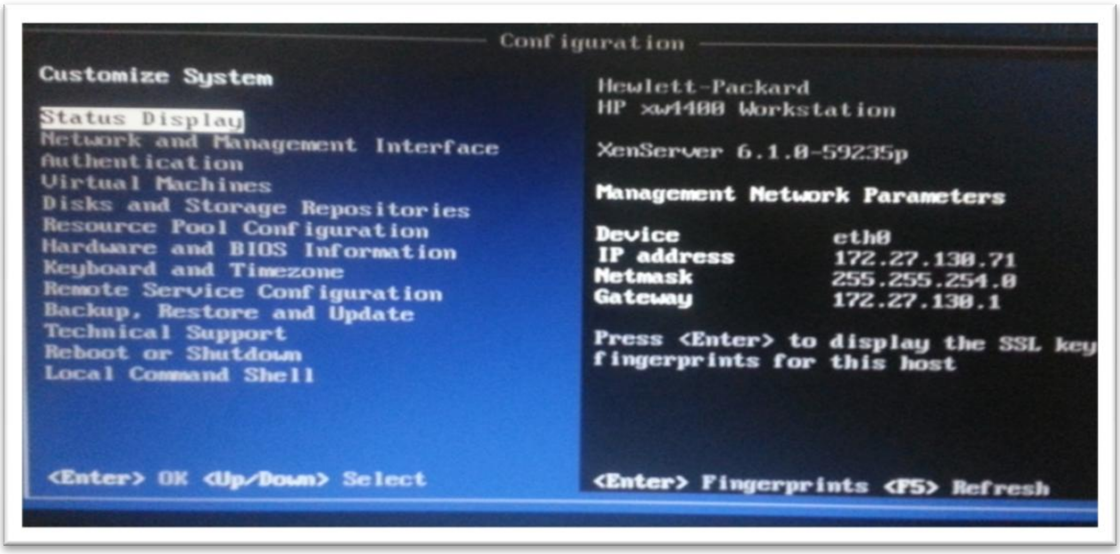


Figure 3.2 Xen Main Window Screenshot

3.2.1.3 Operating System

We used Ubuntu Server 12.04 LTS 64-bit as operating system on our two servers.

3.2.1.4 Database Management System (DBMS)

We used PostgreSQL-9.1 as database management system, it is an open source object-relational database management system (ORDBMS) provided under a flexible BSD-style license, for more details see (chapter2 ….).

3.2.2 Client Side

On the client side we used a PC to connect the two servers; on this client we installed many tools which help to execute experiments

3.2.2.1 XenCenter

XenCenter is the Windows-native graphical user interface for managing XenServer. For more details see (chapter2…), we used this tool to connect and configure the virtual machine. Figure 2 display XenCenter screen shot.

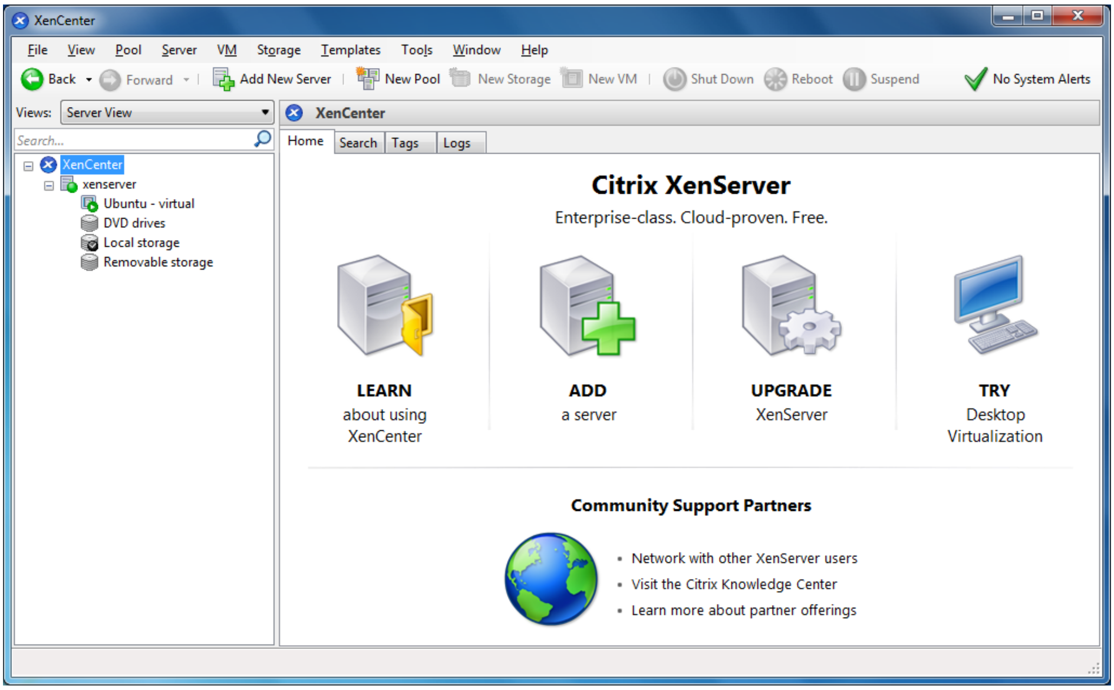


Figure 3.3 xencenter screenshot

3.2.2.2 Pgadmin3

PgAdmin3 is a database design and management application for use with PostgreSQL. The application can be run on any platform. for more details see (chapter2 …), .figure 3 display pgAdmin3 screen shot.

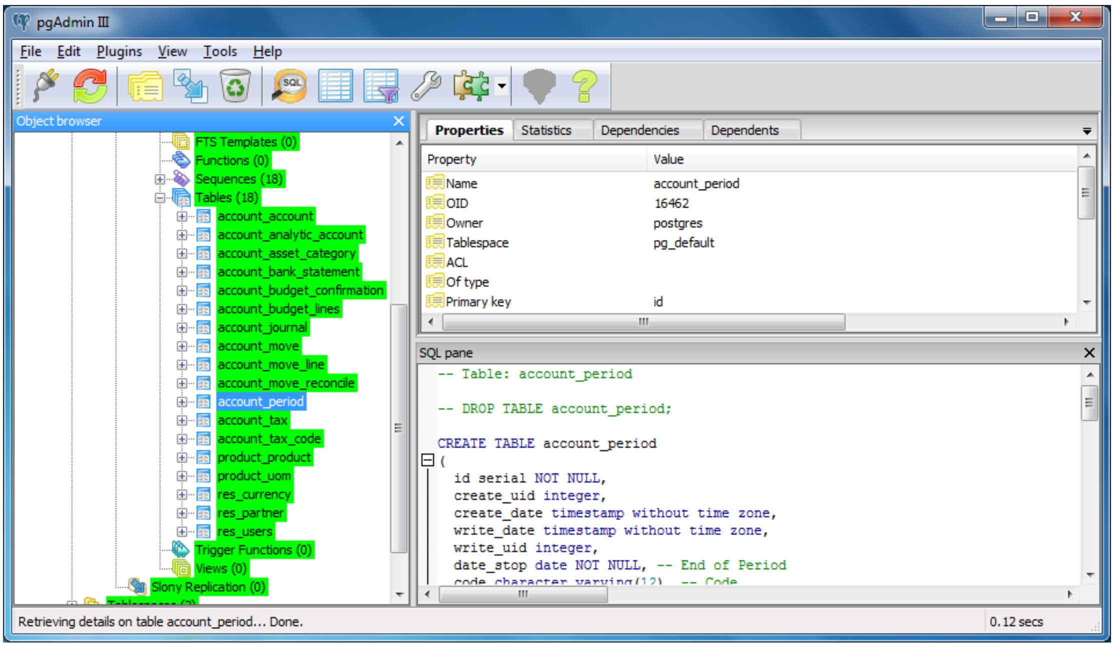


Figure 3.4: pgAdmin3 screenshot.

3.2.2.3 Putty

PuTTY is an SSH and telnet client, we used this tool to configure the OS on native and virtual servers.

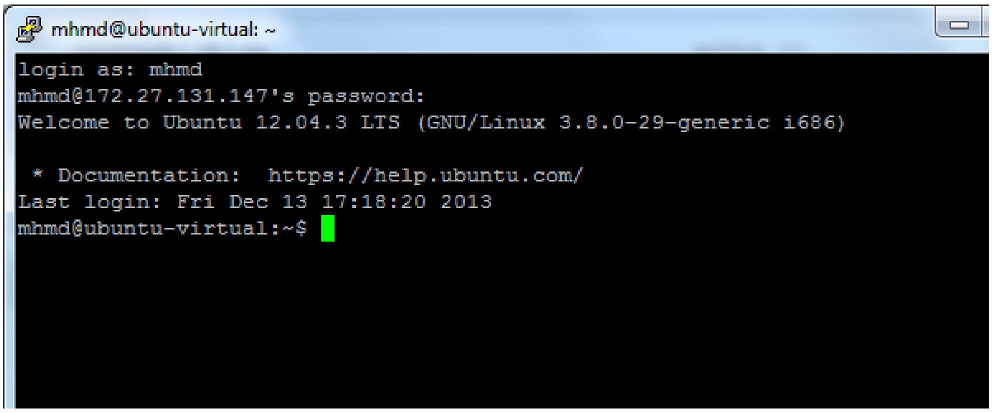


Figure 3.5 Putty screenshot

3.2.2.4 WinSCP

WinSCP (Windows Secure Copy) is a free and open-source SFTP, SCP and FTP client for Microsoft Windows. Its main function is secure file transfer between a local and a remote computer. We used it transfer files between client and servers and vice versa.

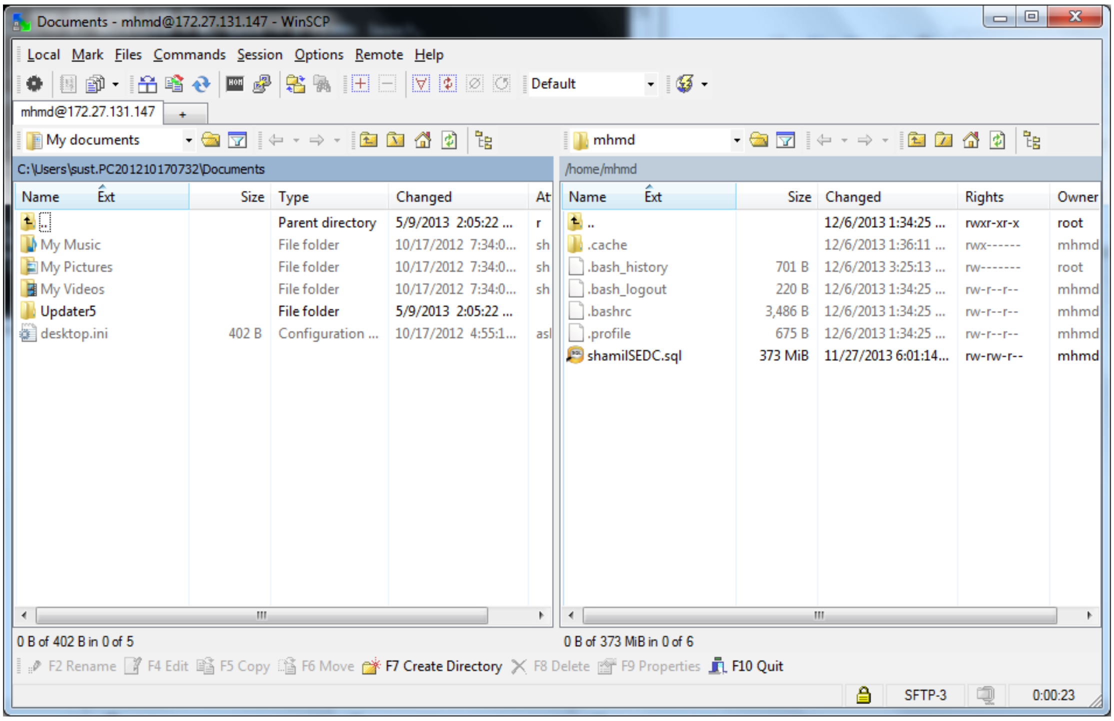


Figure 3.6 WinSCP screenshot.

3.2.3 Hardware specifications

- **System Info:**
  - Manufacturer: Hewlett-Packard
  - Model: HP xw4400 workstation.
- **RAM:**
  - Type :DDRAM
  - Size : 4096MB RAM

- **Processor:**
  - Type : Intel(R) Core(TM)2 CPU 6700
  - Speed: 2.66GHz (2 CPUs), ~2.7GHz

- **Storage:**
  - Size : 360 GB

3.3 Experiment Data

In all of our experiments we used a data that belong to ERP System of Ministry of water resources and Electricity (WRE); more accurate we used only the data of Account Module.

3.3.1 Database Schema

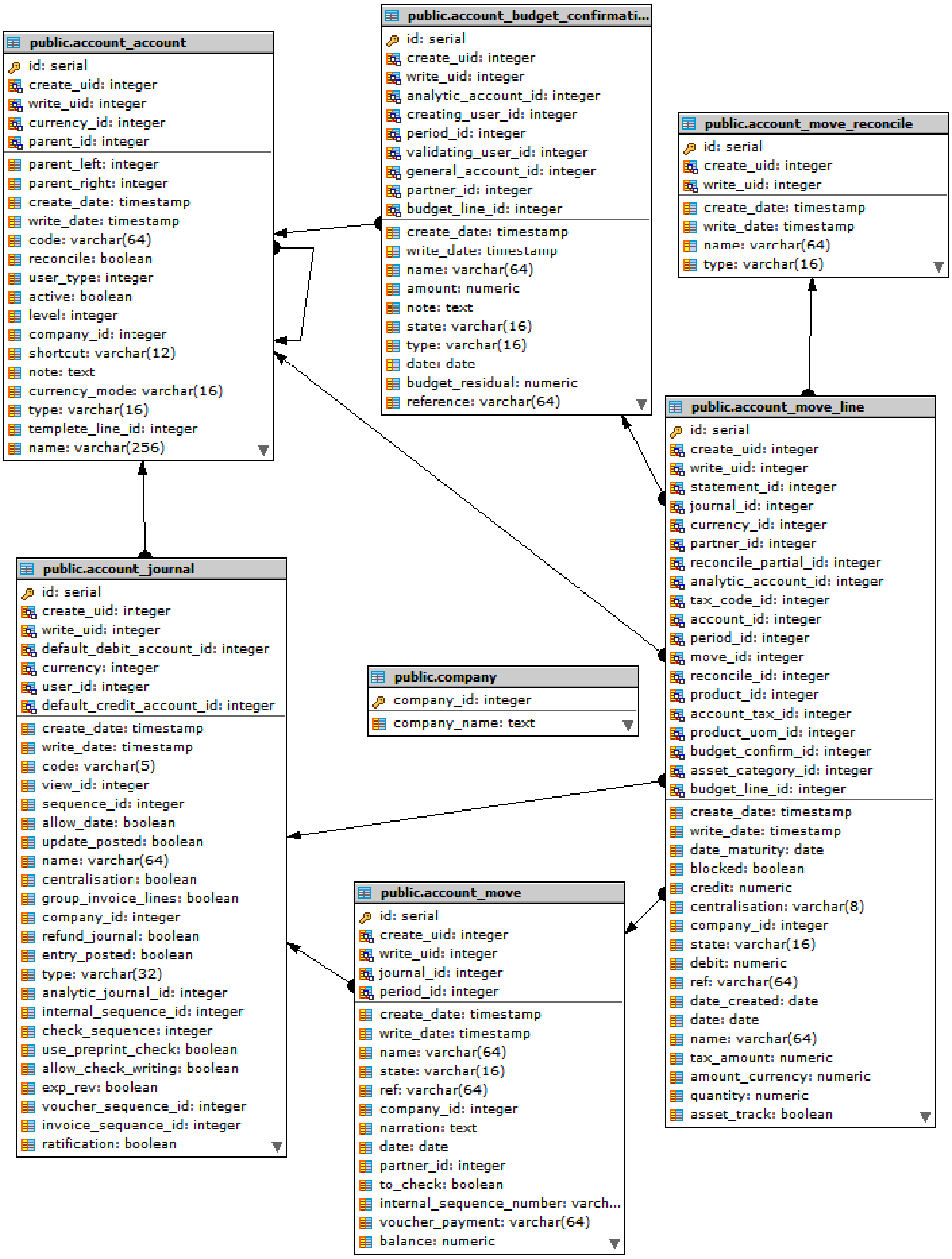


Figure 3.7 Database Schema

3.3.2 Tables Size

#	Table Name	Records No.
1	Company	16 (lockup tab)
2	account_journal	335
3	account_account	3826
4	account_move_reconcile	57435
5	account_budget_confirmation	94458
6	account_move	220137
7	account_move_line	1144310

Table 3.1 Database Tables Size

3.4 Experiment Methodology

The experiment aims to collect statistics and results about the performance of the servers in case of virtualized and native environments , to do experiments we use huge data of ERP system (called SHAMIL) that belong to Sudanese Electricity Distribution Company (SEDC) , these experiments have two parts

1- Measure the performance of Server Memory (RAM).

To achieve this we tried to retrieve huge data from the database using sql queries that consume the memory more than consuming processor because it just retrieve all records from the table, these quires detailed as follow:-

a. SELECT * FROM account journal;
b. SELECT * FROM account_account;
c. SELECT * FROM account_move_reconcile;
d. SELECT * FROM account_budget_confirmation;
e. SELECT * FROM account_move;
f. SELECT * FROM account_move_line;

2- Measure the performance of Server processing power:

To achieve this we tried to execute sql queries that join between two tables, this operation mainly consume the processor, these quires detailed as following:

a. select * from account_journal a,company b where a.company_id=b.company_id;
b. SELECT * FROM account_account A, company b where a.company_id=b.company_id;
c. select * from account_account a, account_move_reconcile b where a.create_uid=b.create_uid
d. select * from account_account a, account_budget_confirmation b where a.create_uid=b.create_uid;
e. select * from account_account a, account_move b where a.create_uid=b.create_uid ;

f. select * from account_account a, account_move_line b where a.create_uid=b.create_uid ;

Each of above quires execute five times for virtualized and native environment in parallel, then we calculate the difference between each two experiments results of virtualized and native environment, also we calculate the mean of five results of the two environments for each query and lastly calculate the difference of means, and we record the results on table formatted as follow:

#	Exp No.	Elapsed time		Difference
		Virtual	Native	
1				
2				
3				
4				
5				
	Average			

Table 3.1. experiments results format.

CHAPTER FOUR

Results and Discussions

4. Results and Discussions

4.1. Introduction

This chapter display results of the experiments that described on the previous chapter and calculate the average time of each experiment, then this chapter discuss the results of these experiments.

4.2. Experiments Results

4.2.1. Memory performance test

This section displays the results of memory performance test. all of the experiments on this section consume the memory rather than other resources.

4.2.1.1. Total result

Each of experiment repeated for five times and for each experiment we record the elapsed time for virtual server and native server and then we calculate the difference between two values, on the footer of the result table we calculated the average time of experiments results.

Table 1: account_journal (335 rows)

SELECT * FROM account_journal;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-R1.1	113	111	2
2	EXP-R1.2	118	111	7
3	EXP-R1.3	112	111	1
4	EXP-R1.4	115	112	3
5	EXP-R1.5	121	111	10
	Average	116	111	5

Table 4.1 Experiment 1.1

Table 2: account_account (3826 rows)

SELECT * FROM account_account;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-R2.1	991	985	6
2	EXP-R2.2	964	964	0
3	EXP-R2.3	965	966	-1
4	EXP-R2.4	967	965	2
5	EXP-R2.5	966	962	4
	Average	970	968	2

Table 4.2 Experiment 1.2

Table 3: account_move_reconcile (57435 rows)

SELECT * FROM account_move_reconcile;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-R3.1	5349	5341	8
2	EXP-R3.2	5328	5325	3
3	EXP-R3.3	5325	5324	1
4	EXP-R3.4	5324	5324	0
5	EXP-R3.5	5337	5324	13
	Average	5332	5327	5

Table 4.3 Experiment 1.3

Table 4: account_budget_confirmation (94458 rows)

SELECT * FROM account_budget_confirmation;

#	EXP ID.	Elapsed time		Difference
		Virtual	Native	
1	EXP-R4.1	47114	46802	312
2	EXP-R4.2	46829	46800	29
3	EXP-R4.3	46851	46798	53
4	EXP-R4.4	46875	46821	54
5	EXP-R4.5	46816	46803	13
	Average	46897	46804.8	92.2

Table 4.4 Experiment 1.4

Table 5: account_move (220137 rows)

SELECT * FROM account_move;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-R5.1	67872	67580	292
2	EXP-R5.2	67430	67270	160
3	EXP-R5.3	67286	68129	-843
4	EXP-R5.4	67286	67637	-351
5	EXP-R5.5	67479	68057	-578
	Average	67470.6	67734.6	-264

Table 4.5 Experiment 1.5

Table 6: account_move_line (1144310 rows)

SELECT * FROM account_move_line;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-R6.1	494644	494942	298
2	EXP-R6.2	494739	496094	1355
3	EXP-R6.3	497060	496514	-546
4	EXP-R6.4	494407	496514	-2107
5	EXP-R6.5	494551	496084	-1533
	Average	495080.2	496029.6	-949.4

Table 4.6 Experiment 1.6

4.2.1.2. Memory experiments average time

On this section all of the six memory performance test experiments average time for native server and virtual server and even the average time of difference between this two values and display this values as table (see table 4.7) and as graph (see graph 4.1).

a. data

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-R1	116	111	5
2	EXP-R2	970	968	2
3	EXP-R3	5332	5327	5
4	EXP-R4	46897	46804.8	18.44
5	EXP-R5	67470.6	96804.8	-264
6	EXP-R6	495080.2	496029.6	-949.4

Table 4.7 Memory experiments average time

b. Graph

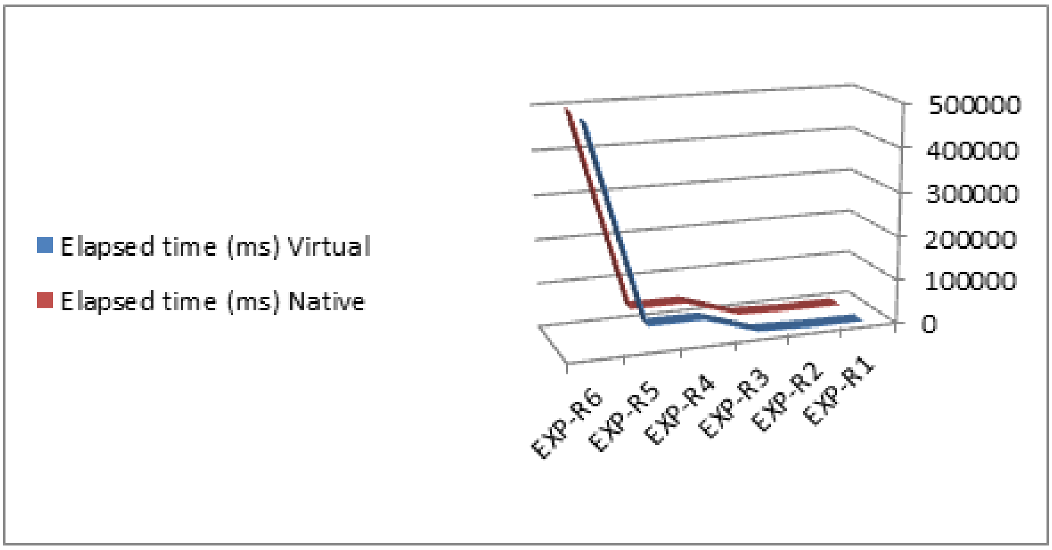


Figure 4.1 Memory Performance average time

4.2.2. Processing performance test

This section displays the results of processing performance test. All of the experiments on this section consume the processor rather than other resources.

4.2.2.1. Total results

Each of experiments repeated for five times and for each experiment we record the elapsed time for virtual server and native server and then we calculate the difference between two values, on the footer of the result table we calculated the average time of experiments results.

Table 1: account_journal (335 rows)

select * from account_journal a, company b where a.company_id=b.company_id;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-P1.1	218	219	1
2	EXP-P1.2	203	187	16
3	EXP-P1.3	188	187	1
4	EXP-P1.4	188	188	0
5	EXP-P1.5	188	188	0
	Average	**977**	**969**	**8**

Table 4.8 Experiment 2.1

Table 2: account_account (3826 rows)

SELECT * FROM account_account A, company b where a.company_id=b.company_id;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-P2.1	1653	1670	17
2	EXP-P2.2	1607	1606	-1
3	EXP-P2.3	1623	1622	-1
4	EXP-P2.4	1622	1622	0
5	EXP-P2.5	1638	1638	0
	Average	**8143**	**8158**	**15**

Table 4.9 Experiment 2.2

Table 3: account_account (3826 rows)

select * from account_account a, account_move_reconcile b where a.create_uid=b.create_uid;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-P3.1	610113	610034	-79
2	EXP-P3.2	609899	610211	312
3	EXP-P3.3	609615	609662	47
4	EXP-P3.4	610101	610242	141
5	EXP-P3.5	608994	609089	95
	Average	**609744.4**	**609847.6**	**103**

Table 4.10 Experiment 2.3

Table 4: account_move_reconcile (57435 rows)

select * from account_account a, account_budget_confirmation b where a.create_uid=b.create_uid;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-P4.1	115861	115815	-46
2	EXP-P4.2	115269	115066	-203
3	EXP-P4.3	115378	115097	-281
4	EXP-P4.4	115191	115066	-125
5	EXP-P4.5	115393	115253	-140
	Average	**115418.4**	**115259.4**	**-159**

Table 4.11 Experiment 2.4

Table 5: account_move (220137 rows)

select * from account_account a, account_move b where a.create_uid=b.create_uid;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-P5.1	1519394	1518861	-533
2	EXP-P5.2	1522009	1521790	-291
3	EXP-P5.3	1519163	1519772	-609
4	EXP-P5.4	1520733	1521188	-455
5	EXP-P5.5	1518220	1519039	-819
	Average	**1519904**	**1520130**	**-541.4**

Table 4.12 Experiment 2.5

Table 6: account_move_line (1144310 rows)

select * from account_account a, account_move_line b where a.create_uid=b.create_uid;

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-P6.1	7273451	7275535	-2084
2	EXP-P6.2	7277611	7280574	-2963
3	EXP-P6.3	7285802	7262246	-2317
4	EXP-P6.4	7283166	7285802	-2636
5	EXP-P6.5	7295673	7292811	-2862
	Average	**7283141**	**7279394**	**-2572.4**

Table 4.13 Experiment 2.6

4.2.3. Processor experiments average time

On this section we gathered all of the six processor performance test experiments average time for native server and virtual server and even the average time of difference between this two values and display this values as table (see table 4.14) and as graph (see graph 4.2).

a. data

#	EXP ID.	Elapsed time (ms)		Difference
		Virtual	Native	
1	EXP-P1	8143	8158	8
2	EXP-P2	8143	8158	15
3	EXP-P3	609744.4	609847.6	103
4	EXP-P4	115418.4	115259.4	-159
5	EXP-P5	1519904	1520130	-541.4
6	EXP-P6	7283141	7279394	-2572.4

Table 4.14 Processor experiments average time

b. graph

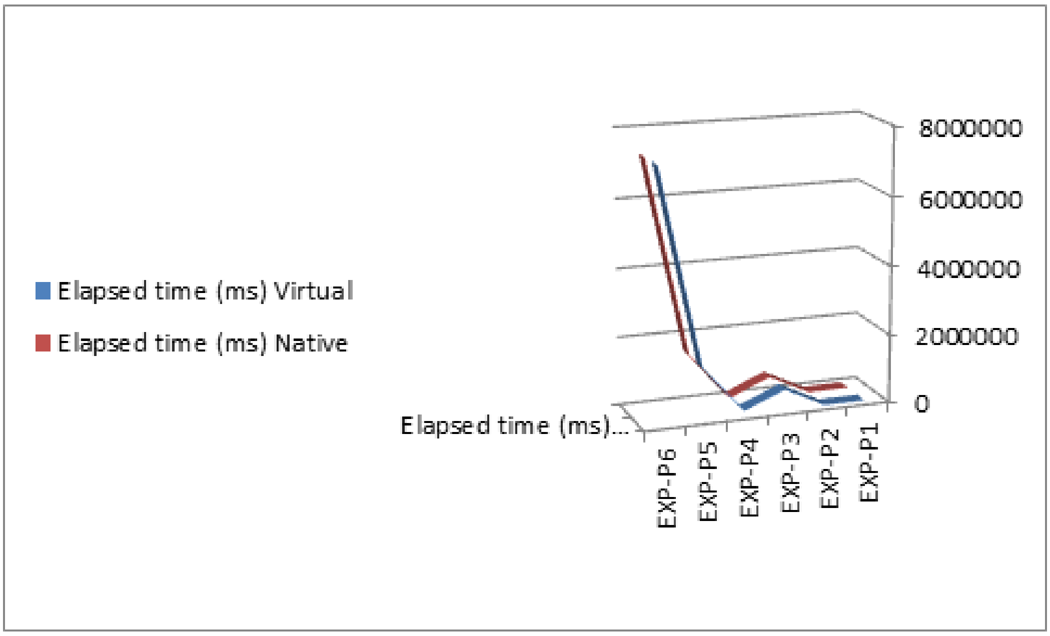


Figure 4.2 Processor Performance average time

4.3. Experiments Discussion

- **Memory performance test**

On Memory performance test experiments (as shown on tables 4.1.1 to 4.1.6) experiments from 1.1 to 1.4 a little superiority of the non-virtual server to the virtual server is noticed, the average differences on performance time in milliseconds were 5,2,5,92 Respectively . But on experiments 5 and 6 which are relatively large scale data, the superiority of the virtual server is noticeable on these two experiments it has finished the two experiments in a shorter time than native server by 264ms and 949ms Respectively.

The summary of above comparison indicates that the two server's performance is exactly the same when small data is used, and the average of differences on performance is about 0.1 second, however, when a huge data is used in the experiments a little superiority of virtual environment appears over the non-virtual environment with about 1 second.

(See this on the graph of memory performance average time Figure 4.1).

- **Processor Performance test**

On the Processor Performance test experiment (as shown on tables 4.2.1 to 4.2.6) experiments from 2.1 to 2.3 a little superiority of the non-virtual server to the virtual server is noticed,, the average differences on performance time in milliseconds were 8,15,103 Respectively. But on experiments from 2.4 to 2.6 which are relatively large scale data, the superiority of the virtual server is noticeable on these experiments, this superiority exceeded 2 seconds on experiment 2.6 so we can say that the performance of two servers is seem to be same when we used the small data (EXP-P1 to EXP-P3), but in case of large database tables (EXP-P5 and EXP-P6) we note that there is a little Superiority to the Virtualized Environment, and the max difference between tow server reached about 2.5 seconds (see table 4.13), but the difference is still small if we compare it with total time of experiments .

(See this on the graph of processor performance average time Figure 4.2).

- **The final Result**

The conclusion of all experiments the virtualized averments and the non-virtualized environments provides the same processing and memory performance when small data is used, however, when large data is used, the virtualization provides better performance than non-virtualization that is due to the tuning and better memory and processor management provided by hardware virtualization tools.

CHAPTER FIVE

Conclusion and Future Work

Introduction

This chapter displays the conclusion of the research and suggests some issues which can become topics of researches.

5.1. Conclusion of the research

The research provides literature review of hardware virtualization environments it displays the types of virtualization in details, in addition the research reviews the famous virtualization technologies and took one of them (which used in research experiments) in details. Then the research reviews number of papers that are related to hardware virtualization fields.

To measure the performance of virtualized and non-virtualized environments there are two type of experiments conducted to obtain the outcome of performance measurement of the two environments. The result of these experiments showed that there is no significant performance difference can be mentioned between the two environments. It is proven that virtualized environment offers performance rate similar to native hardware environment.

5.2. Future work

1- Using huge data and conduct same research experiments that may provide different results.
2- Conduct same research experiments with different virtualization technologies to ensure that the performance outcome is same.
3- Compare the Xen Technology with other Hardware virtualization tool such as Proxmox, VMware and KVM.
4- Mix the hardware virtualization technology with clustering technology
    a. Use two virtual machines one as master and another as failover.
    b. Or apply the load balancing by distributing the load among virtual machines.

References

[1] Dr Nashaat el-Khameesy, Hossam Abdel Rahman Mohamed , A Proposed Virtualization Technique to Enhance IT Services. I.J. Information Technology and Computer Science, 2012, 12, 21-30 Published Online November 2012 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijitcs.2012.12.02.

[2] Kamanashis Biswas, Md. Ashraful Islam, Hardware Virtualization Support in INTEL, AMD and IBM Power processors. (IJCSIS) International Journal of Computer Science and Information Security Vol. 4, No. 1 & 2, 2009.

[3] LEVASSEUR, J., UHLIG, V., CHAPMAN, M., CHUBB, P., LESLIE, B., AND HEISER, G. Pre-virtualization: Slashing the cost of virtualization. Technical Report 2005-30, Fakult¨at f¨ur Informatik, Universit¨at Karlsruhe (TH), Nov. 2005.

[4] MENON, A., SANTOS, J. R., TURNER, Y., JANAKIRAMAN, G., AND ZWAENEPOEL, W. Diagnosing performance overheads in the Xen virtual machine environment. In First ACM/USENIX Conference on Virtual Execution Environments (VEE'05) (June 2005).

[5] Christian Plessl, Marco Platzner: Virtualization of Hardware - Christian Plessl, Marco Platzner: Virtualization Of Hardware - Introduction and Survey. ERSA 2004: 63-69. ERSA 2004: 63-69.

Appendix A

XenServer Installation

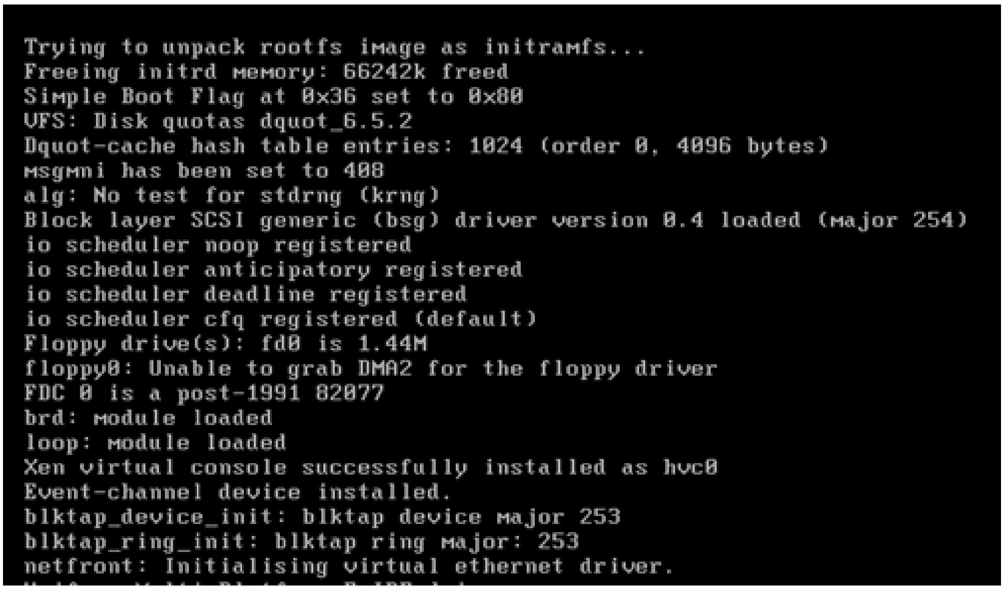
XenCenter Installation steps

1. Burn the downloaded ISO to a CD and insert it into your optical drive or mount it using software like MagicISO.
2. Make sure VT (or AMD-V) is enabled in your server's BIOS.
3. Boot to the CD.
4. Answer the basic installation questions and reboot.
5. Perform initial configurations, such as assigning it an IP address, a host name, password, and so on.
6. Connect to your XenServer using XenCenter.

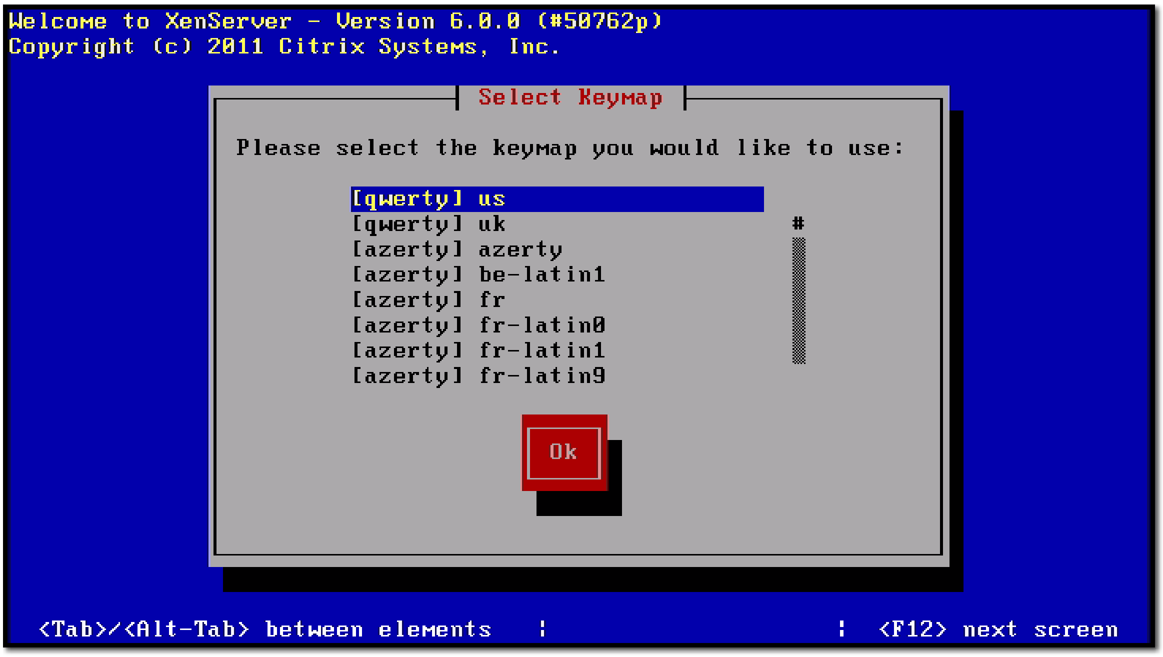
Installing XenServer

Mount the ISO CD on your physical server.

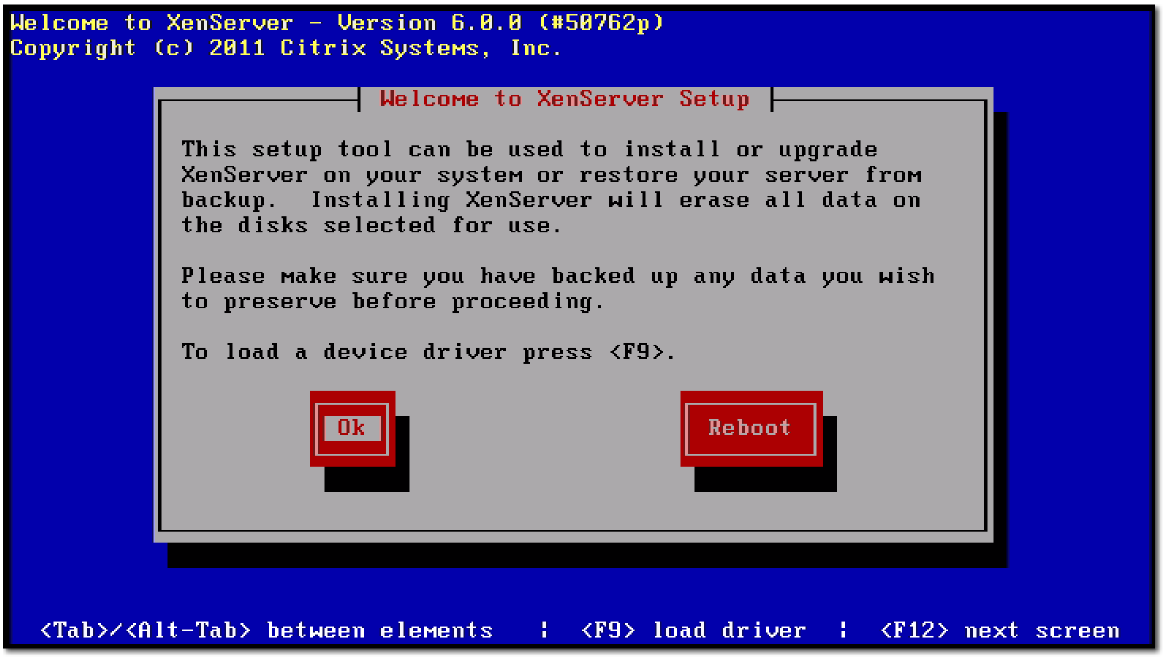
Once it boots up, it's going to start installing automatically.



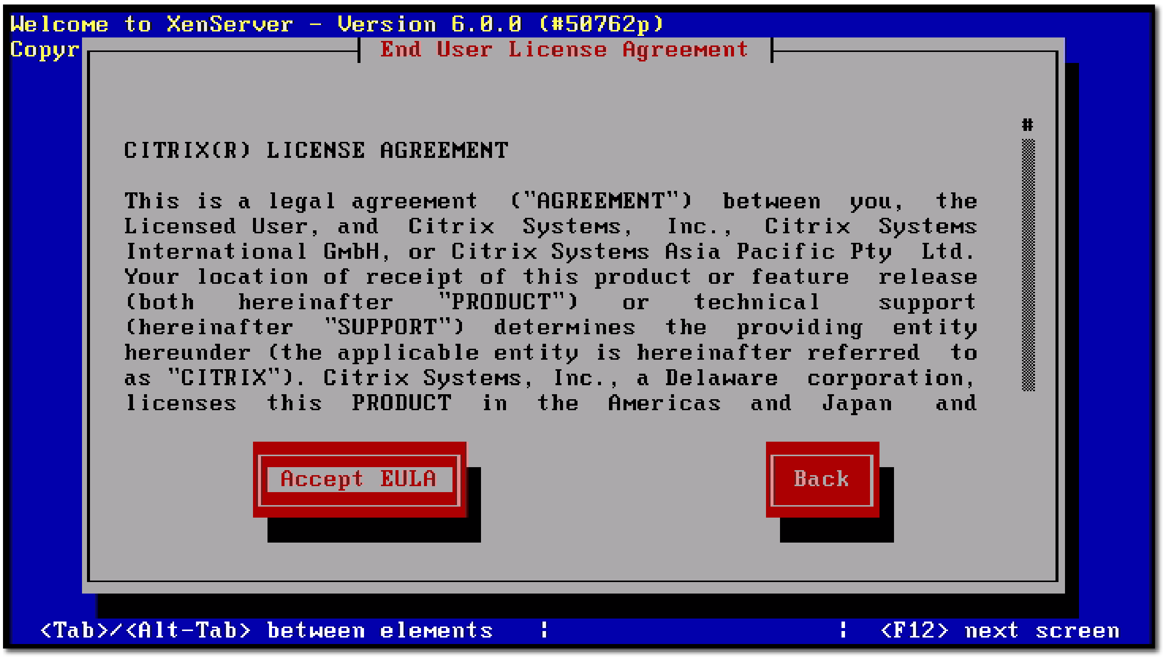
The first thing it will want to know is the type of keyboard you intend to use. Choose one and click OK.



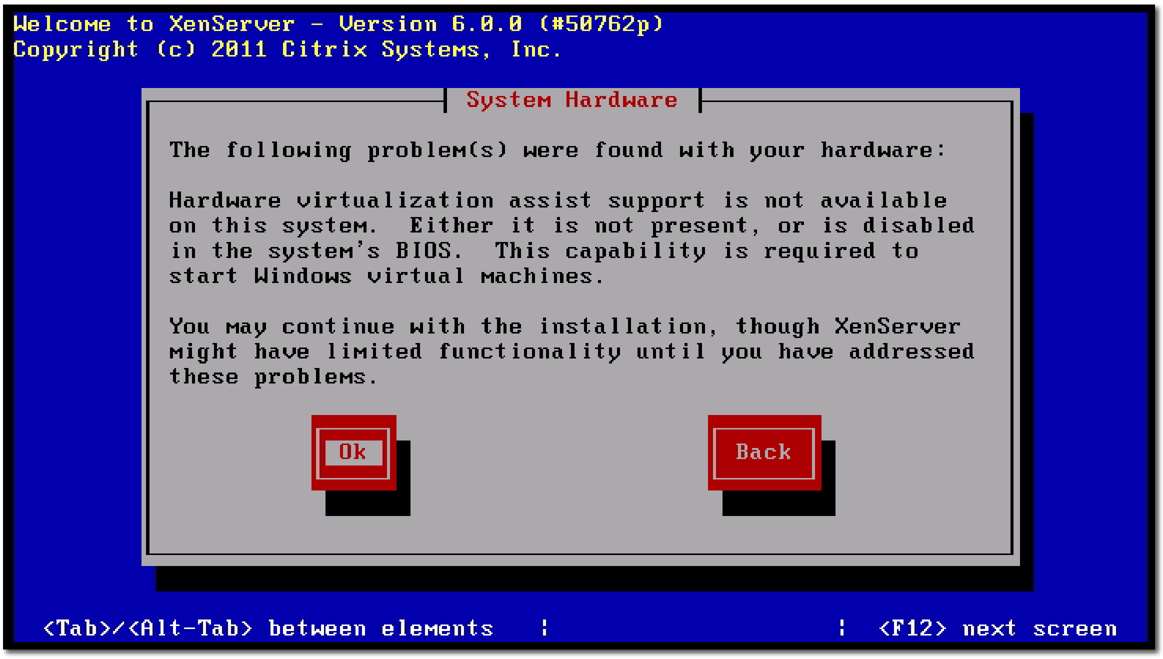
Click **OK** again.



Feel free to read the License Agreement and then click Accept EULA once you're done.

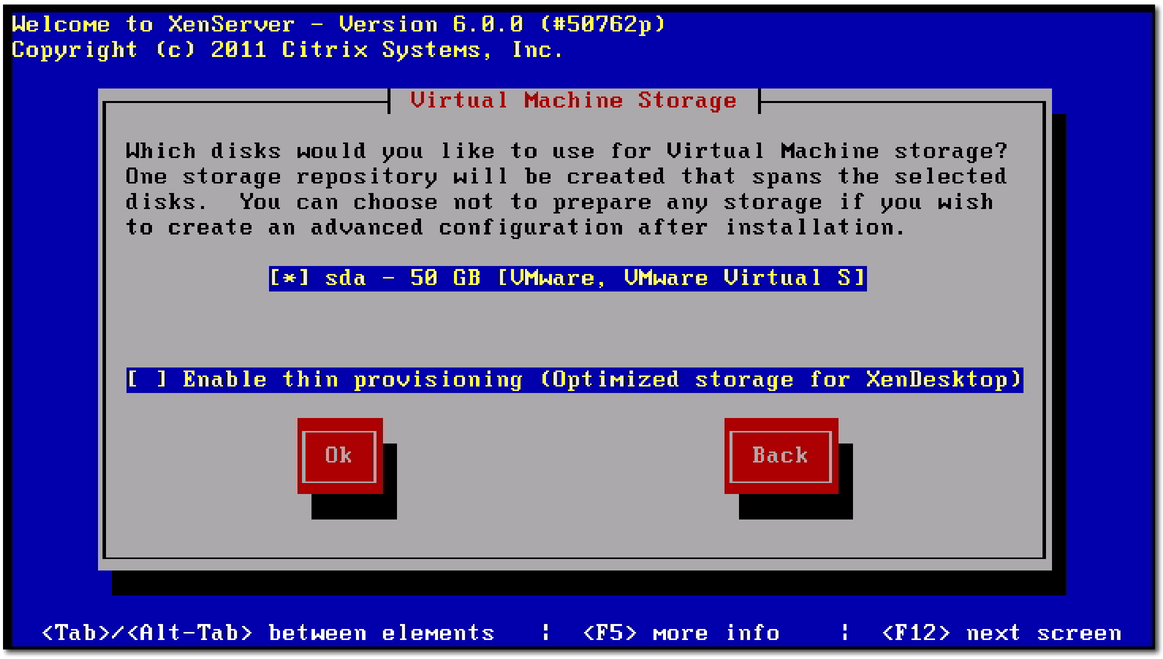


My CPU is already enabled with Intel VT. However, I chose to disable it in order to show you what you'll encounter when installing XenServer on a system that doesn't have Intel VT (or AMD-V) enabled. So, if your system has Intel VT (or AMD-V) but you've forgotten to enable it, this is what you'll see.

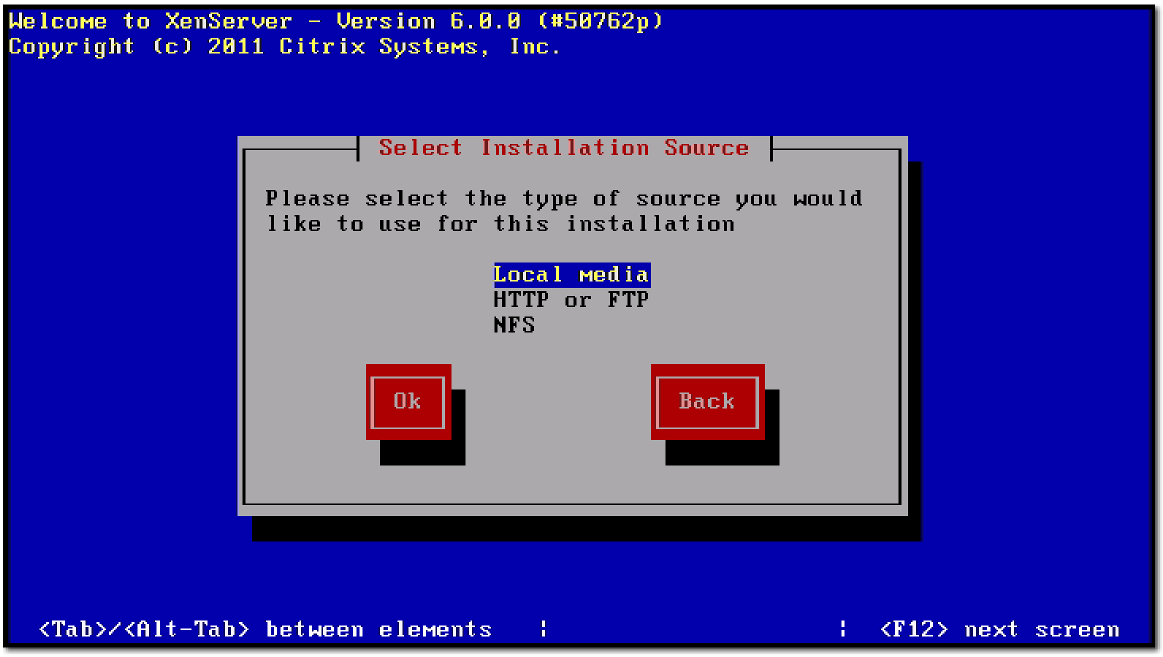


All you have to do is go back into your BIOS and enable it there. For now, let's just see what happens when you click OK.

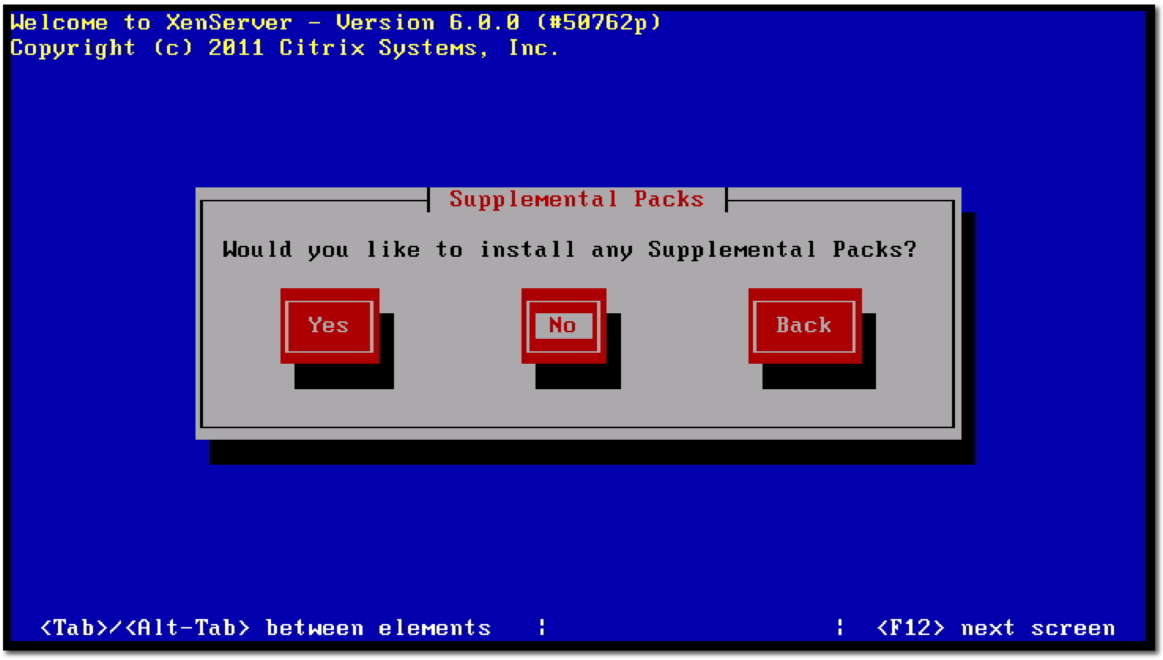
Apparently, the system will proceed to ask you where you would like to install XenServer. In my case, I have 50 GB of local disk space free, so I’m going to go ahead and install my XenServer there. I just select that and click OK.



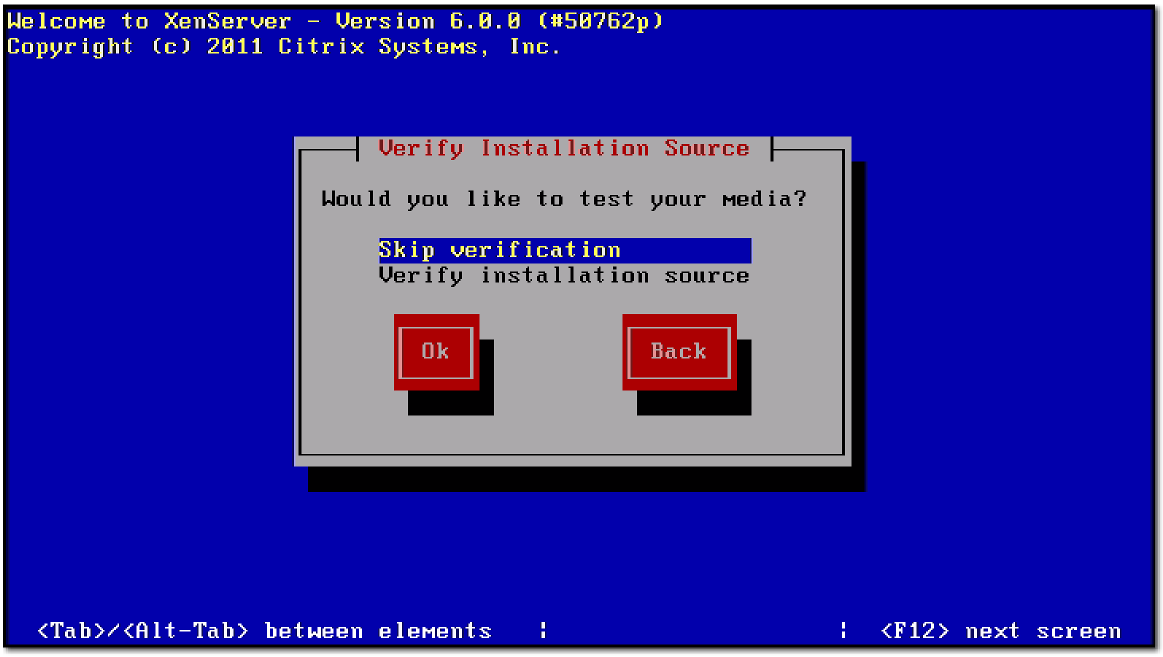
Select the source of your installation media. In my case, its Local media, but it can be HTTP, FTP, or NFS for you. Click OK.



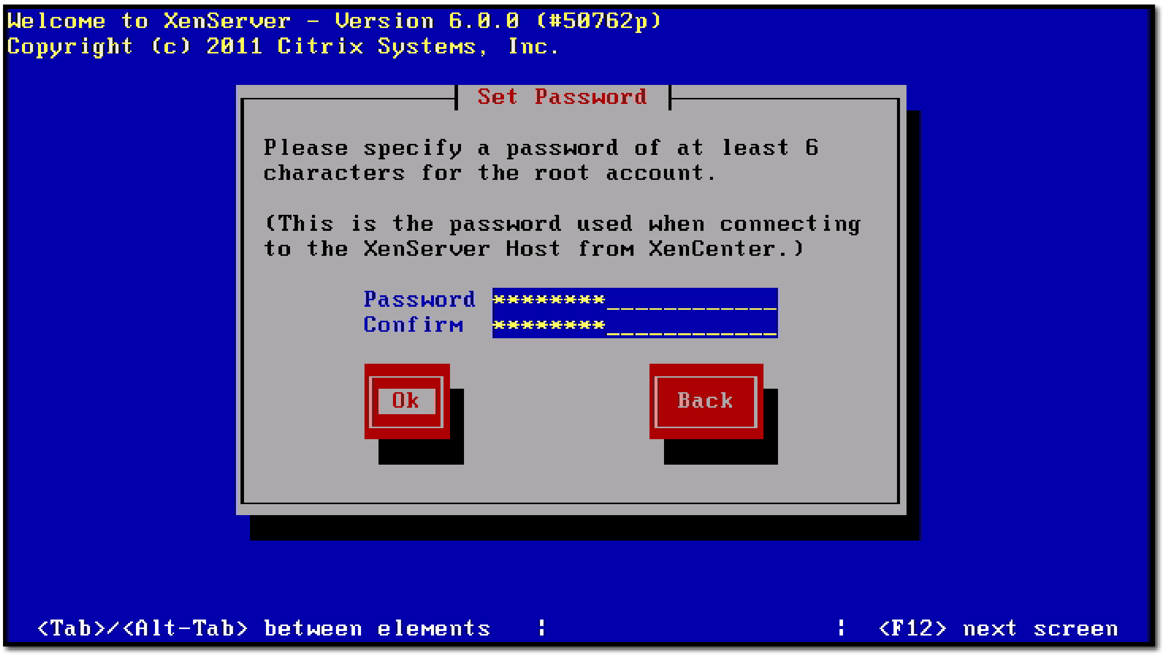
You'll then be asked whether you wish to install any supplemental packs like drivers of anything of that sort. In my case, I don't have any so I just select NO.



After that, you'll be asked whether you want to verify the integrity of the ISO you downloaded to make sure it's not corrupted and that it's a good ISO to install from. Since I'm sure my ISO's good, I'll just opt to skip this part and click OK.

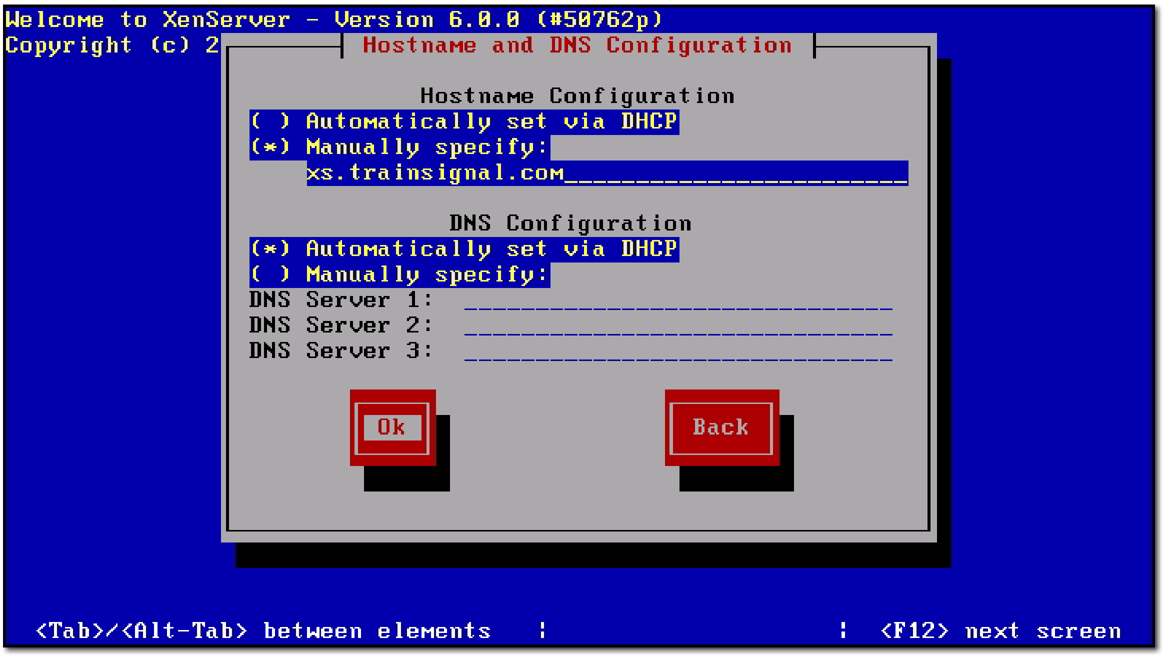


Set a password for your root account and click OK.



You can then choose whether to allow DHCP to assign this installation an IP address or to assign an IP address yourself. In my case, I'll just let DHCP do the assigning.

Next up will be the Hostname Configuration. Again, you can either let DHCP take care of that for you or you can assign a hostname yourself. I've chosen to assign one myself, giving it the name "xs.trainsignal.com". However, I'll leave it to DHCP to set the DNS Configuration. As you can see, you may also do things manually. Click OK.



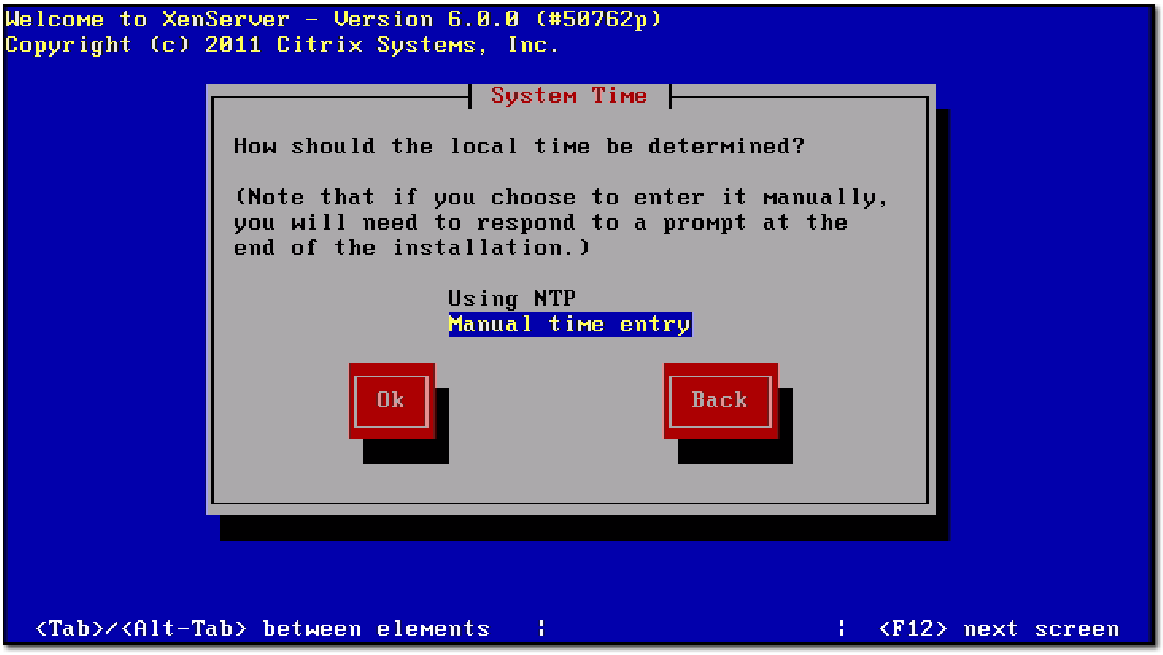
Find your geographical location, select it, and click OK.

Find the city or area you're in, select it, and click OK.

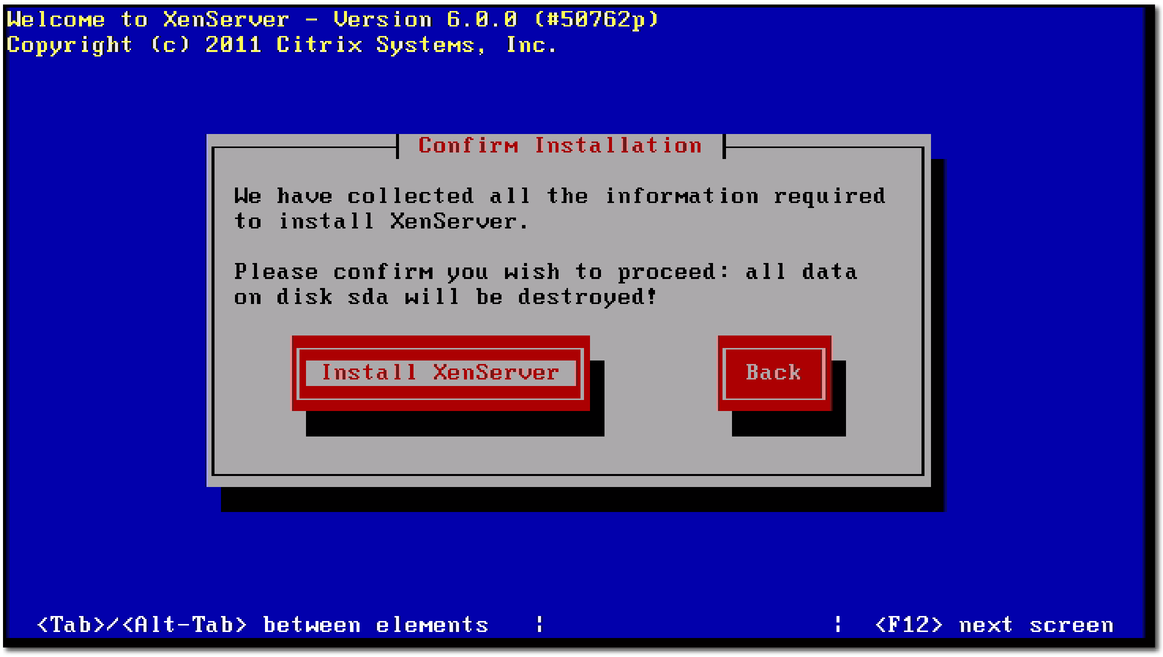
If you have an NTP server, it is critical that you configure the time settings on your XenServer properly. And really, if you'll be doing multiple installations of XenServer, you should have an NTP server of some sort running on your network. Once that's ready, you can select Using NTP on this screen and then enter your NTP server's IP address on the succeeding screen.

But for this walk through, I'm just going to go ahead and select the Manual time entry option.

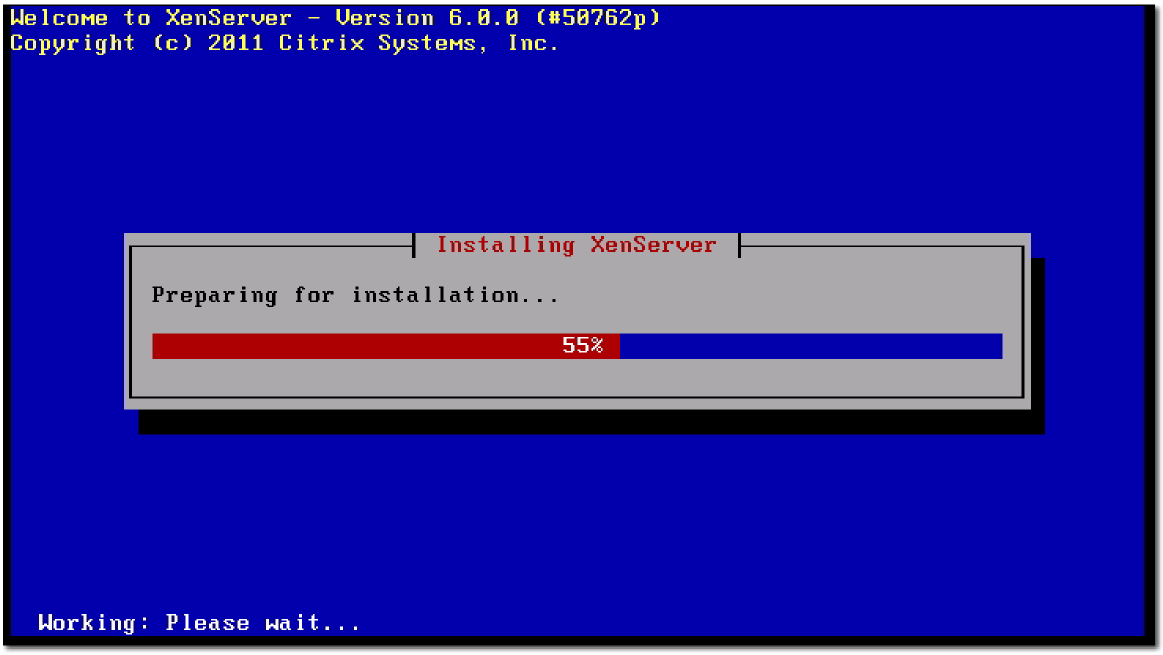
Click OK.



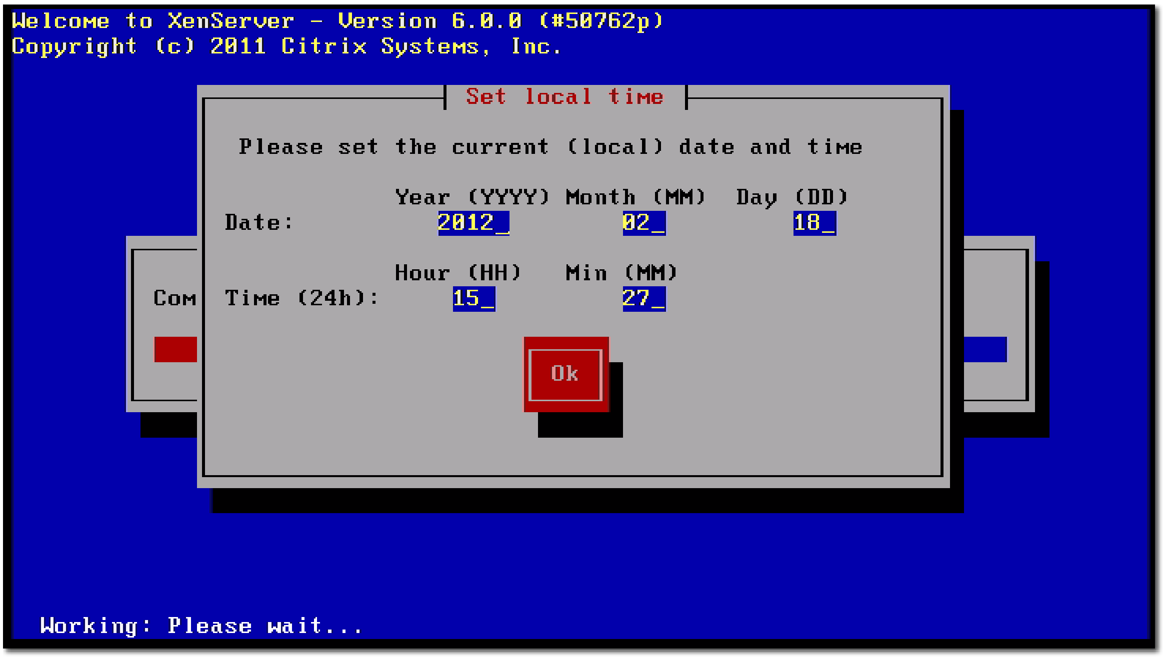
At this point, you should be ready to install XenServer, so go ahead and click Install XenServer.



Next you'll see the Preparing for Installation screen.

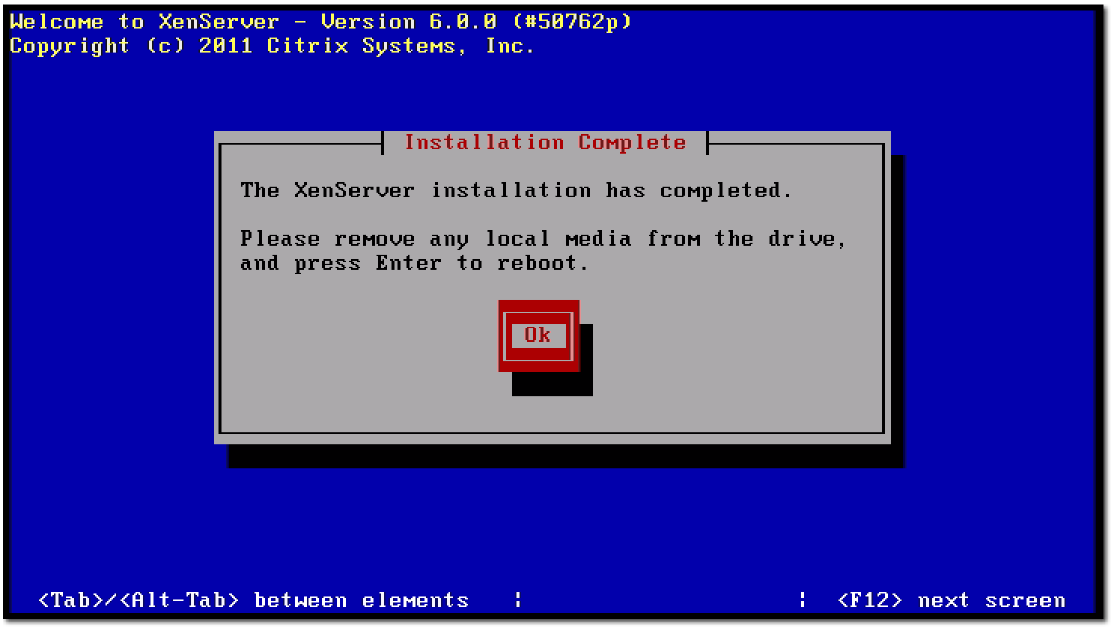


If you chose to set the time manually like we did, you'll see something like this:



Enter the correct time (or see to it that everything's alright) and then click OK.

When you get to the Installation Complete screen, click OK to complete the process.



Appendix B

Installing XenCenter

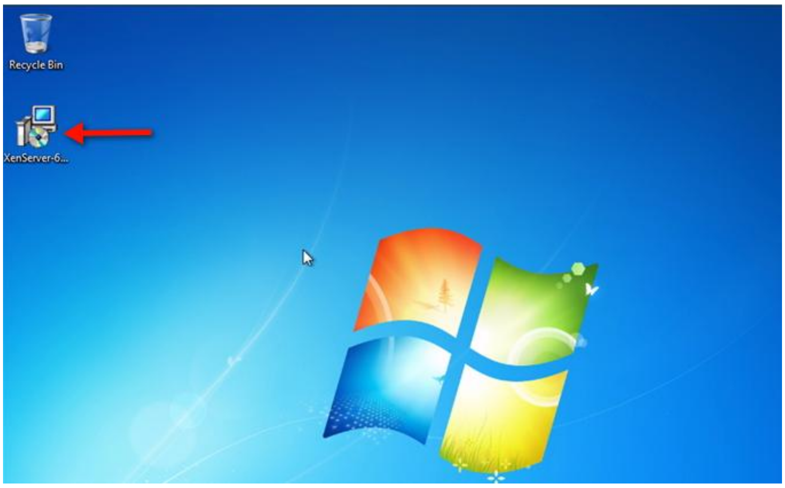
- XenCenter Installation Requirements

Citrix XenCenter is a Windows-based application which can be installed on either a physical or virtual machine.

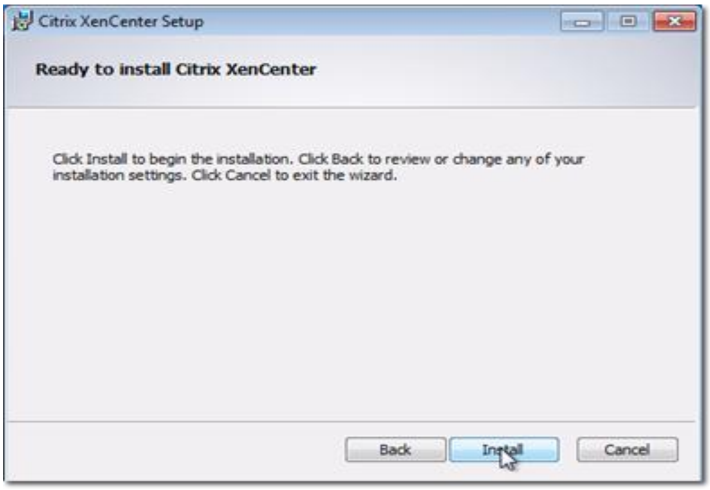
Its minimum CPU requirement is 750 MHz but 1 GHz is recommended. Minimum required RAM is 1 GB but the recommended is 2 GB. It occupies 100 MB of disk space and operates on a NIC with at least 100 Mb speed. Finally, the minimum required screen resolution is 1024 x 768.

- Installing XenCenter

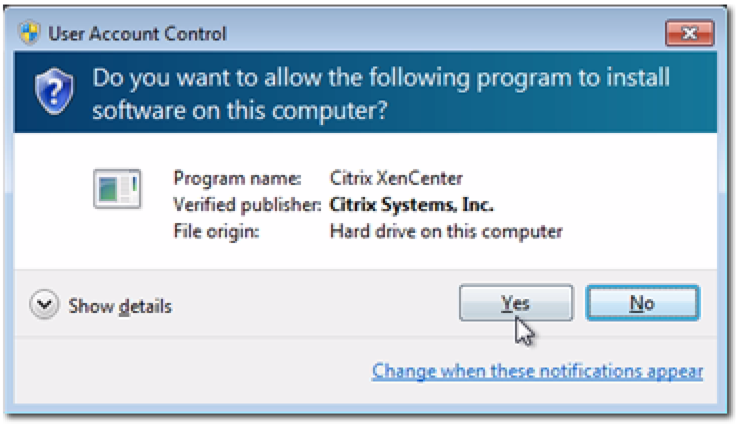
Download the installation file for XenCenter to a remote Windows environment.



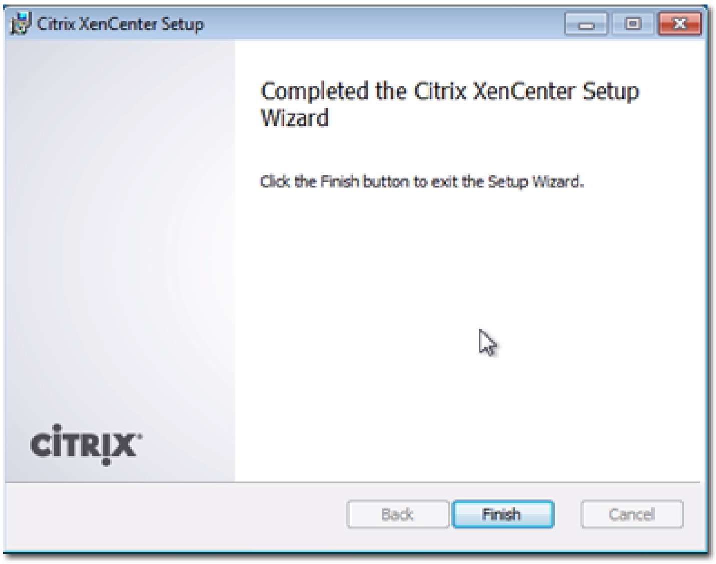
Once it's ready, launch the installer. It's just a simple installation, so basically you just need to keep on clicking **Next** until you get to the screen with the **Install** button. Click that.



Click **Yes.**

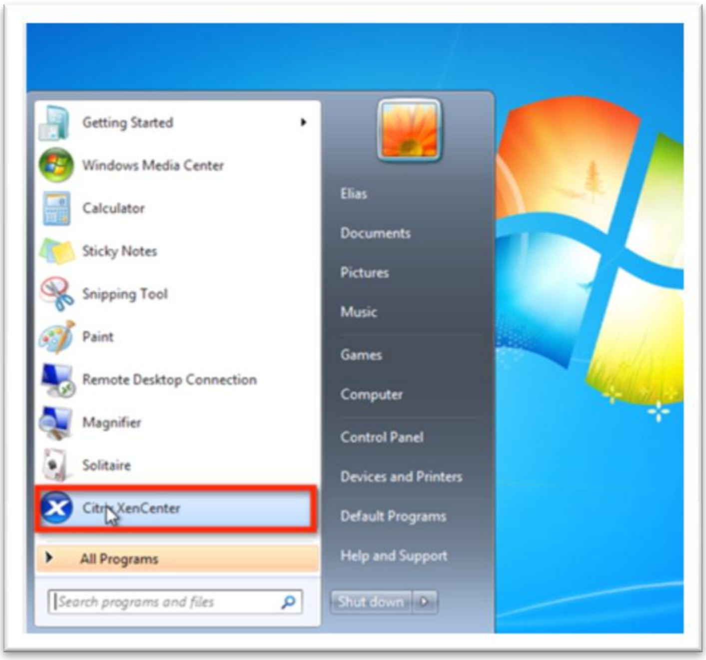


And then click **Finish.**

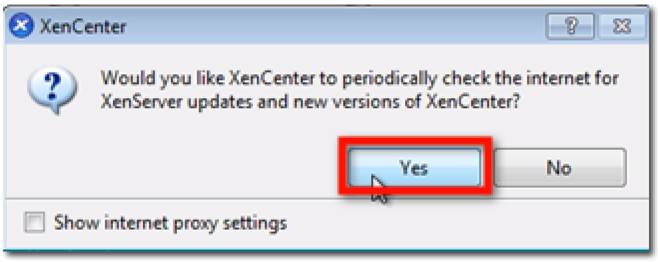


Tour of XenCenter

Before we wrap things up, let me give you a brief tour of XenCenter. Launch the program from the Start menu.

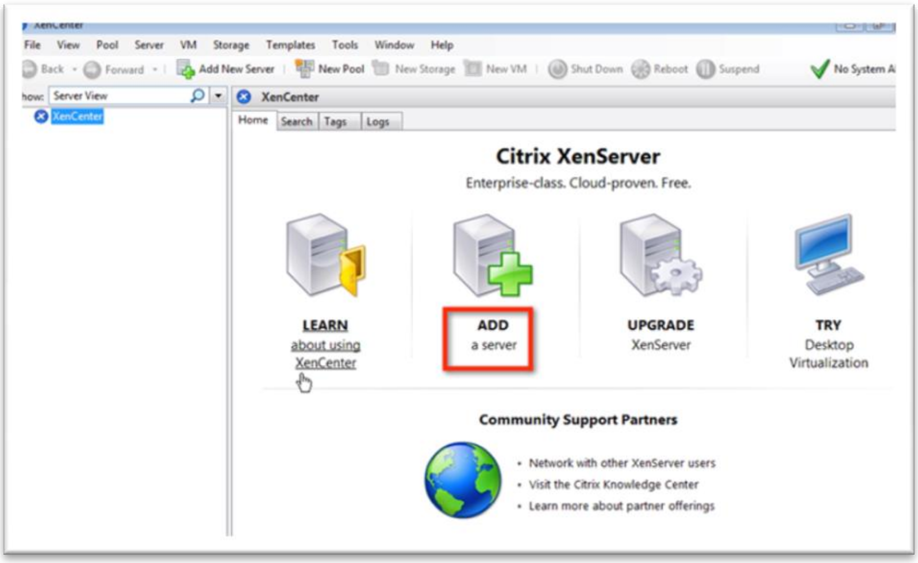


The first thing it's going to ask you is whether you want it to periodically check the Internet for updates, click **yes**.

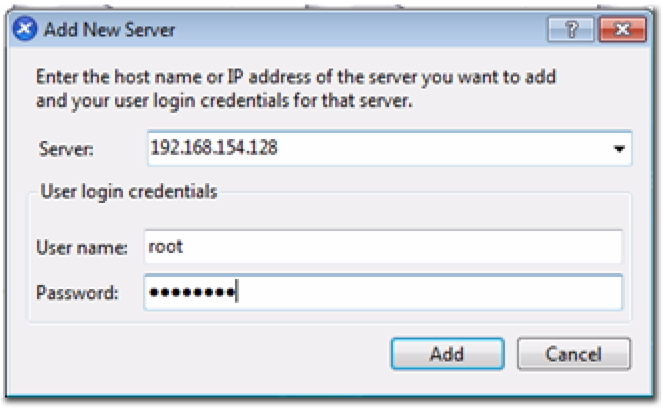


Here's how XenCenter looks like (see screenshot below) before any hosts, resource pools, and so on, are added to it.

To connect to the XenServer host you configured earlier, click **Add a server**.

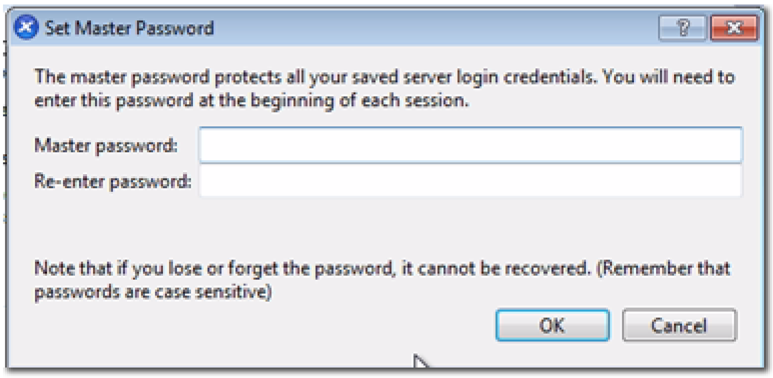


Enter the IP address I asked you to take note of earlier. Also enter the password you assigned for your root account. Click **Add**.



One of the first things you want to make sure as you're adding a new XenServer to XenCenter is to save and restore the server connection state on startup. Check the box that will do just that.

Once you do that, you will be allowed to configure a master password for all the XenServers you'll be associating with this XenCenter. Click the **require a master password** checkbox if that's what you want to do, and then enter your desired master password in the fields provided.



After you click **OK**, you'll be brought back to the main screen, where you'll see your XenServer already added to XenCenter.