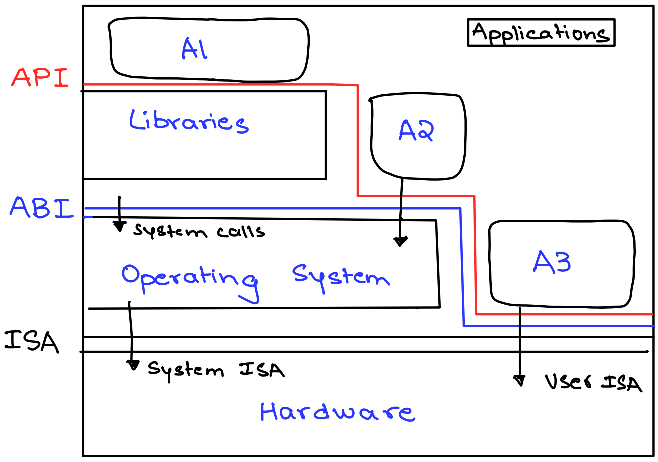
UNIT 3

LAYERING

- managing system complexity - layering
- minimizes interactions
- 2 modes → user & privileged



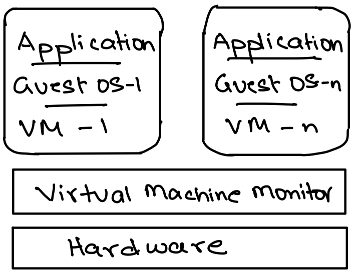
- 1st interface – ISA – boundary b/w s/ware & h/ware – processor's set of instructions
- ABI (Appⁿ Binary Int.) – H/ware access to the appⁿ's & library modules. Doesn't include privileged instructions.
- API – defines set of instr the h/ware was designed to execute – gives appⁿ access to the ISA.

CONDITIONS FOR EFFICIENT VIRTUALIZATION

1. A program running under the VMM must exhibit behaviour essentially similar to that demonstrated when the appⁿ runs directly on an equivalent machine.
2. The VMM should be in complete control of the virtualized resources.
3. A statistically significant fraction of the machine instructions should be executed without the intervention of the VMM.

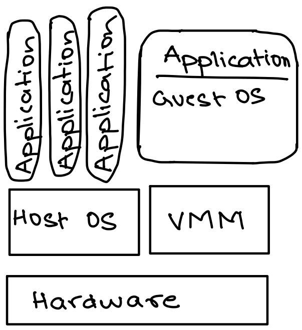
TRADITIONAL

- Thin layer of s/ware that runs directly on the host machine's h/ware
- Main adv. is performance



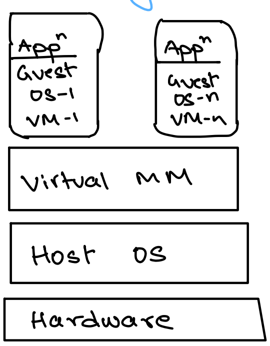
HYBRID

- Shares the h/ware w/ the existing OS.



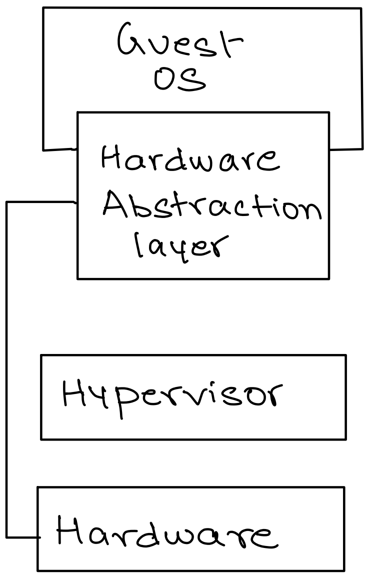
HOSTED

- The VM runs directly on top of the existing OS.



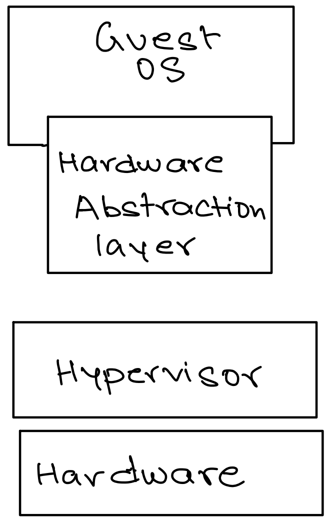
FULL VIRTUALIZATION

- Each VM runs on an exact copy of the actual h/ware.
- Requires a fully virtualizable architecture; the h/ware is fully exposed to the guest OS, that runs unchanged.



PARAVIRTUALIZATION

- VM runs on a modified copy of the actual h/ware.
- It is done as some architectures are not easily virtualizable.
- Demands that the guest OS be modified to run under the VMM.



PROBLEMS FACED BY VIRTUALIZATION OF THE x86 ARCHITECTURE

① RING DEPRIVILEGING

VMM forces the guest s/ware, the OS, & the apps to run at a privilege level > 0 (not in 64 bit)
solns → (0/1/3) mode
→ (0/3/3) mode

② RING ALIASING

Problem when a guest OS is forced to run at a privilege level that it wasn't designed for.

③ ADDRESS SPACE COMPRESSION

VMM uses parts of the guest address space to store system DSs. These DS must be protected, but the guest s/ware must have access to them.

④ INTERRUPT VIRTUALIZATION

VMM generates a "virtual interrupt" in response to a physical interrupt & delivers it to the target guest OS. Guest OS's have the ability to mask interrupts – complicates the VMM & increases overhead.

⑤ ACCESS TO HIDDEN STATE

Elements of the system state are hidden; there is no mechanism for saving & restoring hidden components after a context switch from 1 VM to another.

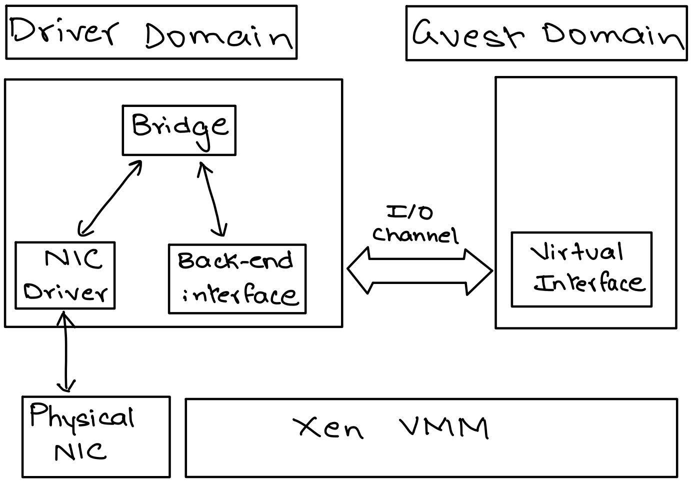
⑥ FREQUENT ACCESS TO PRIVILEGED RESOURCES INCREASES VMM OVERHEAD

The* TPR is frequently used by a guest OS. The VMM must protect access to this reg. & trap all attempts to access it. This can cause performance degradation.

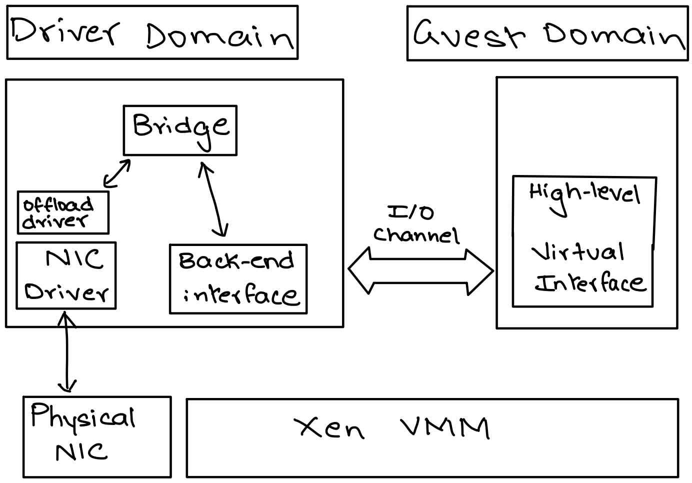
*TPR – Task Priority Register

XEN ARCHITECTURE

ORIGINAL



OPTIMIZED



XEN NETWORK OPTIMIZATIONS

1 THE VIRTUAL INTERFACE

- The original " " network provides the guest domain w/ the abstraction of a simple low-level network interface supporting sending & receiving primitives.
- This design supports a wide range of physical devices attached to the driver domain but doesn't take advantage of the capabilities of some physical devices.
- These features are supported by the high-level virtual interface of the optimized system

2 THE I/O CHANNEL

- Rather than copying a data buffer holding a packet, each packet is allocated in a new page & this physical page is remapped to the guest domain.
- This strategy contributes to a better than 4x increase in the send data rate.

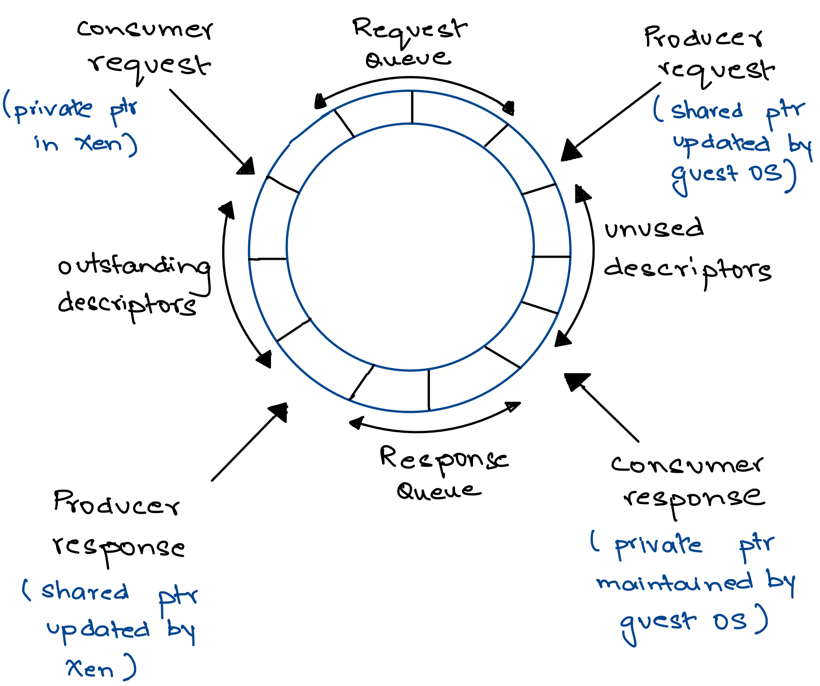
3 VIRTUAL MEMORY

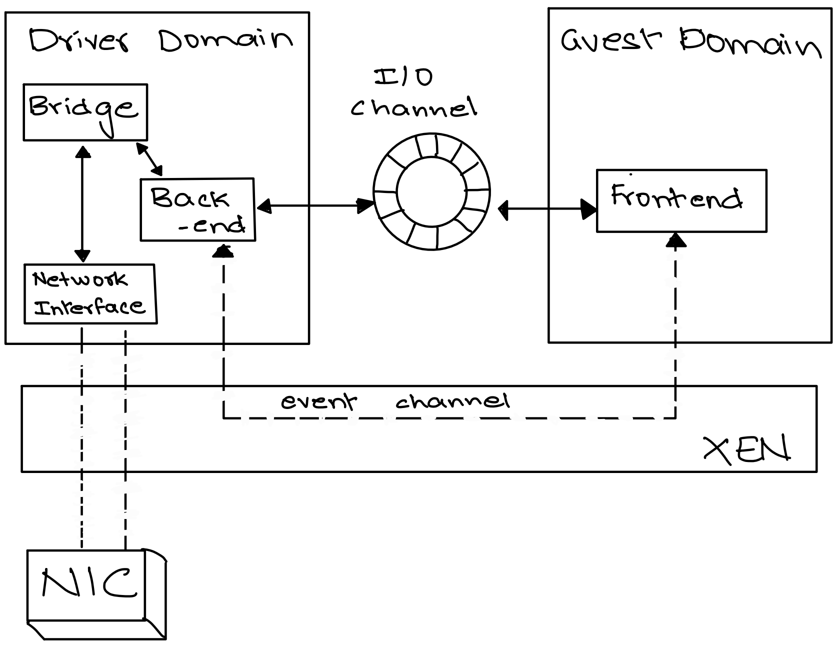
- " " in Xen 2.0 takes advantage of superpage & global page-mapping hardware features.
- A superpage increases the granularity of dynamic address translation – a superpage entry covers 1024 pages of physical memory, & address translation maps a set of contiguous pages to a set of contiguous physical pages.
- This reduces the no. of *TLB misses.
- The optimized version uses a special memory allocator to avoid the problem where the system is forced to use traditional page-mapping rather than superpage mapping

*Translation lookaside buffer: memory cache used to reduce the time taken to access a user memory location.

XEN ZERO-COPY SEMANTICS FOR DATA TRANSFER USING I/O RINGS

CIRCULAR RING OF BUFFERS





* IR → Instruction Register
* RSE → Register Stack Engine

vBlades project

- The Itanium processor supports 4 privilege rings — PL0, PL1, PL2, PL3.
- Privileged instructions can only be executed by the kernel at PL0.
- Appⁿs run at PL3 & can execute only non-privileged instructions.
- The VMM uses ring compression & runs itself at PL0 & PL1 while forcing a guest OS to run at PL2.
- A problem — privilege leaking — several nonprivileged instructions allow an appⁿ to determine the current privilege level (CPL).
- Itanium was selected because of its multiple functional units & multithreading support.

- CPU Virtualization:
  when a guest OS attempts to execute a privileged instruction the VMM traps & emulates the instruction.
  - Complication — Itanium doesn't have an *IR & the VMM has to use state info to determine whether an instr. is privileged.
  - Complication — caused by *RSE, which operates concurrently w/ the processor & may attempt to access memory & generate a page fault.

- A no. of privileged-sensitive instr. behave differently as a function of the privilege level. The VMM replaces each one of them w/ a privileged instr. during the dynamic transformation of the instruction stream.

contd.

contd.

- Memory virtualization is guided by the realization that a VMM should not be involved in most read & write operations to prevent a significant degradation of performance, but at the same time the VMM should exercise tight control & prevent a guest OS from acting maliciously.
- The vBlades VMM doesn't allow a guest OS to acess memory directly. It inserts an additional layer of indirection called metaphysical addressing b/w virtual & real addressing.
- A guest OS is placed in metaphysical addressing mode. If the address is virtual, the VMM first checks if the guest OS is allowed to access that address, & if it is, it provides the regular address translation. If the address is physical, the VMM is not involved.