Xen and the Art of Virtualization

CSE231

Nayan Sanjay Bhatia

Motivation

- X86 doesn’t support full virtualization.
- Situations for host OS to see real as well as virtual resources

1. Providing both real and virtual time allows a guest OS for time sensitive task ,handle tcp timeouts and RTT estimates.
2. Exposing machine addresses to improve performance via superpages.

Xen- Paravirtualization

- Paravirtualization: Guest OS is modified to run on Xen Hypervisor for big improvement in performance and VMM simplicity.
- XenoLinux is modified Linux OS that runs on Xen hypervisor
- ABI (Application Binary Interface) need not change so no change to guest application
- Benefit: Better performance than binary translation

- Disadvantage: requires source code change to OS

Design Principle of XEN

- Support for unmodified application binaries.
- Supporting full multi-application operating systems
- Paravirtualization is necessary to obtain high performance and strong resource isolation on uncooperative machine architectures such as x86.
- Even on cooperative machine architectures, completely hiding the effects of resource virtualization from guest OSes risks both correctness and performance.

Xen Architecture

- Type 1 hypervisor: runs directly on the hardware
- Xen sits in ring 0, guest OS in ring 1
- Guest OS traps to Xen to perform privileged actions
- A guest VM is called a domain
- Special domain called dom0 runs control /management software
- The configuration software is not part of hypervisor making it lightweight

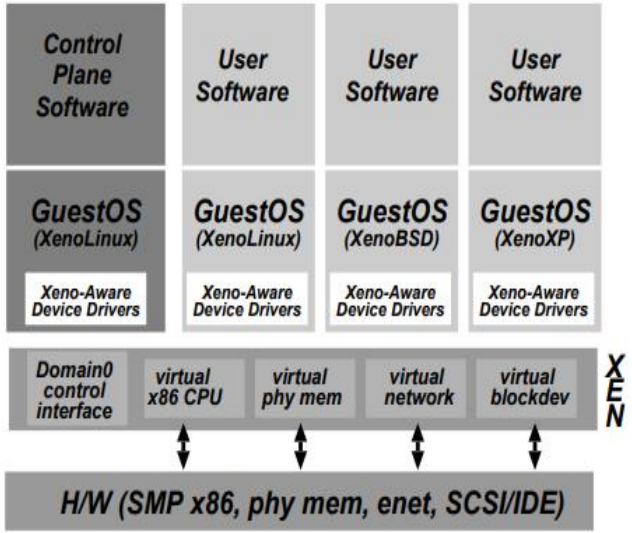


Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including *Domain0* running control software in a XenoLinux environment.

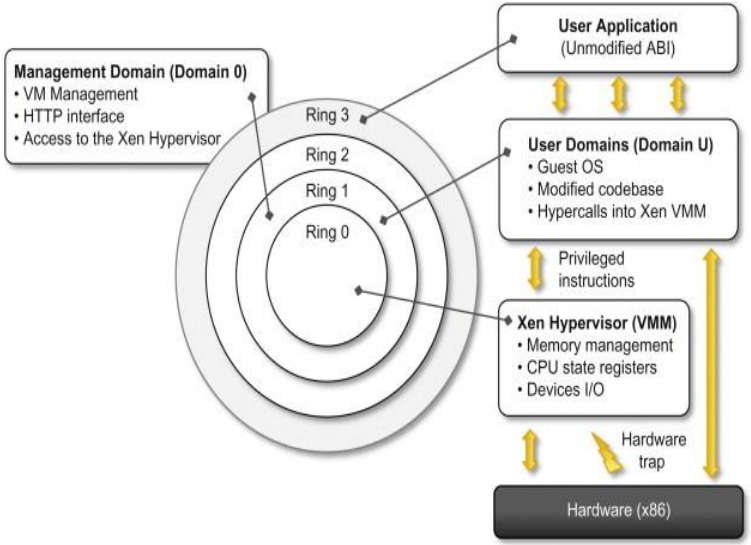
Memory Management	
Segmentation	Cannot install fully-privileged segment descriptors and cannot overlap with the top end of the linear address space.
Paging	Guest OS has direct read access to hardware page tables, but updates are batched and validated by the hypervisor. A domain may be allocated discontiguous machine pages.
CPU	
Protection	Guest OS must run at a lower privilege level than Xen.
Exceptions	Guest OS must register a descriptor table for exception handlers with Xen. Aside from page faults, the handlers remain the same.
System Calls	Guest OS may install a 'fast' handler for system calls, allowing direct calls from an application into its guest OS and avoiding indirecting through Xen on every call.
Interrupts	Hardware interrupts are replaced with a lightweight event system.
Time	Each guest OS has a timer interface and is aware of both 'real' and 'virtual' time.
Device I/O	
Network, Disk, etc.	Virtual devices are elegant and simple to access. Data is transferred using asynchronous I/O rings. An event mechanism replaces hardware interrupts for notifications.

Table 1: The paravirtualized x86 interface.

CPU virtualization in Xen

- Guest OS modified to not provoke any privilege instruction.
- All privilege operation traps to ring 0
- Hypercalls - similar to system call. Guest OS voluntarily invokes Xen to perform privilege operation.
- Synchronous: Guest pauses while Xen services the hypercall

- Asynchronous Event mechanism: communication from Xen to domain
- Much like interrupts from hardware to kernel.
- Used to deliver hardware interrupts and other notification to domain
- Domain registers event handler callback functions



Page fault and system calls

- Frequent in nature to affect performance.
- System calls: Guest registers a “fast interrupt handler” and it is validated by Xen for install
- Page fault: Address is in CR2 and needs to be propagated by Xen.

Memory Management in virtualized environment

- A software managed TLB or a tagged TLB can be virtualized easily.
- X86 has none of these feature.
- X86- hardware managed TLB
- Flushed upon context switch

Hence...

- Guest OS is responsible for allocating and managing hardware page
- Xen exist in a 64 MB section at top of every address space avoiding TLB flushes. This address region is not used by common x86 ABI hence wont break application compatibility.
- Validation by Xen. This can be a performance overhead.

I/O virtualization in Xen

- Shared memory rings between guest domain and Xen/domain0.
- I/O requests placed in shared queue by guest domain.
- Request handled by Xen/domain0, placed in ring.
- Descriptors in queue: pointers to request data (DMA buffers with data for writes, empty DMA buffer for read, etc)

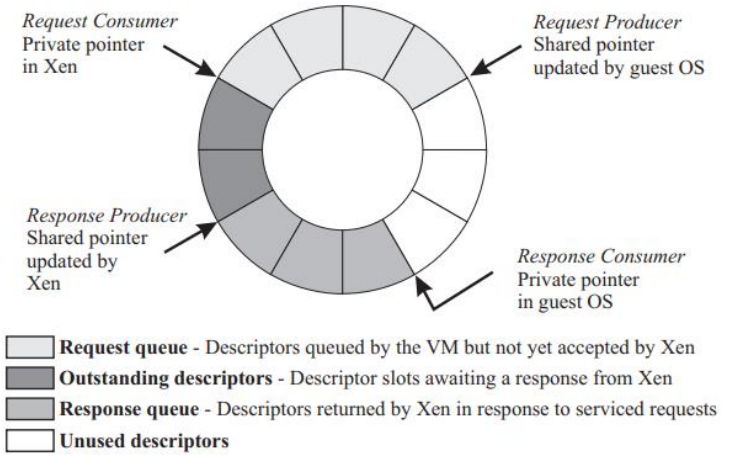


Figure 2: The structure of asynchronous I/O rings, which are used for data transfer between Xen and guest OSes.

Performance

- Xen performs well – Multiple domain can be hosted without any noticeable loss of performance.
- Tests demonstrate that 128 domains can be run with only 7.5% loss of throughput relative to standalone Linux.

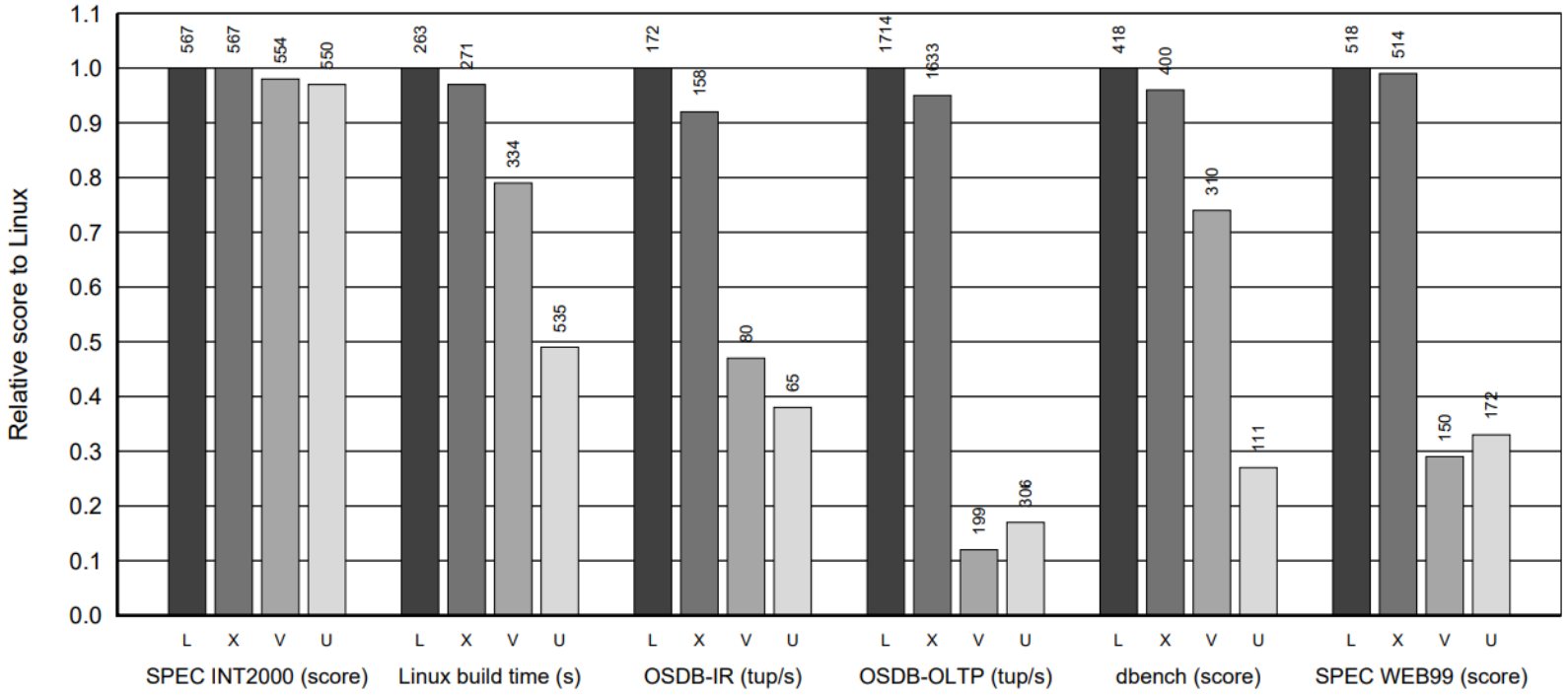


Figure 3: Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).

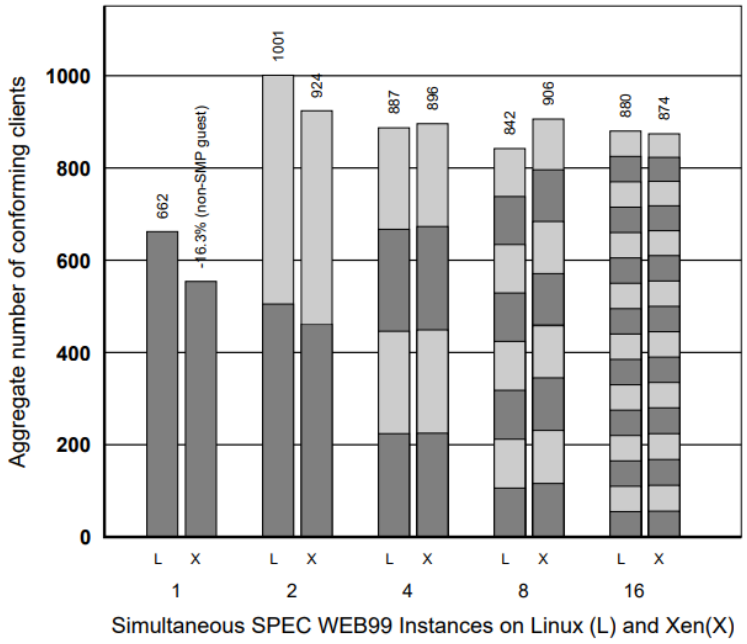


Figure 4: SPEC WEB99 for 1, 2, 4, 8 and 16 concurrent Apache servers: higher values are better.

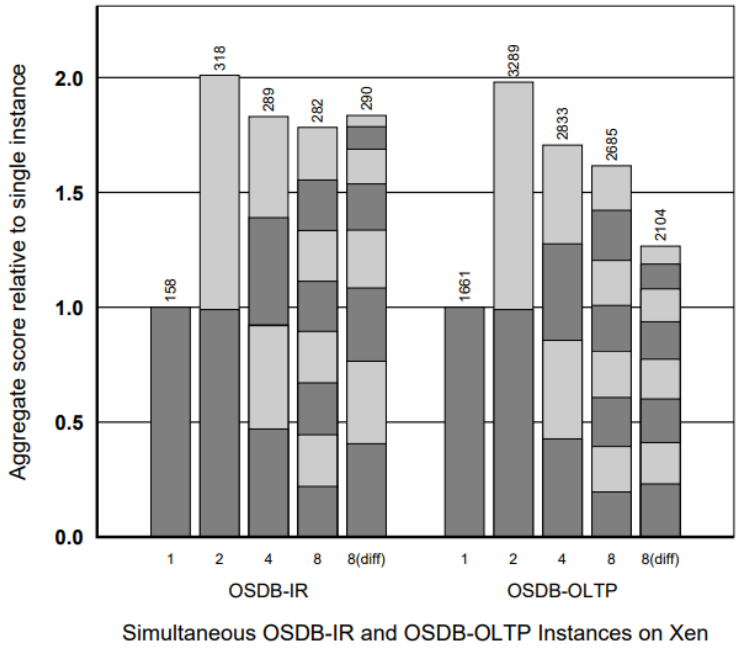


Figure 5: Performance of multiple instances of PostgreSQL running OSDB in separate Xen domains. 8(diff) bars show performance variation with different scheduler weights.

Impact

- Administrators can "live migrate" Xen virtual machines between physical hosts across a LAN without loss of availability.
- Like the exokernel, the paper stresses on the separation between mechanism from policy. The Xen provides only basic control operations and does not provide policy decisions such as packet filtering just as the exokernel avoids these policy decisions.
- Since paravirtualization doesn't require processor extensions, like Intel VT and AMD-V, paravirtualization can be deployed on hardware platforms that don't offer hardware-assisted virtualization.
- Xen is still used in Amazon AWS and Citrix XenServer.

Discussion and questions

1. Can collaboration between guest os scheduler and hypervisor os scheduler help in scheduling more efficiently?
2. Hardware-assisted virtualization can solve a lot of issues in a fully virtualized environment. So is paravirtualization still the right choice?
3. Similarities between philosophies of Xen and Exokernel

References

- https://www.youtube.com/watch?v=2moUsgMOie4
- https://www.youtube.com/watch?v=4XGbDWbEkU0
- https://slidetodoc.com/xen-and-the-art-of-virtualization-cse291-cloud/
- https://sites.google.com/site/masumzh/articles/x86-architecture-basics/x86-architecture-basics
- https://ucsc-cse231-21.hotcrp.com/doc/ucsc-cse231-21-paper14.pdf