

# Virtualization

## Introduction

**Virtualization** decouples the physical hardware from the operating system with an '**abstraction layer**'. The abstraction layer is a piece of software that runs on the **host** operating system and presents interfaces to the **guest** operating systems such that each 'thinks' it's running on a particular set of hardware. The term 'virtual' arises from the fact that what the guests 'see' is not real hardware.

Virtualization allows you to:

- Run multiple virtual machines with heterogeneous operating systems at the same time on the same physical machine;
- Create fully configured isolated virtual machines, each with its own set of virtual hardware to run an operating system and applications;
- Rapidly save, copy and provision virtual machines that can be moved from one physical server to another for workload consolidation and zero downtime maintenance.”

*VMware website*

In a server environment, virtualization is used to reduce the amount of physical hardware (and hence running costs) and improve maintainability. One set of server hardware appears to users as a number of different computers, thus saving space, power and other resources. Each instance runs in a 'sandbox' so, if one 'machine' crashes the rest will keep going.

From our point of view, virtualization has advantages over the classic 'dual-boot' configuration. It also allows us to test new operating systems and distributions without affecting our base system.

We FOSS desktop users are usually interested in running Microsoft Windows on top of Linux (to run those annoying applications that won't run on anything else but Windows) or, perhaps, we could be interested in comparing several Linux distributions. Virtualization enables us to test how a website we are developing works/appears when accessed by users with different operating system/browser combinations.

Of course, Microsoft thinks that the only way anyone would want to run a virtual set-up is to have a version of MS Windows as the host, with other OSs running as guests. They have complex licensing schemes, backed up by EULAs and hidden pitfalls in their code, to make it difficult to do anything that they don't want you to do and to extract money from you for every instance of every installed OS if you capitulate. Apple has a similar mindset – you can run virtual machines on OS X, but the EULA prohibits installing OS X on any non-Apple machine (*Apple are currently pursuing a start-up, Psystar, through the courts in the US for doing just this!*).

## Pro & Con

- ◆ Unlike 'dual-boot' you can switch instantaneously from one OS to another. The virtual machine appears as a window on your desktop. *In fact, the virtual desktop can be displayed 'full-screen'; alternatively, with a bit of 'smart' the desktop can display the Gnome task bar at the top and the Windows task bar at the bottom.*

## Virtualization

- ◆ If a virtual copy of a Windows installation becomes infected by malware, you can ditch the whole thing and revert to the previous saved image – which is a lot less trouble than doing a full re-install!
- ◆ Doesn't suffer from the problems that prevent some MS Windows applications running under WINE, but ...
- ◆ The virtualization software provides dummy resources for the guest OS to connect with. These may be more restrictive than the 'bare-metal'. So, for example, getting USB and/or SATA drives to work may require extra effort.
- ◆ There is a performance penalty, but these days not much, especially if your CPU includes the virtualization tweaks (VT-x for Intel, AMD-V for AMD) *None of mine do – Intel only provide these features on their high specification models, and my AMD processor pre-dates their general introduction!*

## Products

Commonly mentioned products include:

- ◆ KVM (Kernel-based Virtual Machine) (Red Hat – now incorporated in Linux kernel)
- ◆ Parallels (for Apple Mac)
- ◆ QEMU
- ◆ VirtualBox (Sun) (has both a proprietary version – downloadable for free under a Personal Use and Evaluation Licence (PUEL) – and an open-source edition (OSE) released under the GPL. The OSE version is in the Ubuntu repositories.)
- ◆ VMware (proprietary, but has an open-source version for non-commercial use)
- ◆ Xen (link-up with Microsoft)

I have used VirtualBox and VMware. Examples used in this talk will be based around VirtualBox.

### Note:

*'The use case targeted when KVM was moved into main is "server virtualization". This means that even though KVM can be used to serve other purposes, it has been designed to be run on Ubuntu Server Edition to host **non-graphical** [my emphasis] server operating systems. If you are looking for software to serve graphically-based virtual machines, VirtualBox, Parallels Workstation (or Parallels Desktop for Mac), or VMware Player/Server are more suitable alternatives.'*

<https://help.ubuntu.com/community/KVM>

## Requirements

- ◆ A reasonably powerful CPU (preferably with the additional virtualization instructions)
- ◆ Enough hard drive space (say 10 GB)
- ◆ Enough RAM to run the host and guest at the same time (say, at least 1GB)

# Virtualization

## How

- ◆ Install virtualization software of choice,  
[VirtualBox OSE is in the Ubuntu repositories; if you want the PUEL version, you need to download it from the VirtualBox website – there is a .deb file for Ubuntu [http://www.virtualbox.org/wiki/Linux\\_Downloads](http://www.virtualbox.org/wiki/Linux_Downloads)]. It's also a good idea to download the User Manual from here: <http://www.virtualbox.org/wiki/Downloads> (it's 269 pages).

### Method 1:

- ◆ Install OS of choice from CD, USB stick or even an ISO image. Instructions are in the 'GettingStarted' PDF.  
**Note: if installing Windows you need a full retail CD** – a recovery CD won't do.  
**Problem:** after installing from the CD, then it will be necessary to do all of Microsoft's updates since the CD was pressed (multiple restarts inevitable), install anti-malware products and install all of the applications to be run under this OS in the usual way.

### Method 2:

- ◆ Alternatively, you may be able to make a virtual image of an existing installation (VMware has a tool, but you need another drive/computer to park the image on) and just drop it into the appropriate directory. *If it's MS Windows, the image may not recognise the virtual hardware and insist on being re-registered. Many OEM deals are cheap because the software is tied to the particular set of hardware on which it was initially installed – legally, if this is the case, you can only install the image on the machine from which it was taken. As a significant upgrade e.g. larger hard drive, BIOS update from someone other than the original manufacturer e.g. eSupport, change of MAC address ... can trigger a requirement to re-register, even complying with this restriction won't get round the need to re-register. Businesses don't have a problem as they have a bulk licence for x copies of MS Windows, with fewer restrictions on where/how they install them.*  
**Advantage:** All of your applications and settings get transferred too.  
**Problem:** Strictly, this is a one-way jump as having the original Windows installation and the image on the same machine, according to Microsoft, is 'two instances = two licences'.

### Method 3:

- ◆ Point your virtualization software at the existing Windows installation. VirtualBox can do this (but it's marked 'for experts only', as a mistake can trash the whole system). Again this is normally a one-way trip with a re-register for Microsoft Windows. However, according to recent postings, an 'either-or' set-up is possible – for VMware, at any rate, even though Microsoft added something in Windows XP SP3 (and later versions?) to prevent this being done. The 'hack' for XP and VMware is described here:  
<http://mazimi.wordpress.com/2007/07/11/getting-around-windows-activation-when-virtualizing/> – final comment  
**Advantages:** ability to run Windows on 'bare metal' as needed for some applications e.g. manufacturer's updates and only one set of OS and application updates to do.

# Virtualization

## What can you do?

- ◆ Practically anything you can do with a non-virtual system: run applications, use the network connection to access the internet, access peripherals e.g. printers ... ,
- ◆ Switch instantaneously from a window presenting, say, a MS-Windows desktop to your 'home' desktop. *A 'seamless' mode is included in the PUEL Guest Additions – just select from the File menu.*
- ◆ Virtualization is very useful for a developer who needs to test a website under development and check its behaviour when viewed using different OSs, as well as different browsers.
- ◆ If you set up a shared area, both host and guests can access the files it contains. See Section 4.6, Folder Sharing, on pages 68 ff of the User Manual. You cannot use a shared clipboard with MS Windows guests or drag-and-drop across the Virtual machine 's window boundary.

## Notes:

### Games

As the virtual image is using an 'imaginary' graphics card, the full range of hardware acceleration etc. beloved of gamers isn't available so games that rely on fast screen refreshes will work, but appear to be running on low specification hardware.

### VirtualBox – Compiz

A contribution to Full Circle No.29 contains the following help about using VirtualBox and Compiz with transparency together:

- ◆ Launch CompizConfig;
- ◆ Select the 'Window Rules';
- ◆ plugin in the 'Window Management' tab;
- ◆ For the 'No ARGB' rule, add the line:  
title=Sun VirtualBox

This will turn off the transparency for VirtualBox only, without changing the other Compiz settings.

### Latest versions (as at 31 October 2009):

- ◆ Karmic OSE VirtualBox version 3.0.8
- ◆ Karmic PUEL VirtualBox version 3.0.10

**USB** – DON'T need to be in USB user's group if using Guest Additions.