Xen, an x86 virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality.

full virtualization vs paravirtualization

We distill the discussion so far into a set of design principles:

1. Support for unmodified application binaries is essential, or users will not transition to Xen. Hence we must virtualize all architectural features required by existing standard ABIs.

2. Supporting full multi-application operating systems is important, as this allows complex server configurations to be virtualized within a single guest OS instance.

3. Paravirtualization is necessary to obtain high performance and strong resource isolation on uncooperative machine architectures such as x86.

4. Even on cooperative machine architectures, completely hiding the effects of resource virtualization from guest OSes risks both correctness and performance.

guest operating system to refer to one of the OSes that Xen can host

domain to refer to a running virtual machine within which a guest OS executes

The Virtual Machine Interface

- Memory management
  - guest OSes are responsible for allocating and managing the hardware page tables, with minimal involvement from Xen to ensure safety and isolation
  - Xen exists in a 64MB section at the top of every address space, thus avoiding a TLB flush when entering and leaving the hypervisor
- CPU
  - In order to protect the hypervisor from OS misbehavior guest OSes must be modified to run at a lower privilege level. Run VMM at ring 0, OS at ring 1 (app stays at ring 3)
  - Privileged instructions are paravirtualized by requiring them to be validated and executed within Xen
  - Typically only two types of exception occur frequently enough toaffect system performance: system calls and page faults.
  - Safety is ensured by validating exception handlers when they are presented to Xen.
- Device I/O
  - Rather than emulating existing hardware devices,Xen exposes a set of clean and simple device abstractions.
  - I/O data is transferred to and from each domain via Xen, using shared-memory, asynchronous buffer-descriptor rings.
  - Xen supports a lightweight eventdelivery mechanism which is used for sending asynchronous notifications to a domain.

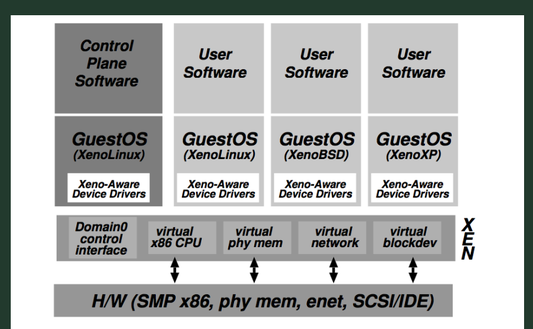


**Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including *Domain0* running control software in a XenoLinux environment.**

Control and Management → Domain0

- Control Transfer: Hypercalls and Events
  - synchronous calls from a domain to Xen may be made using a hypercall
  - notifications are delivered to domains from Xen using an asynchronous event mechanism
- Data Transfer: I/O Rings

DETAILED DESIGN

- CPU scheduling — Borrowed Virtual Time (BVT) scheduling algorithm
- Time and timers — Xen provides guest OSes with notions of real time, virtual time and wall-clock time.
- Subsystem Virtualization
  - Virtual address translation
  - Physical memory
  - Network
  - Disk

参考资料：
http://pages.cs.wisc.edu/~remzi/Classes/838/Spring2013/Notes/xen.pdf
https://zhu45.org/posts/2019/Mar/11/xen-and-the-art-of-virtualization/
https://www.cse-class.org/slides/2016-04-20-xen/