

As for performance isolation, it is difficult for an operating system kernel, which is an extremely complicated piece of software, to account for all the resources that are being used by a particular user. An example is the *fork bomb*, where a user starts an exponentially growing number of processes. This rapidly consumes all of the processor's resources and causes denial of service to other users. Hence multiuser systems typically require some amount of trust or etiquette between users, so that attacks like these are not carried out.

---

In Figure 7-1 we see two virtual machines running on top of a hypervisor. The first virtual machine runs Operating System 1 (e.g., Microsoft Windows) and two applications; the second virtual machine runs Operating System 2 (e.g., Linux), an application, and a service.

It turns out that virtualization is useful for other things as well. For example, in many data centers, a dedicated server is used for each application, such as a database or a web server, but each server uses only a fraction of its processor, memory, and bandwidth resources. Obviously, it would be better if these were consolidated onto fewer physical machines, saving power, space, and component maintenance. However, simply running the applications on the same operating system can give poor results. The various applications, when run together, might cause unpredictable poor performance. Worse, there is a risk of correlated failures, which occur when one application crashes and causes the others to crash as well. By placing each application in a virtual machine and then running them on top of a hypervisor, the hypervisor can protect the applications and ensure that each gets a fair share of the server's resources.

The idea of using virtualization for utility computing has gained currency in recent years. One of the best-known utility computing services is Amazon's EC2, which allows customers to create virtual machines that run on servers in Amazon's data centers. The customer is then charged for the processor time and network bandwidth that his virtual machine uses. In fact, the servers run Xen as their virtualization software, making it even closer to the Xenoservers vision (although it admits only a single service provider).

Virtualization has also had an influence on grid computing. Globus, the de facto standard middleware for grid computing, now supports *virtual workspaces*, which marry virtual machines with existing grid security and resource management protocols. An added benefit is that a virtual workspace—like any virtual machine—can be *migrated* to another physical location if conditions change.

The key advantage of virtual machines in the Xenoservers model is that they can be used to run popular commodity operating systems and existing applications. In practice, this means running on the dominant x86 architecture, which presents the hypervisor developer with several challenges.