Xen and the Art of Virtualization

Patrick Araya

1/29/13
The key for creating a virtualized environment is balancing the tradeoff between allocating resources between the guest and host machines. There are several different methodologies to consider when choosing to set up the environment. Some require complete compatibility with certain binaries to ensure a program can run, while others are inclined to performance while sacrificing security and availability. Most however, are only best-effort. The paper attempts to create a Xen environment where neither performance nor functionality is surrendered and all variations of operating systems can be ported with minimal effort.

Modern computing technologies has helped develop virtualization with its rising interest. However, there are several challenges when creating virtual environments. Machines must be separated from one another as not to hinder on the privacy or trust issue from another station and to not impede on the performance of the other VM. Machines must also be able to support end users’ applications. If the application is not supported, the entire system would be useless to the user. The most important issue for developing a virtualized environment is performance. If a system is too slow to run an application then the user will not like the system enough to use it.

There are certain drawbacks to some virtualized environments in that the VMM is required or called upon to handle a task from the VM. Paravirtualization overcomes this hindrance by compiling an operating system prior to installation, which in turn provides better performance. This also provides greater resource isolation from the VMs. The biggest obstacle when creating a paravirtualized environment is memory allocation. A page table is taken from a VM’s memory and registered with Xen each time a new page table is required. Xen now controls the write privileges for the page-table. The CPU in a virtual environment is protected by the hypervisor and privilege levels. The hypervisor is run below the operating system of a guest. This is why the guest OS must run at a privilege level lower than the hypervisor. There are four hardware based privilege levels or “rings,” with only two that are used.
Level 0 runs the OS code while level 3 is reserved for applications. Levels one and two are vestigial and are no longer required necessary.