Introduction

With the increasing processor speed and network performance enjoyed by even domestic computer users, more and more things that would have been considered impractical only a few years ago are becoming commonplace. A decade ago when a small technology company called VMWare started up in California, the idea of running a completely virtual computer as software within a physical computer was viewed as rather esoteric. After all, if you have a computer, why slow it down by adding a virtualization layer only to run what you'd be running anyway? The software you used needed the full power of the hardware to run, and as you could simply buy more machines to do more work if needed, what would be the point?

A decade later we all see the benefits of virtual machines. Hardware is so fast that modern machines can run many virtual machines without a major impact on overall performance, and the importance of software services is so significant that the security and reliability benefits of isolating them completely in virtual machines are clear.

However, pure virtualization has its problems, as it relies on some degree of hardware support* to function and is therefore exposed to instabilities caused by such close links to the physical machine. Emulators, by contrast, are virtual computers built entirely in software, and therefore have no specific requirements on the underlying hardware. This means the emulated machine is completely separated from the real hardware. Its presence on the system neither suffers nor causes any additional instabilities than the normal application software would. As running application software is the raison d'être of a computer in the first place, an emulator will always be the most stable and secure means to create virtual machines.

As with virtualization a decade ago, current critiques of emulation focus on the speed penalty incurred, which is often significant. However, history shows that speed issues are resolved by technological progress, but given the ever increasing complexity of modern hardware and software stacks, ever more difficult issues arise from subtle interactions between hardware, operating systems, and application software. Separating systems at a very low level in a very robust and secure way while still sharing the physical resource will therefore become increasingly necessary but increasingly difficult. Emulation offers the required degree of robustness, security, and flexibility, and so will become an increasingly compelling option.

There are a number of emulators currently available, and the most notable examples for emulating an x86 PC are Bochs and QEMU. Both have been developed over a number of years and are sufficiently accurate to boot modern operating systems and run application software. However, both are written in native code (C/C++) and need recompilation if they are to run on a new underlying hardware architecture/OS stack (i.e., a new type of host system). Furthermore, for very high-security applications, there is always the concern that the emulator has been tampered with for nefarious purposes, or to let guest code do nefarious things, or that

* At a bare minimum you need hardware that is the same as that being "virtualized." Thus products such as Xen and VMWare enable virtual x86 PCs to be created on x86 hardware only.

Download at WoweBook.Com