DOI:10.1145/1831407.1831428

Technical Perspective A VM 'Engine' That Makes a Difference

By Carl Waldspurger

THE PAST DECADE has witnessed a renaissance in server virtualization, which is transforming enterprise computing by offering new capabilities and efficiencies. The following paper by Diwaker Gupta et al. presents a novel approach for significantly improving the efficiency of virtualized servers. Their "Difference Engine" eliminates memory redundancy by exploiting similarity both within and across virtual machines.

A virtual machine (VM) is a software abstraction that behaves like hardware. The classic definition by Popek and Goldberg is "an efficient, isolated duplicate of a real machine." For example, a VM that presents the illusion of being a physical x86 server may run an unmodified operating system designed for that platform, such as Windows or Linux. Neither the OS nor its users need be aware they are interacting with a VM instead of dedicated hardware.

Little more than a decade ago, virtual machines were considered a fairly exotic mainframe technology. Today, VMs are pervasive in corporate datacenters, and serve as the foundation for cloud-computing platforms. The commercial success of virtual machines has influenced the design of high-volume processor architectures, which now contain special-purpose hardware to accelerate virtualization.

Why have VMs proliferated so rapidly? One reason is that virtualization is an extremely versatile technology.

There is a well-known adage: "All problems in computer science can be solved by another level of indirection." The virtualization software layer, known as a hypervisor, provides this level of indirection, decoupling an OS and its applications from physical hardware. Eliminating the traditional "one machine, one OS" constraint opens up numerous possibilities.

Initially, the most compelling use of VMs was basic partitioning and server consolidation. In typical unvirtualized environments, individual servers were grossly underutilized. Virtualization allowed many servers to be consolidated as VMs onto a single physical machine, resulting in significantly lower capital and management costs. This ability to "do more with less" fueled the rapid adoption of virtualization, even through economic downturns.

As virtualization became more mainstream, innovations arose for managing distributed systems consisting of many virtualized servers. Since VMs are independent of the particular hardware on which they execute, they are inherently portable. Live, running VMs can migrate between different physical servers, enabling zero-downtime infrastructure maintenance, and supporting automated dynamic load balancing in production datacenters and clouds.

Additional virtualization features leverage indirection to offer capabilities beyond those of physical platforms. By interposing on VM operations transparently, no changes are required to the software running within the VM. Examples include improving security by adding checks that cannot be defeated by compromised software within the VM, and replicating VM state across physical machines for fault tolerance.

While core virtualization techniques are now reasonably mature, researchers continue to develop innovative ways to optimize VM efficiency and improve server utilization. Today, limited hardware memory often constrains the degree of server consolidation on modern machines equipped with many processor cores. The Difference Engine cleverly exploits the extra level of indirection in virtualized memory systems to reduce the memory footprint of VMs. Since higher consolidation ratios translate directly into cost savings, such techniques are incredibly valuable.

Due to consolidation, many VMs on the same physical machine typically run similar OS instances and applications, or contain common data. The Difference Engine extends the hypervisor with several mechanisms that reclaim memory by eliminating redundancy. First, when identical memory pages are found, they are deduplicated by retaining only a single instance that is shared copy-on-write, similar to the page-sharing feature that we introduced in VMware's hypervisor.

However, the Difference Engine goes much further, taking advantage of deduplication opportunities that are left on the table when sharing is restricted to completely-identical pages. By observing that many more pages are nearly identical, sharing at sub-page granularity becomes very attractive. Candidates for sub-page sharing are identified by hashing small portions of pages, and patches are generated against reference pages to store near-duplicates compactly. When pages are not sufficiently similar, a conventional compression algorithm is applied to wring out any remaining intra-page redundancy.

By combining these mechanisms to eliminate full-page, sub-page, and intra-page redundancy, the Difference Engine achieves impressive space savings—more than twice as much as full-page sharing alone for VMs running disparate workloads. Of course, these savings aren't free; compressed pages and sub-pages still incur page faults, and hashing, patching, and compression are compute-intensive operations.

But given current trends, it's a safe bet that spare processor cycles will be easier to find than spare memory pages. The emergence of dense flash memory, phase-change memory, and other technologies will surely shift bottlenecks and trade-offs, ensuring this research area remains interesting.

Given the long history and extensive literature associated with both virtualization and memory management, it's refreshing to find a paper that is both stimulating and practical. As virtual machines become increasingly ubiquitous, I'm confident that similar ideas will be leveraged by both commercial and research hypervisors. I strongly urge you to get a glimpse of this future now by reading this paper. C

Carl Waldspurger (carl@vmware.com) is a Principal Engineer at VMware, Palo Alto, where he oversees core resource management and virtualization technologies.