

VIRTUALIZATION IN LINUX

“Virtualisation, in computing, is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system, a storage device or network resources.” For the purposes of this article, I will deal primarily with operating system virtualisation.

Operating system virtualisation allows one OS (the guest) to be run completely within another OS (the host), like an application. For example, Windows can be run on Linux, or Mac OS X.

Virtualisation is one of the technologies that will shape the way the world computes in the future. It makes it possible to carve up, allocate and manage compute resources in new ways, allowing new degrees of freedom in systems design.

While virtualisation has already found a solid footing in the enterprise market, it is still a relatively new technology in the user space. I will compare three popular virtualisation products, and grade them on ease of use, performance, features, licensing and support. As I go through the comparisons, I will attempt to cover not only the specifics of the products themselves, but also the underlying concepts. Finally, I will analyse the scores of different products from the standpoint of different types of users.

The products I will be comparing are VMWare Player, VirtualBox and KVM. I have chosen these since they are all free of charge for personal use, and have large user communities. I have deliberately excluded VMWare ESX and Xen, since they are both targeted at enterprises. Due to this, despite the fact that they shine on expensive enterprise-grade servers, neither is designed to work with commodity hardware. ESX, for example, refused to install on my test system, altogether.

Some basics

Virtualisation being a relatively new technology, its language is still evolving. Most terms in the virtualisation space have several meanings, depending on which vendor you speak to. In order to avoid confusion, here are translations into English for some of the jargon that I will use:

- ♣ Virtual machine: A VM is a completely isolated instance of an operating system (guest) running inside another operating system (host).
- ♣ Hypervisor: A hypervisor, or virtual machine manager, is a software layer that presents the virtualised hardware platform on which the VM instances run. There are two types of hypervisors:
 - ♣ Type 1 hypervisors run on “bare metal” — that is, they do not require a host OS. In this case, the hypervisor acts almost as the OS itself.
 - ♣ Type 2 hypervisors require a host OS, and run as an application.

- ♣ Full virtualisation: In full virtualisation, every part of the underlying hardware is virtualised and presented to the guest. This means that the guest can be completely unaware that it is running in a VM and not a physical machine.
- ♣ Para-virtualisation: In para-virtualisation, the underlying physical hardware performs tasks that function poorly in a virtual environment. This requires the guest OS to be aware that it is running in a virtual instance. Some virtualisation products provide this functionality as a set of drivers to be installed into a guest OS, while others require the OS itself to be modified.
- ♣ Snapshots: A snapshot is a point-in-time capture of the VM. Different products classify snapshots differently. However, the most common understanding of the feature is that a snapshot preserves the machine in suspended animation — that is, the contents of memory and state of the CPU are written to disk, as is. When the machine is restored, the operation resumes from where it was left off. This allows point-in-time recovery, and an ability to roll back unwanted changes.
- ♣ Live migration: In the context of virtualisation, live migration is the seamless transfer of a running guest from one physical host to another. Live migration is an advanced feature, and many hypervisors impose restrictions on the source and destination system, most commonly that the CPUs of both need to be identical (homogeneous systems). Live migration between non-homogeneous systems is more complex, and consequently supported by fewer hypervisors.

Test system

For the purposes of the evaluation, I used an Ubuntu Server (Natty Narwhal) installation on a white-box desktop-class machine, with an AMD Phenom II X6 1090T hexa-core processor, 8 GB of RAM, and a Seagate 7200 RPM, 1 TB SATA hard drive.

Asking the right questions

One route to evaluating anything is to understand the need. The first question on this route is: “Why, if at all, should I use virtualisation?”

This question is important, because unless you understand your need, it is very simple to get stuck with the wrong solution — and nothing is worse than spending weeks tweaking your VMs to get them configured just enough to work, and then discovering that the feature you need is available out of the box in a different product.

Some reasons to virtualise:

- ♣ You need a different OS to run a particular application — for example, games that run only on Windows, code that can only be compiled on a native Linux machine, etc.
- ♣ The need to standardise environments across a group, such as a locked-down development environment for a contractor, several developers collaborating on an open source project, etc.
- ♣ Centralising, optimising and sharing resources, as in virtual desktops/servers sharing data-centre hardware.

The list goes on and on. Ultimately, there are as many use-cases for virtualisation as there are users. While it is possible to analyse one's use-case to the last CPU cycle, and choose the right product based on mathematics, it is usually overkill from the standpoint of the average user. It is often simpler to make a choice based on the user's skill level and patience.

The question then becomes: "How do you classify yourself as a user?" I usually see users in four categories, based on their profile (see the Box). Given that the user's profile usually correlates to the use-case, at least in a broad sense, it is simpler to choose the product that works best for the user.

Later in this article, I will apply this approach and rate the products, first objectively, and then relative to the user profile.

How do you classify yourself as a user?

The enthusiast

The enthusiast is usually at the forefront of the technology adoption curve, picking up the hottest new technologies from the bleeding edge. Enthusiasts are usually motivated by the possibility of bragging rights, and so will go to some lengths to learn and use a new piece of technology, even if they don't necessarily need it for any particular purpose.

The technology is an end in itself. Enthusiasts are typically willing to sit through hours of troubleshooting just to get a cool feature to work. Once they get it working, they like to talk about it and write articles for other users. Enthusiasts are usually motivated by features, performance, and very often, licensing. They will commend ease of use when they see it, but usually won't stop using a product just because it is tough to use.

The architect

Architects are usually visionaries. They look at new technology less emotionally than enthusiasts, and more from a practical standpoint of what it can accomplish for them. Architects are usually motivated by the same things as enthusiasts, with the difference that their priorities on different metrics are based on the job they need to accomplish. For example, an architect might choose a product with a great set of user-level features for a user-oriented project, and reject the same product for a different project where performance is more important.

The executive

The executive typically is not a technologist, but uses technology to supplement his primary function. For example, an accountant who owns a Mac may run a Windows VM to run Tally. Executives typically have a fixed set of requirements, and rely on architects and enthusiasts — architects more than enthusiasts — to tell them which product will work for

How do you classify yourself as a user?

them.

Executives typically look for ease of use first, with features a close second. Performance is “nice to have”, and licensing is usually not a concern for executives, unless it requires them to pay.

The follower

Followers use technology mostly because everyone else is using it and they are forced to, in order to comply. This bunch is primarily concerned with ease of use and minimal user involvement. On features, they look for things that make their life easier. On performance, they are usually willing to trade better performance for better usability. This group, too, is usually clueless about licensing.

Source : <http://www.opensourceforu.com/2011/09/vmware-player-virtualbox-kvm-virtualization-comparative-review/>