network device might be attached to a software network bridge, which is itself attached to a real network card. The ring buffer abstraction ensures that the frontend and backend are totally decoupled. One backend can support frontends from Linux, BSD, or Windows, whereas the same frontend can be used with various backends, so features such as copy-on-write, encryption, and compression can be added transparently to the guest. Like the Internet Protocol, the Xen split device model can operate on a vast array of hardware, and it supports a multitude of higher-level clients, as shown in Figure 7-3.

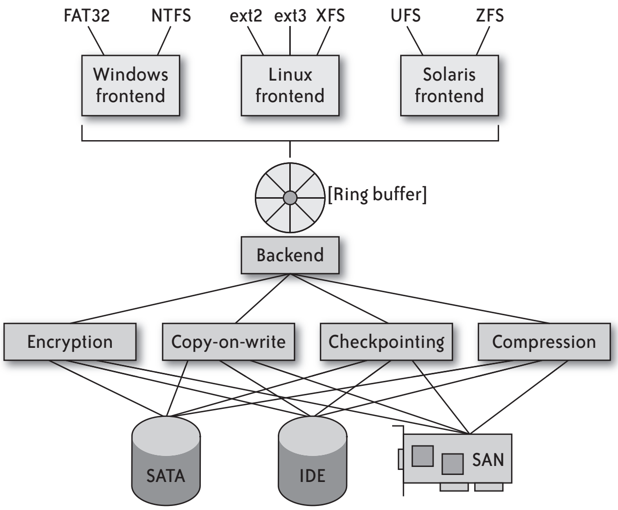

FIGURE 7-3. Hourglass architecture of a split block device

Paravirtualization encompasses far more than these examples. For example, the meaning of time changes when an operating system can be switched out from the CPU, and Xen introduces a virtual time concept to ensure that operating systems still behave as expected. There are many more virtual devices, and the paravirtualization of memory makes use of many more optimizations to ensure efficient performance. For more details on these, see the "Further Reading" section at the end of the chapter.

The Changing Shape of Xen

The traditional representation of a Xen-based system shows several virtual machines (known in Xen as *domains*) sitting on top of the hypervisor, which itself sits directly on the hardware (Figure 7-4). When it boots up, the hypervisor launches a special domain, known as *domain zero*. Domain zero has special privileges that allow it to manage the rest of the system, and it is analogous to a root or administrator process in a regular operating system. Figure 7-4 shows a typical Xen-based system, with domain zero and several *guest domains* (known as *DomUs* in Xen jargon) running on top of the Xen hypervisor.