Editorial Board Thoughts: Just Like Being There, or How I Learned To Stop Coveting Bare Metal and Learned to Love My VM

Mark Cyzyk

Because this is a family program and because we are all polite people, I can't really use the term I want to here. Let's just say that I am an operating system [insert term here for someone who is highly promiscuous]. I simply love to install and play around with various operating systems, primarily free operating systems (OSes), primarily Linux distributions. And the more exotic, the better, even though I always dutifully return home at the end of the evening to my beautiful and beloved Ubuntu. In the past year or two I can recall installing (and in some cases actually using) the following: Gentoo, Mint, Fedora, Debian, moonOS, Knoppix, Damn Small Linux, EasyPeasy, Ubuntu Netbook Remix, Xubuntu, openSuse, NetBSD, Sabayon, SimplyMEPIS, CentOS, GeeXboX, and ReactOS. (Aside from stock Ubuntu and all things Canonical, the one I keep a constant eye on is moonOS [http://www.moonos.org/], a stunningly beautiful and eminently usable Ubuntu-based remix by a young artist and programmer in Cambodia, Chanrithy Thim.)

In the old days I would have rustled up an old, sloughed-off PC to use as an experimental "server" upon which I would unleash each of these OSes, one at a time. But those were the old days, and these are the new days. My boss kindly bought me a big honkin' Windows-based workstation about a year and a half ago, a box with plenty of processing power and memory (can you even buy a new workstation these days that's not incredibly powerful, and incredibly inexpensive?), so my need for hardware above and beyond what I use in my daily life is mitigated. Specifically, it's mitigated through use of virtual machines.

I have long used VirtualBox (http://www.virtualbox.org/) to create virtual machines (VMs), lopped-off hunks of RAM and disk space to be used for the installation of a completely different OS. With VirtualBox, you first describe the specifications of the VM you'd like to create—how much of the host's RAM to provide, how large a virtual hard disk, boot order, access to host CD drives, USB devices, etc. You click a button to create it, then you install an OS onto it, the "guest" OS, in the usual way. (Well, not exactly the usual way; it's actually easier to install an OS here because you can boot directly from a CD image, or iso file, negating the need to mess with anything so distasteful and old-fashioned and *outre* as an actual, physical CD-ROM.) In my experience, you can create a new VM in mere seconds; then it's all a matter of how difficult the OS is to install, and the Linux distributions are becoming easier and easier to install as the months plow on. At any rate, as far as your new OS is concerned, it is being installed on bare metal. Virtual? Real? For most intents and purposes the guest OS knows no difference.

In the titillatingly dangerous and virus-ridden cyberworld in which we live, I'll not mention the prophylactic uses of VMs because, again, this is a family program and we're all polite people. Suffice it to say, the typical network connection of a VM is NATed behind the NIC of the host machine, so at least as far as active network–based attacks are concerned, your guest VM is at least as secure as its host, even more so because it sits in its own private network space. Avoiding software-based viruses and trojans inside your VM? Let's just say that the wisdom passed down the cybergenerations still holds: When it rains, you wear a raincoat—if you see what I'm saying.

Aside from enabling, even promoting my shameless OS promiscuity, how are VMs useful in an actual work setting?

For one, as a longtime Windows guy, if I need to install and test something that is *NIX-only, I don't need a separate box with which to do so. (And vice versa too for all you Unix-weaned ladies and gentlemen who find the need to test something on a Rocker from Redmond.) If there is a software dependency on a particular OS, a particular *version* of a particular OS, or even if the configuration of what I'm trying to test is so peculiar I just don't want to attempt to mix it in with an existing, stable VM, I can easily and painlessly whip up a new instance of the required OS and let it fly. And deleting all this when I'm done is easily accomplished within the VirtualBox GUI.

Using a virtual machine facilitates the easy exploration of new operating systems and new applications, and moving toward using virtual machines is similar to when I first started using a digital camera. You are free to click click click with no further expense accrued. You don't like what you've done? Blow it away and begin anew.

All this VM business has spread, at my home institution, from workstation to data center. I now run both a development and test server on VMs physically sitting on a massive production server in our data center—the kind of machine that when switched on causes a brown-out in the tri-state area. This is a very efficient way to do things though because when I needed access to my own server, our system administrator merely whipped up a VM for me to use. To me, real or virtual, it was all the same; to the system administrator, it greatly simplified operations. And I may joke about the loud clank of the host server's power switch and subsequent dimming of the lights, but doing things this way has been shown to be more energy efficient than running a server farm in which each server

Mark Cyzyk (mcyzyk@jhu.edu) is the Scholarly Communication Architect in The Sheridan Libraries, Johns Hopkins University, Baltimore, Maryland.

sucks in enough juice to quench the thirst of its redundant power supplies. (They're redundant, they repeat themselves; they're redundant, they repeat themselves—so you don't want too many of them around slurping up the wattage, slurping up the wattage . . .)

Virtual machines: Zero-cost playgrounds for the promiscuous, and energy efficient, staff saving tools for system operations. What's not to like?

Throw dual monitors into the mix (one for the host OS; one for the guest), and it's just like being there.