



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

IMPLEMENTATION OF XEN VIRTUAL MACHINE AND DATA MIGRATION VIA NFS

¹C.Sathya, ²A.G. Aruna, ³Dr S. Agilan,

^{1,2}Assistant Professor, Dept. of Computer Technology and Applications,

³Associate Professor, Dept of Physics,

^{1,2,3}Coimbatore Institute of Technology, Coimbatore, India.

Abstract: This provides an easy access of storage and retrieves data from other machines. This system creates and runs multiple virtual machines (VM), using unmodified Linux guest images and returns their behavior. The Virtual bridges are created to make communicate between virtual machine and physical machine so that data transfer will be easier without any complexity. Network file system (NFS) is used for sharing files and directories over a network, and it requires configuration of an NFS Server (source) and NFS Clients (Destination). This supports for faster and secured data transfer between machines.

Keywords: Virtual Machine; Data Migration; NFS; Xen

I. Introduction

The primary functionality of the System is high secured data transfer between different hosts and Virtual Machines. XEN Hypervisor is implemented to run DOM-0 and Virtual Machines are created. Each Virtual Machine contains different individual Operating System. It involves in the application of the NFS, which acts as a layer between the underlying hardware and operating systems with XEN in terms of bridges to connect VMs which acts as an interface for communicating VMs and transfer data between them. These virtual machines are controlled and monitored by XEN hypervisor which act as virtual machine monitor. To make communicate between machines, bridges are created in Virtual Machines and host machine. Ethernet is used to communicate each bridge. Network File System (NFS) is used to communicate between virtual machines via bridges. to create these components, incorporating the applicable criteria that follow.

A. Xen Based Virtual Machine

A XEN-based virtual machine also referred to as a *VM Guest* or *Domain U* consists of the following components: At least one virtual disk that contains a bootable operating system. The virtual disk can be based on a file, partition, volume, or other type of block device. Virtual machine configuration information, which can be modified by exporting a text-based configuration file from xend or through Virtual Machine Manager. A number of network devices, connected to the virtual network provided by the controlling domain. Virtualization allows you to run multiple virtual systems on one physical machine. data transfer is possible between different operating systems. Secured data transfer with the knowledge of administrator. Any number of operating systems can be run and accessed simultaneously. Transfer of data will be quicker and easier.

B. The XEN Hypervisor

The XEN hypervisor sometimes referred to generically as a virtual machine monitor, is an open-source software program that coordinates the low-level interaction between Virtual machines and physical hardware.

C. The Domain0

The virtual machine host environment also referred to as Domain0 or controlling domain, is comprised of several components, such as: The SuSE Linux operating system, which gives the administrator a graphical and command line environment to manage the virtual machine host components and its virtual machines. The xend daemon (xend), which stores configuration information about each virtual machine and controls how virtual machines are created and managed.

D. VM Guests

It is often needed to dedicate specific CPU resources to a virtual machine. By default, a virtual machine uses any available CPU core. Its performance can be improved by assigning a reasonable number of physical processors to it as other VM Guests are not allowed to make use of them after them.

II. Architectural Design

XEN architecture system is the structure of the system, which compromise software and hardware elements, the externally visible properties of those elements and the relationship among them.

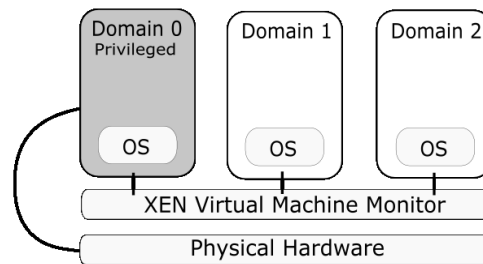


Figure 1 Architectural Design

A. Architecture of XEN Framework

The above Fig.1. illustrates the architecture of XEN framework. This system is a framework for implementation of XEN hypervisor architecture. It is designed to act as a middleware between hardware and domain guests. The XEN consist of virtual CPU, virtual memory, scheduling. The virtual CPU is mainly XEN virtualization, a thin software layer known as the XEN hypervisor is inserted between the server's hardware and the operating system. This provides an abstraction layer that allows each physical server to run one or more "virtual servers," effectively decoupling the operating system and its applications from the underlying physical server.

B. Creation Of Virtual Machines

Each virtual machine must have a unique name. The name entered on this page is used to create and name the virtual machine's configuration file. The Hardware page allows you to specify the amount of memory and number of Virtual processors for your virtual machine.

Memory

Initial Memory The amount of memory initially allocated to the virtual machine (specified in megabytes). **Maximum Memory** The largest amount of memory the virtual machine will ever need.

Virtual Processors

If desired, you can specify that the virtual machine has more virtual CPUs than the number of physical CPUs. You can specify up to 32 virtual CPUs: however, for best performance, the number of virtual processors should be less than or equal to the number of physical processors. a. Select the graphics mode to use: b. No Graphics Support, The virtual machine operates like a server without a monitor. Access the operating system through operating system supported services. Requires that an appropriate graphics driver is installed in the operating system.

Virtual Disk

A virtual machine must have at least one virtual disk. Virtual disks can be: File backed, which means that the virtual disk is a single image file on a larger physical disk. A sparse image file, which means that the virtual disk is a single image file, but the space is not preallocated. Configured from a block device, such as an entire disk, partition, or volume, Instead of the real DVD or CD-ROM drive, we can also add the ISO image of an installation medium.

Virtual network Card

By default, a single virtual network card is created for the virtual machine. It has a randomly generated MAC address that you can change to fit desired configuration.

Virtual Network

The network interface to the VM Guest system is made of a split device driver, which means, that any virtual Ethernet device has a corresponding network interface in Domain0. This interface is set up to access a virtual network that is run in Domain0. The bridged virtual network is fully integrated into the system configuration of SUSE Linux Enterprise Server and can be configured with YaST. When installing a XEN VM Host Server, a

bridged network configuration will be proposed during normal network configuration. The user can choose to change the configuration during the installation and customize it to the local needs. If desired, XEN VM Host Server can be installed after performing a default Physical Server installation using the Install Hypervisor and Tools module in YaST

C. Create Virtual bridges

When using SUSE Linux Enterprise Server the system configures one bridge for each physical network device by default. For each virtual bridge, a physical Ethernet device is enslaved, and the IP address assigned to the bridge. To add a new bridge, for example, after installing an additional Ethernet device, or to create a bridge that is not connected to a real network. Either use *Dynamic Address (DHCP)* as selected by default, or assign a static IP address to the bridge. Using *Dynamic Address* is only useful, when also assigning a device to the bridge that is connected to some DHCP server.

If intended to create a virtual bridge that has no connection to a real Ethernet device, use *Statically assigned IP Address*. In this case, it is a good idea to use addresses from the private IP address ranges, for example, 192.168.x.x or 10.x.x.x. To create a bridge that should only serve as a connection between the different guests without connection to the host system, set the IP address to 0.0.0.0 and the net mask to 255.255.255.255. The network scripts handle this special address as an unset IP address.

After the bridge is created, it may be used by any of the XEN VM Guest systems. A purely virtual bridge without connection to a real network device is good to provide fast network connections between different VM Guest systems. If provided a DHCP server on Domain0 that also defines routing information to the respective guest for the bridge, the network setup of the respective VM Guest is simplified. To get an overview about which interface is connected to which bridge, After creating Ethernet and Bridges, have to check the bridge connection using PING command. PING command is used to show the connection between two machines. If connection is established then machines are ready to transfer the files. Else it shows message as connection is unreachable.

III. Network File System

NFS is designed for sharing Files and directories over a network, and it requires configuration of an NFS server (where the files and directories are located) and NFS clients (computers that access the files and directories remotely). File systems are exported by an NFS server, and they appear and behave on a NFS client as if they were located on a local machine. For example, each user's home directory can be exported by an NFS server and imported to a client, so the same home directories are accessible from every workstation on the network. NFS is an RPC (Remote Procedure Call) service. An essential component for RPC services is rpcbind that manages these services and needs to be started first.

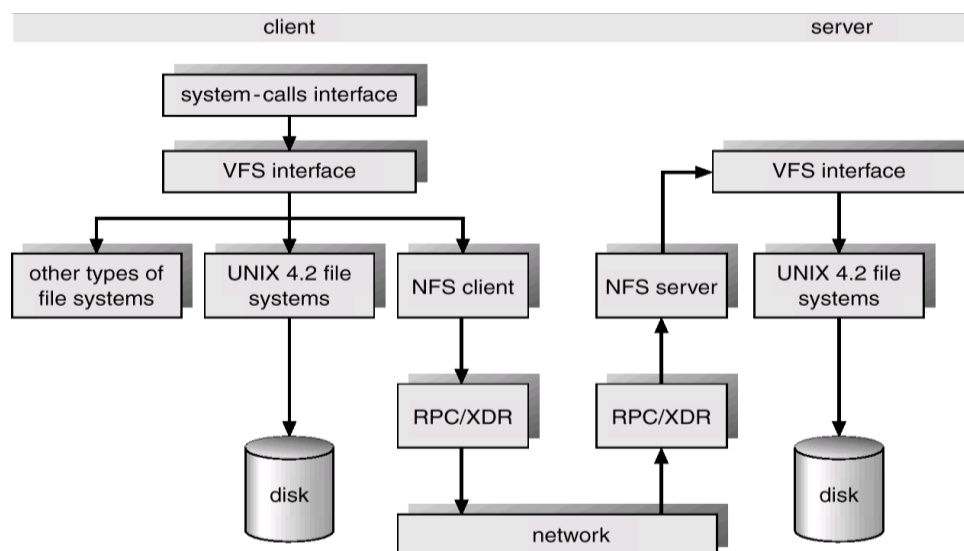


Figure 2 Network File System

A. NFS Server

To Add the /export/documentation directory to the list of directories: Select Add Directory.i. dialog appears where you have to specify the directory to export. Type /export/documentation, then select OK. In case the directory does not exist, a message informs of the fact and asks if want to create it. After confirmation, a dialog appears with fields for specify a Host Wild Card and Options. Change the preset values to match the following, then select OK. Hosts Wild Card: Options: rw, root_squash, sync, no_subtree_check (make sure replace "ro" with "rw"). The directory is added to the list. At the terminal window, verify that the file system was exported by entering the following: `Showmount -e localhostby`. View the entry made by YaST to the file /etc/exports by entering `cat /etc/exports`.

B. NFS Client

Add a remote file system to the NFS Client Configuration. Start the NFS Client Configuration from the terminal window by entering `yast2 nfs &`. Mount a remote file system by selecting the NFS Shares tab, then click Add. A dialog appears for adding the remote file system. Specify the following, then select OK. NFS Server Hostname: 127.0.0.1 (this is the local host address). Remote Directory: /export/documentation. MountPoint (local): /import/docs. Select the NFS Settings tab and deselect Enable NFSv4. List the files in the mounted file system by entering: `ls -l /import/docs`. Check the entry entered by YaST in the /etc/fstab file by entering `cat /etc/fstab`. This entry ensures that the files system is mounted each time he server boots.

IV. Virtual file system

Gives clean layer between user and filesystem. Acts as deflection point by using global vnodes. Understands the difference between local and remote names. Keeps in memory information about what should be deflected (mounted directories) and how to get to these remote directories.

A. System call interface

Presents sanitized validated requests in a uniform way to the VFS. Main goal of NFS protocol: NFS protocol designed without states. That's why it's very easy to recover server or client, because there are no states for them. NFS designed to support UNIX file system semantic, but protocol design can be adopted to support any file system semantic. Security and access check mechanisms based on Unix UID and GID mechanism. Stateless protocol. No need for server to hold information about which client is working with which file. To get their work done, server needs only information from RPC requests. Extensively use server cache to boost performance.

V. Experiment with NFS

Create a /export/documentation directory, copy documents from /user/share/doc/manual/ into it, and export it to others using NFS. In the second part, create the /import/docs directory and use it as mount point to import the /export/documentation directory from own server using NFS. Create an /etc/fstab entry to mount the directory automatically at boot time. In the third part, create the /data and /export/data directories, and then create some files in /data. Export the /export and /data directories using NFSv4. The /export directory should be the pseudo-root directory, with the content of /data appearing in /mountpoint/data for the clients. Mount the exported pseudo-root directory to /mnt using NFSv4 and check if the content of /mnt/ and /mnt/data is as expected.

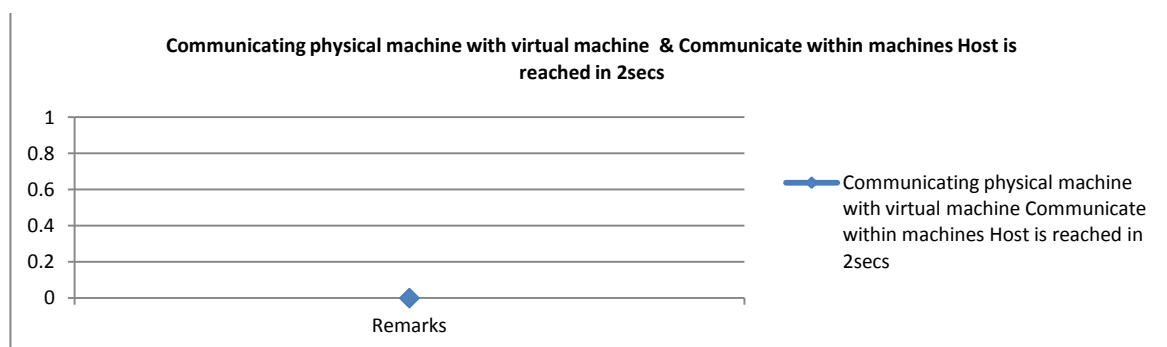


Figure3

A. Experimental Results

S.no	Name of the Test case	Feature being tested	Expected Output	Actual Output
1	Creating new Virtual Machine	Create a new virtual machine	The Virtual Machine should be created with given configuration	Domain-0 does not run
2	Creating new Virtual Machine	Create a new virtual machine		XEN kernel is not running
3	Creating new Virtual machine	Create new virtual machine		Virtual Machine created
4	Creating Ethernet in physical machine	Create Ethernet	The Ethernet is created with required Configuration	Install Packages to create Ethernet
6	Creating Network bridge in Physical machine	Create Network Bridge	The Bridge is created with Static IP Address	Invalid IP Address
7	Creating Network bridge in Physical machine	Create Network Bridge		Bridge is Created
8	Create Virtual Bridge in Virtual Machine	Create Virtual bridge	The Bridge is created Dynamic IP address	Invalid IP Address
9	Create Virtual Bridge in Virtual Machine	Create Virtual bridge		Virtual Bridge is Created
10	Communicating physical machine with virtual machine through NFS	Communicate within machines	Machines are been Communicated	Host is unreachable
11	Communicating physical machine with virtual machine through NFS	Communicate within machines		Host is reached in 2secs

Table1

VI. Conclusion

Benchmarks showed that the network performance provided by the virtualization layer is comparable to the supports for faster and secured data transfer between machines.xen hypervisor seems to be the best solution, particularly when using the para-virtualized approach. These tests have been very useful for us and gave some input on future investigations and tests to performance.

VII. References

- [1] David Schmidt and Puneet Dhawan, “ Live Migration with Xen Virtualization Software”, Dell Power Solutions ,2006.
Tim Abels , Puneet Dhawan,Balasubramnian ,Chandrasekaran,” An Overview of Xen Virtualization” Dell Power solutions, 2005.
- [2] Paul Barham,Boris Dragovic,Keir Fraser, Steven Hand ,Tim Harris, Alex Ho,”Xen and the art of virtualization” In Proceedings of the ACM on Operating Systems 3 ,New yark, oct 2003.
- [3] Whitaker, M. Shaw, and S. Gribble. Denali: Lightweight virtual machines for distributed and networked applications. In *Proceedings of the USENIX Annual Technical Conference*, June 2002.
- [4] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul,C. Kimpach, I. Pratt, and W. Warfield “Live Migration of Virtual Machines” In 2nd USENIX Symposium on Networked Systems, Design and Implementation (NSDI 05), **pages 273286, May 200.**
- [5] Weikuan Yu, Ranjit Noronha, Lei Chai, Shuang Liang, and Dhabaleswar K. Panda, Optimizing OpenSolaris NFS over RDMA. OSU-CISRC-4/060TR43.

- [6] Linux Enterprise11 administration manual by Novell Training Services & XEN-Virtualization” By Novell training services.
- [7] “Linux as a Hypervisor”, By Andrea Arcangeli , July 21, 2008.
- [8] W. Huang, J. Liu, M. Koop, B. Abali and D.K. Panda. Nomad: Migrating OS-bypass Networks in Virtual Machines. The 3rd International ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments (VEE'07), San Diego, CA, June 2007.
- [9] Laadan O, Nieh J (2010) Operating system virtualization: practice and experience. In: SYSTOR 2010, Haifa, Israel, 24–26 May 2010.