Real-time Performance with vSphere: How It's Done

By Robert Campbell, Staff Technical Alliance Manager, VMware

A few weeks ago, Johanna Holopainen blogged about virtualizing voice and other real-time applications. I want to continue that discussion by looking at the performance requirements for real-time applications in a virtualized environment.

First, let's be clear about the term "real time," because it means different things in different settings. Computer scientists use "hard real-time" for a system that must perform within a given time frame (sub milliseconds) to avoid catastrophic results: think pacemakers, anti-lock brakes, and aircraft control systems. By that standard, voice over IP (VoIP) can be thought of as "soft real-time" or "near real-time": nobody dies if a few voice packets arrive late (although if you're responsible for a large VoIP implementation, too much latency may not be healthy for your career). In the rest of this blog, when I say "real-time," think "near real-time--tens of milliseconds"

Real-time and virtualization are somewhat at odds with each other. Virtualization spreads computing resources across multiple virtual machines as a way to increase utilization and flexibility. Real-time applications dedicate specific resources to reduce computing overhead and ensure low latency. Not too long ago, running real-time applications in a virtualized environment was risky. However, as we will see, it's different now, thanks to developments in hypervisor design and hardware-assisted memory management.

Hypervisor Design

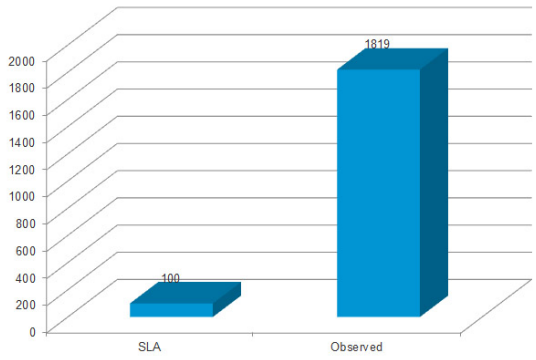
First-generation hypervisors—VMware's included—focused on IT efficiency, resource utilization, availability and other business-oriented benefits appropriate to enterprise applications. While they achieved these goals in spectacular fashion, latency and other performance-related parameters took a back seat. That has to do primarily with the scheduler, the hypervisor module that allocates resources to virtual machines. As workload increases, the scheduler spends more time allocating CPU and memory. That's okay for enterprise applications, but isn't optimum for real-time.

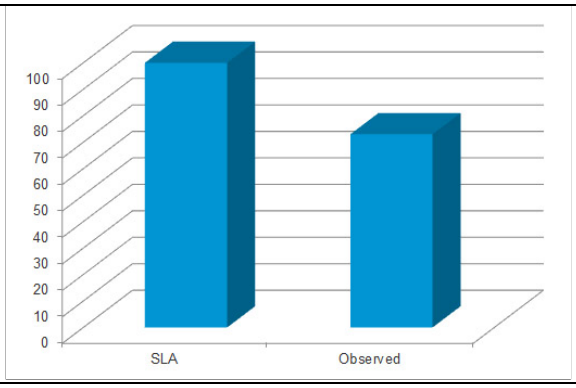
Starting with vSphere 4, VMware redesigned the hypervisor scheduler to support real-time applications. For voice, latency has been cut by 4-5X just due to hypervisor improvements. I'll talk more about hypervisor design technology in a future blog, but here's a white paper that covers the subject in detail.

Hardware-assisted Memory Management

Memory management is one of the primary bottlenecks affecting performance of virtual machines. Intel and AMD have added hardware assistance for memory management to their server CPU architectures. This feature—called Extended Page Tables (EPT) in Intel Nehalem processors and Rapid Virtualization Indexing (RVI) in AMD devices—effectively reduces the memory footprint, which cuts latency for real-time applications.

The figure below shows benchmark test results for two different Intel CPUs—the Intel 5450 and Intel 5560 (Nehalem) with EPT—running an MMU-intensive workload. The improvement with EPT is impressive: Worst-case latency is well below the target level SLA of 100 milliseconds (note the 20X scale difference between the two diagrams).





Intel 5450	Intel 5560 (Nehalem) with EPT

Typical worst-case latencies for MMU-intensive application

Takeaway

So here's what you need to know about running voice applications in a virtual environment:

- The hypervisor scheduler must be optimized for real-time (which is the case for vSphere 4.0 and later)
- The processor must have hardware-assisted memory management (EPT for Intel, RVI for AMD)

In case you think this is just a theoretical discussion, check out Mitel Virtual Solutions, a family of unified communications (UC) solutions optimized for vSphere environments. Mitel is using virtual appliances to good advantage—and that's another topic for a future blog.

Have your own experience running real-time applications on VMware? Tell us about it.

(about 575 words)