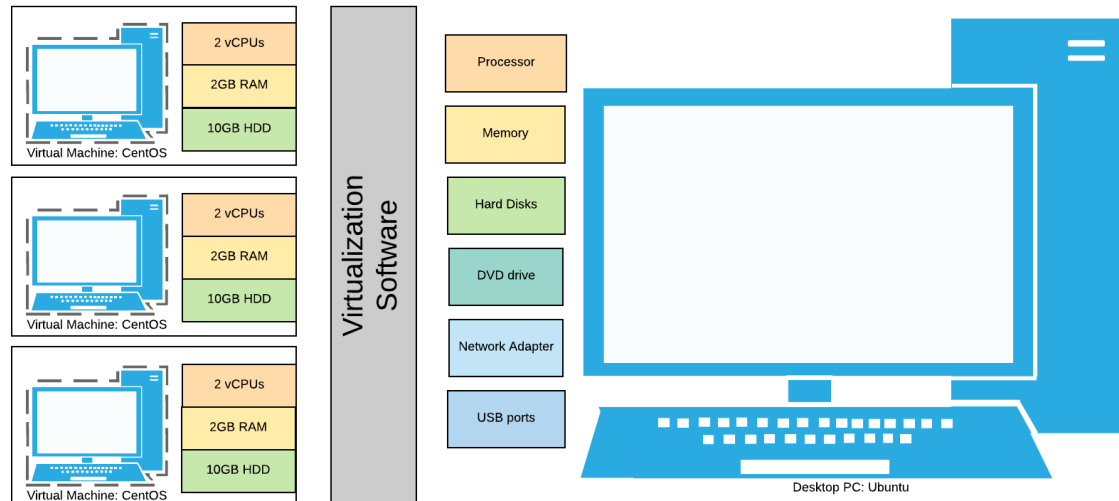


# HW Virtualization

## Hardware Virtualization: a Desktop Virtualization Example



This is a quick follow-up post to the rather popular [What Hardware Virtualization Really Means](#) post of mine. I constantly see “hw virtualization” queries to this blog, so would like to expand a bit on this topic and key terminology around hardware virtualization.

## HW virtualization definition

**HW virtualization** means the same thing as **hardware virtualization**: it’s a virtualization solution where end result (virtualization unit, usually a [Virtual Machine](#)) provides a completely isolated virtual representation of a hardware platform, running on top of specialised software and hardware.

The purpose of **hw virtualization** is to allow you run multiple virtual environments with virtual hardware, all sharing physical hardware available on your system. This achieves higher density: instead of running one OS environment per physical server, you can potentially run dozens of same or different OS environments on the same server.

Most common examples of such solutions are [desktop virtualization](#) products:

- [VirtualBox](#)
- Parallels
- VMware Workstation

All of these software solutions allow you to create a virtual machine by specifying desired number of virtual CPUs, virtual disks and allocated memory.

## Hardware Virtualization vs Hardware Emulation

If you look closely, most of the well known solutions today all provide virtualized environments of the same architecture: x86. This usually means the host (physical server) should be of the same hardware architecture. That's because virtualization software provides all the VMs with virtualized interfaces to real hardware on your system. The most important component of that is processor (CPU) and CPU instruction set.

If you are looking for a way to run different (non-x86) architectures inside virtual environments, you are going to need a different kind of software – called emulator. Such solutions exist and some indeed provide virtualization capabilities (let you run multiple emulated VMs on top of the same physical system), but at a performance cost: emulators have to emulate (implement using relatively slow software code) every CPU instruction, instead of virtualizing CPU – allowing virtual machines abstracted access to physical CPU where CPU runs its instructions much faster.

**HAV – Hardware Assisted**

# Virtualization

This is actually what most people mean when they say hw virtualization: they're referring to hardware assisted virtualization on desktop and server grade hardware.

Namely, both Intel and AMD processors have special functionality that allows for much more flexible and performant virtualization. Both operating systems and virtualization software check for such HAV support on your system and will usually fail to virtualize on older processors which don't have hardware assisted virtualization.

## Running a Typical x86 Virtual Machine

Once a VM is created, you can attach a virtual DVD drive to it and slot a virtual disk into it – an ISO image of your favourite operating system. Such virtual machine will then be booted and run the installation from ISO image, allowing you to complete OS install and end up with a virtual environment that looks and feels like a real desktop PC, only running rather slowly and inside a window of desktop virtualization software:

```
lgreys@localhost ~]$  
lgreys@localhost ~]$  
lgreys@localhost ~]$  
lgreys@localhost ~]$  
lgreys@localhost ~]$  
lgreys@localhost ~]$ cat /etc/redhat-release  
CentOS Linux release 7.4.1708 (Core)  
lgreys@localhost ~]$  
lgreys@localhost ~]$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/centos-root 6.2G  897M  5.4G  15% /  
devtmpfs        486M   0  486M   0% /dev  
tmpfs           497M   0  497M   0% /dev/shm  
tmpfs           497M  6.7M  490M   2% /run  
tmpfs           497M   0  497M   0% /sys/fs/cgroup  
/dev/sda1       1014M 125M  890M  13% /boot  
tmpfs           100M   0  100M   0% /run/user/0  
tmpfs           100M   0  100M   0% /run/user/1000  
lgreys@localhost ~]$ _
```

The screenshot also shows a standard Linux desktop environment with a taskbar at the bottom containing icons for network, volume, and other system utilities, along with the text 'Left ⌘'.

## VirtualBox 6 running CentOS 7.4 VM

Specifically for improving virtualization and performance of certain components like I/O, memory management and virtual networking, all of such virtualization solutions supply a collection of purpose-built drivers: [VMware Tools or Open-VM-Tools](#), for example. Installing these drivers lets you access VM disks from your host system, set advanced display resolutions, etc.

## Running Multiple Virtual Machines

The really attractive benefit of using hardware virtualization is running multiple VMs on the same physical system. Using desktop virtualization example diagram at the top of this post, you can see how a single desktop system running Ubuntu

Linux can host and run many virtual machines at once.

Here are examples of various virtualization processes happening in a typical [desktop virtualization](#) scenario:

- Processor virtualization – so instead of 1 physical CPU with 4 cores, you can present 1 or more virtual CPUs (vCPUs) to each of the virtual machines
- Disk is virtualized and shared – instead of presenting your whole desktop's disk to each VM, you can create relatively small virtual disks – one or more of them assigned and attached exclusively to each VM.
- RAM is shared – each VM is presented with a small portion of the actual RAM physically available on the desktop.
- Network virtualization – each VM has its own virtual network adapter (and you can have more than one), with static or DHCP IP addresses and various network access modes – shared or NAT, etc
- USB and DVD drives – you can map a physical resource like DVD drive or USB port into a particular VM – this means that the installation DVD with your favourite OS can be used to boot and install OS inside such a VM.

## See Also

- [VirtualBox](#)
- [Hardware Virtualization](#)
- [Virtual Machines](#)
- [Desktop Virtualization](#)
- [VMware Tools and OpenVM tools](#)