Virtualiztion

Blackhoodie 2017

What

- Typically refers to the creation of virtual machine that can emulate or simulate all of the hardware resources, including processors, memory, storage, and network connectivity.
- A logical representation of a computer in software.

Requirements

- Provide an equivalent environment
- Secured control of virtualized resources
- At least similar performance

Why

- Increase the utilization of costly hardware resources
- Teach\Test\Research
- Flexibility
- Duplicate environments easily
- Console emulation

- Cloud-based solution

Types of virtualization

- Hardware virtualization
- Operating-system-level virtualization
- Application virtualization
- Memory \ Storage \ Data virtualization
    - Including distributed file systems

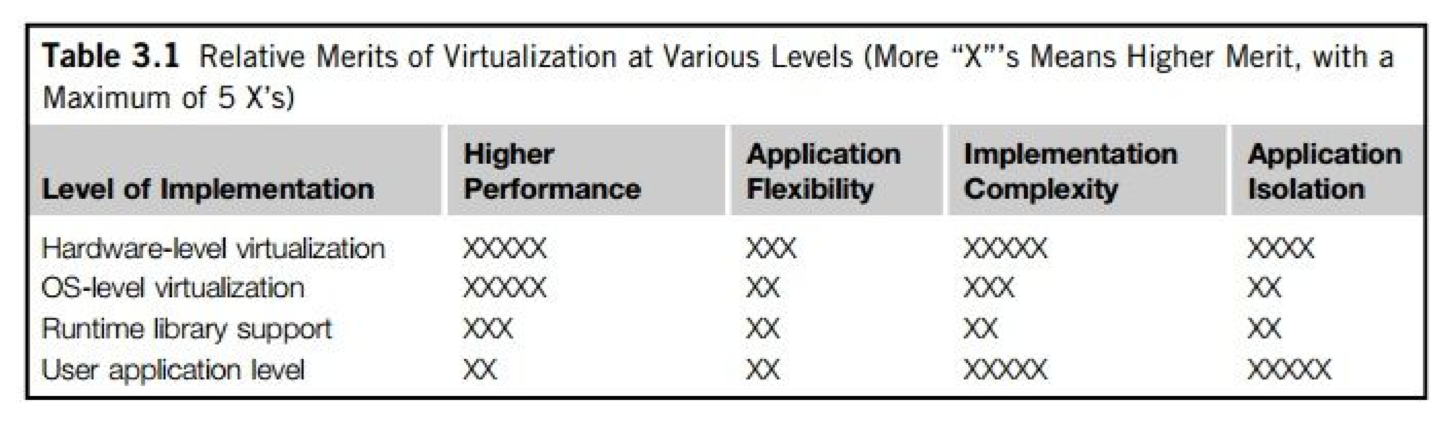


Table 3.1 Relative Merits of Virtualization at Various Levels (More "X"'s Means Higher Merit, with a Maximum of 5 X's)

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

Types of Hardware Virtualization

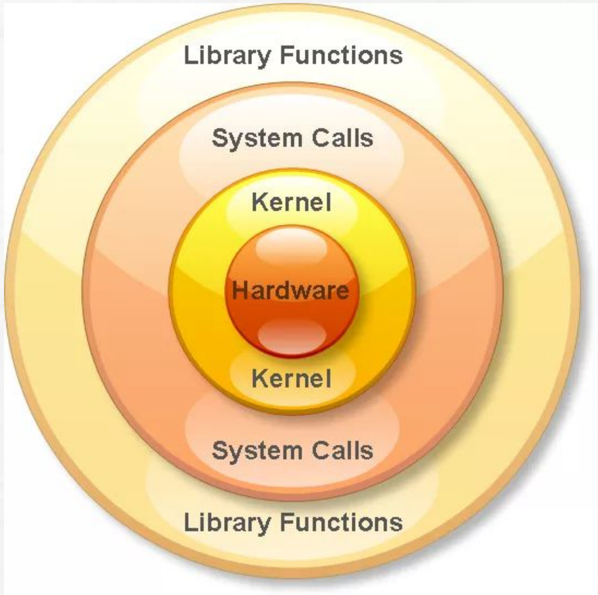
- Full virtualization
- Paravirtualization
- Hardware-assisted virtualization

Hypervisors

- Enables communication between hardware and a virtual machine.
- Hypervisor vs. VMM
- Type1 - sitting on hardware - also called bare metal hypervisor
- Type 2 - on top of the operating system

Introduction to virtualization

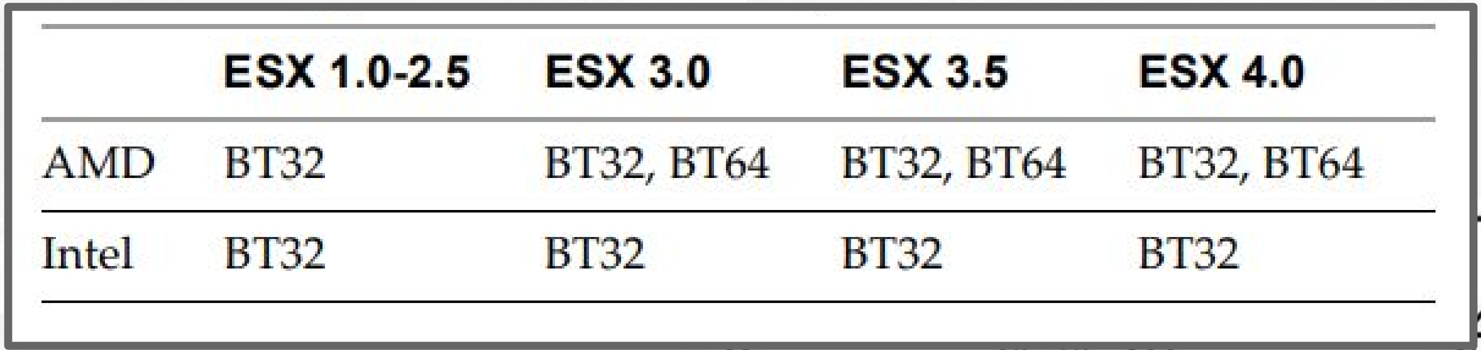
- Host machine vs. guest machine
- x86 privilege levels (protection rings)
- Segmentation - a hardware feature of the x86 CPU that limits access of memory.



Binary Translation

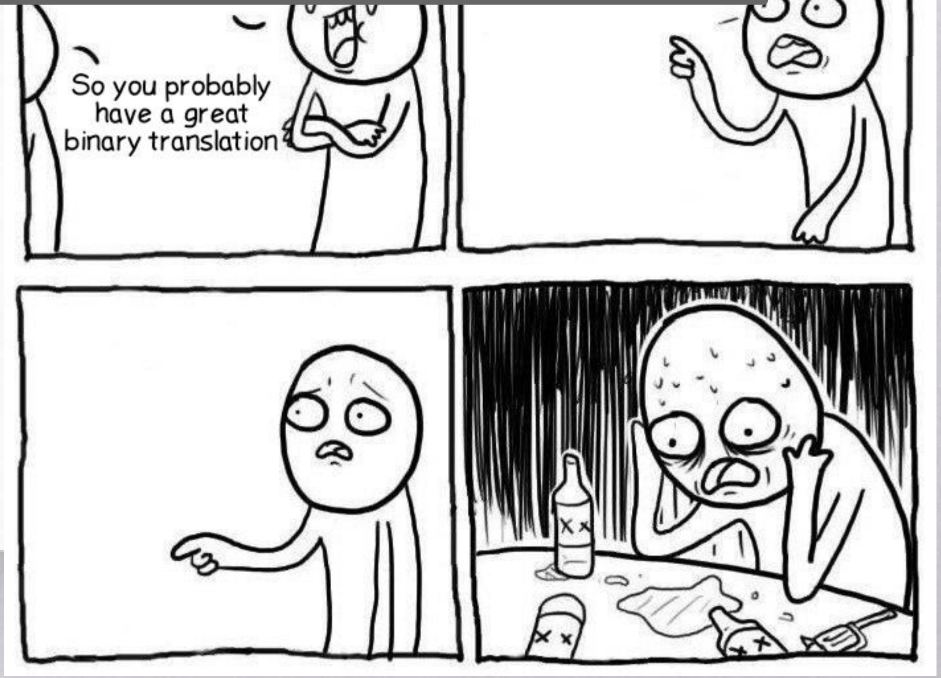
- Replaces privileged instructions with sequences of instructions that perform the privileged operations in the virtual machine rather than on the physical machine
- Often uses a translation cache
- Combined with direct execution of user mode code running in the virtual machine
- VMMs enforce usage of VM memory only using hardware segmentation

Full Virtualization



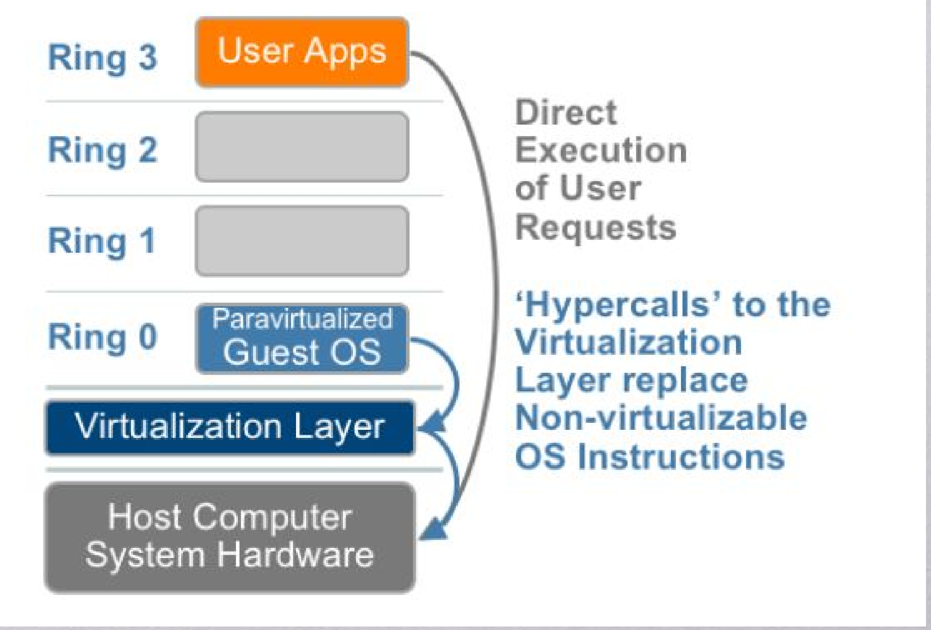
	ESX 1.0-2.5	ESX 3.0	ESX 3.5	ESX 4.0
AMD	BT32	BT32, BT64	BT32, BT64	BT32, BT64
Intel	BT32	BT32	BT32	BT32

- Uses binary translation
- Who?
  - Microsoft Virtual Server
  - VMware ESXi (VT-x and AMD-V)



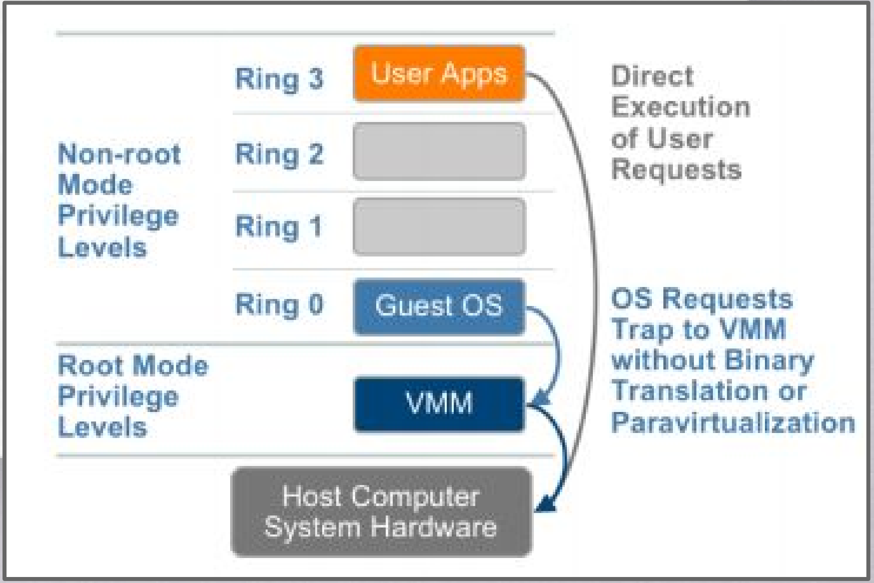
Paravirtualization

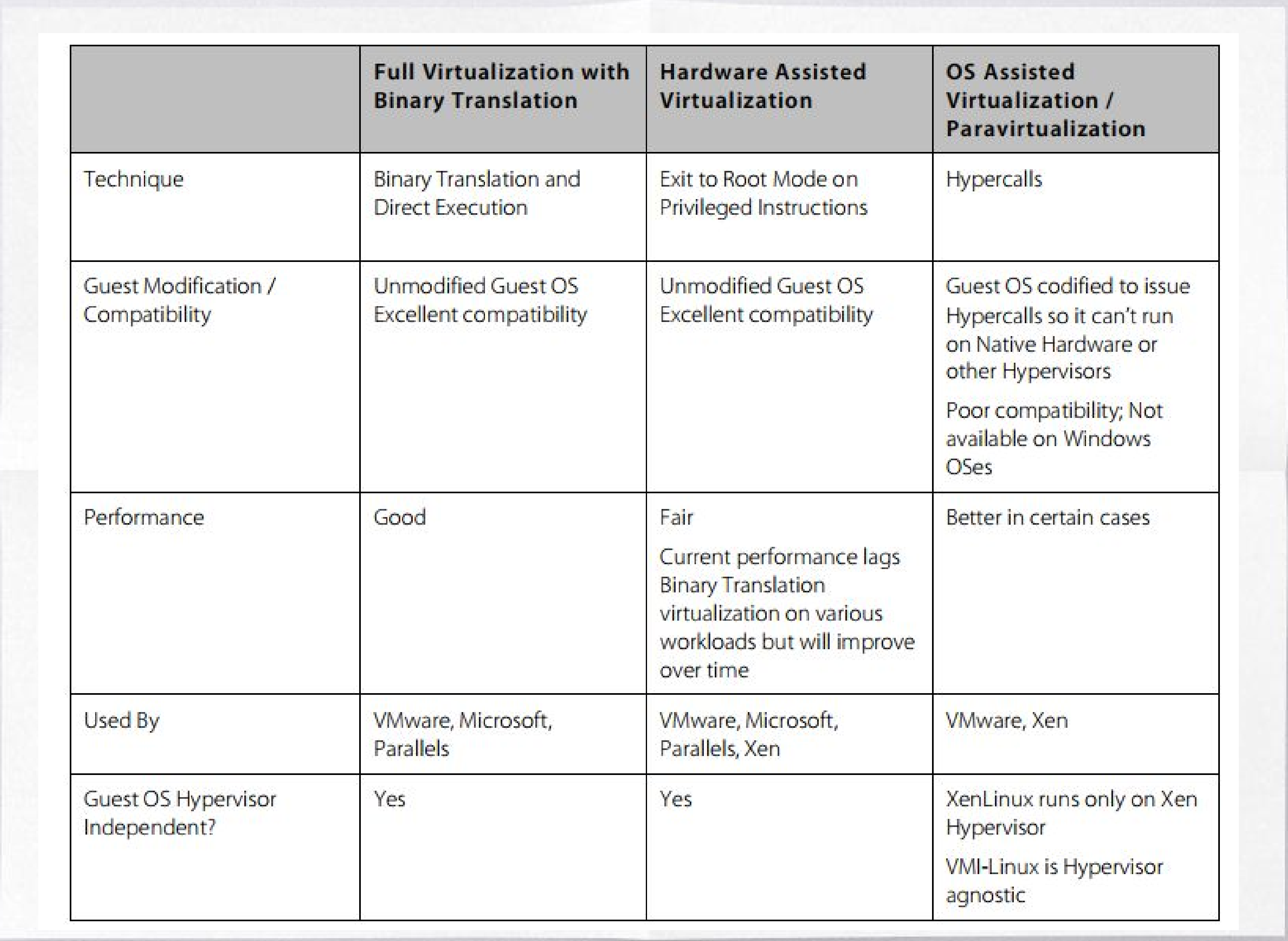
- Created to replace the big overhead of full virtualization
- Naturally, more suitable to OS's that run external VMMs
- paravirtualization cannot support unmodified operating systems


Ring 3
User Apps
Ring 2
Ring 1
Ring 0
Paravirtualized Guest OS
Virtualization Layer
Host Computer System Hardware
Direct Execution of User Requests
'Hypercalls' to the Virtualization Layer replace Non-virtualizable OS Instructions

Hardware Assisted Virtualization

- Privileged and sensitive calls are set to automatically trap the hypervisor
- The guest state is stored in Virtual Machine Control Structures (Intel’s VT-x) or Virtual Machine Control Blocks (AMD-V).
- VMware’s binary translation outperforms hardware assist implementations

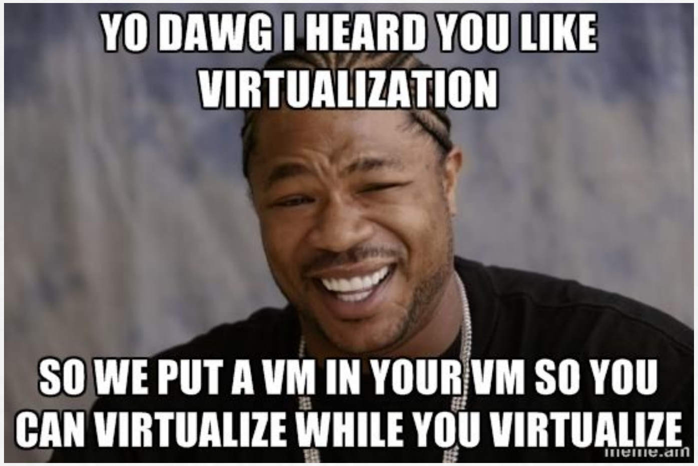

Ring 3
User Apps
Direct Execution of User Requests
Non-root Mode Privilege Levels
Ring 2
Ring 1
Ring 0
Guest OS
OS Requests Trap to VMM without Binary Translation or Paravirtualization
Root Mode Privilege Levels
VMM
Host Computer System Hardware



	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic

So, what are You doing?

- I’m working in the virtualization group of Ravello systems - now Oracle
- We povide a solution to whoever wants to virtualize their VMs and network over the cloud


YO DAWG I HEARD YOU LIKE VIRTUALIZATION
SO WE PUT A VM IN YOUR VM SO YOU CAN VIRTUALIZE WHILE YOU VIRTUALIZE
meme.am

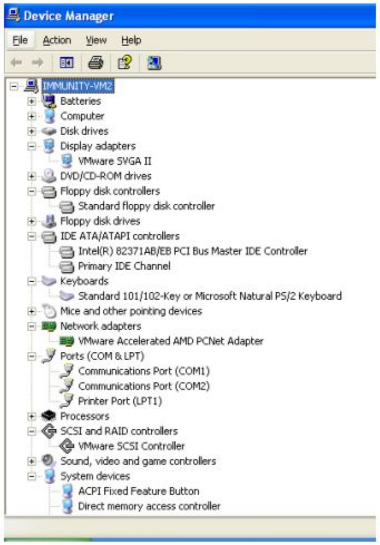
- @cbelle1234
- Carinebellef at gmail

Security issues?

- Using a virtual machine is a good thing – but don’t assume that only by using a virtual machine you’ll be completely protected
- (Anti analysis tricks)
- VM escaping - Breaking out of a virtual machine and interacting with the host operating system
- Pwn2Own vs. google
- Cloudbust - presented by Kostya Kortchinsky at Blackhat USA 2009

Cloudburst

- Most known vulnerabilities are related to shared-folders and I/O devices, using them to access the files and files-systems.



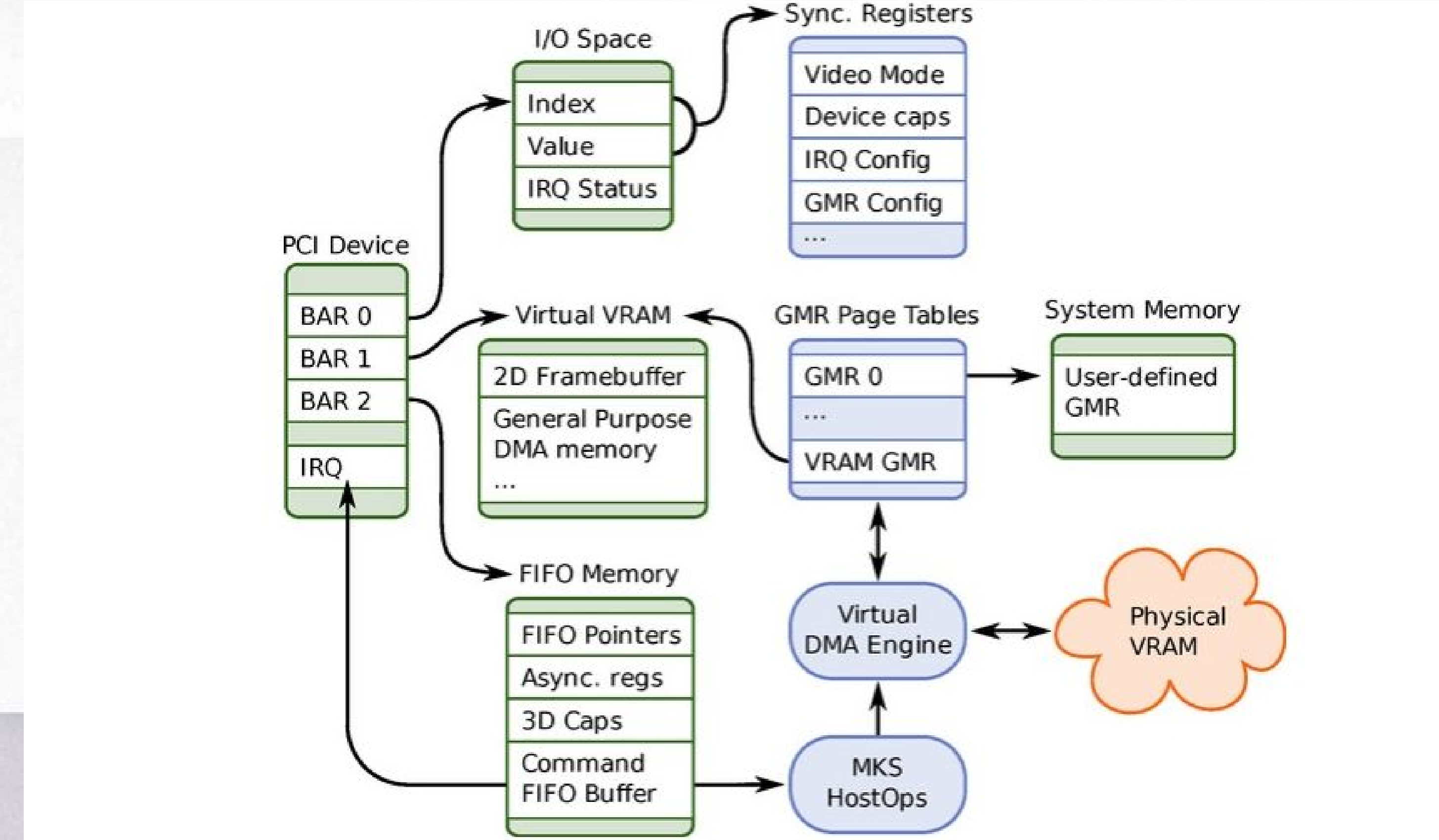
```
IMMUNITY-VM2
    Batteries
    Computer
    Disk drives
    Display adapters
        VMware SVGA II
    DVD/CD-ROM drives
    Floppy disk controllers
        Standard floppy disk controller
    Floppy disk drives
    IDE ATA/ATAPI controllers
        Intel(R) 82371AB/EB PCI Bus Master IDE Controller
        Primary IDE Channel
    Keyboards
        Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
    Mice and other pointing devices
    Network adapters
        VMware Accelerated AMD PCNet Adapter
    Ports (COM & LPT)
        Communications Port (COM1)
        Communications Port (COM2)
        Printer Port (LPT1)
    Processors
    SCSI and RAID controllers
        VMware SCSI Controller
    Sound, video and game controllers
    System devices
        ACPI Fixed Feature Button
        Direct memory access controller
```

Windows XP SP3 (ESX)

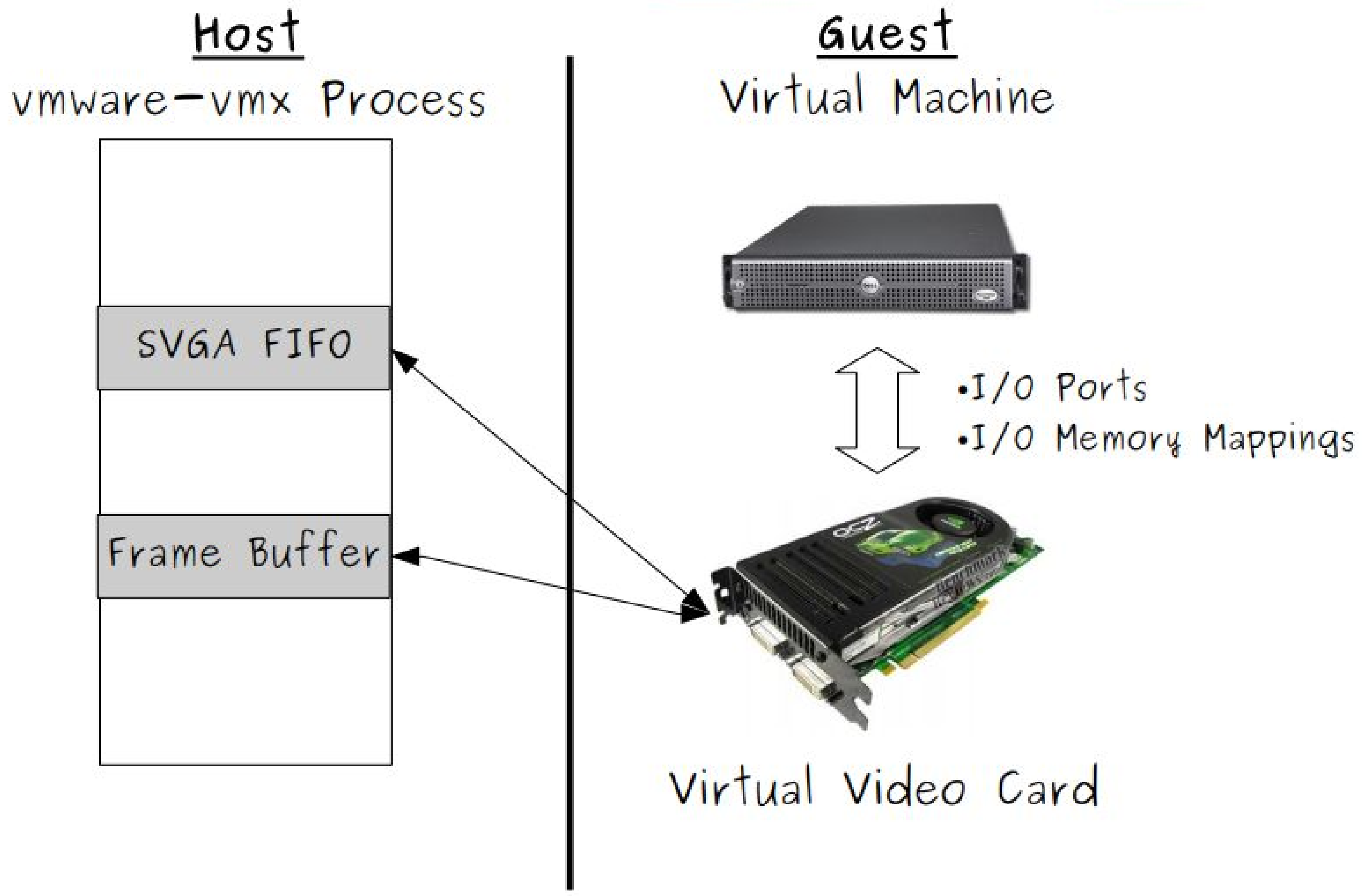
1. **Video adapter**
2. Floppy controller
3. IDE controller
4. Keyboard controller
5. Network Adapter
6. COM/LPT controller
7. SCSI controller(s)
8. DMA controller
9. ~~USB controller (WKS)~~
10. ~~Audio adapter (WKS)~~

Cloudburst

- 3 different ways that the pci device can communicate with the host process



Cloudburst



Host

vmware-vmx Process

SVGA FIFO

Frame Buffer

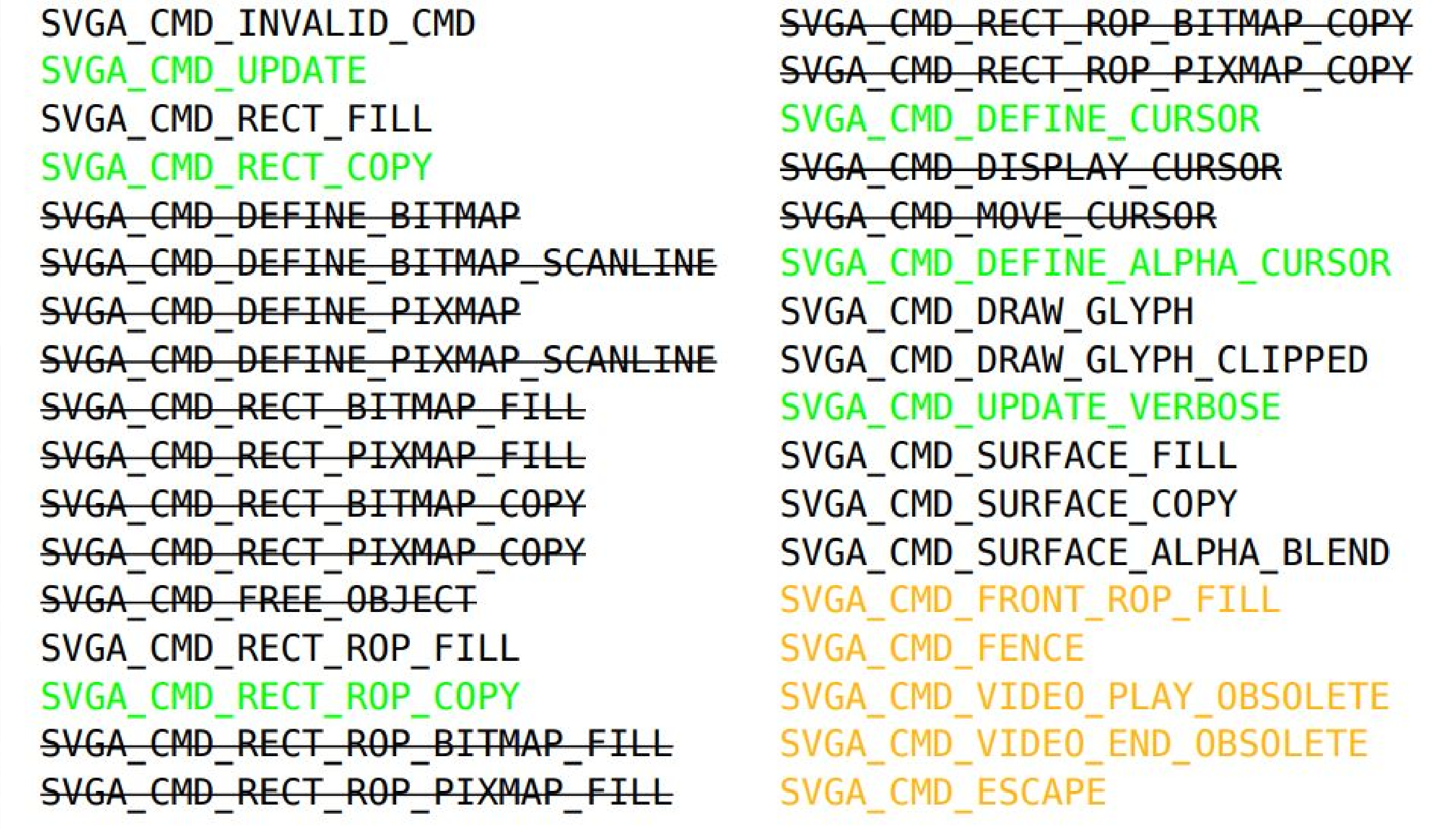
Guest

Virtual Machine

- I/O Ports
- I/O Memory Mappings

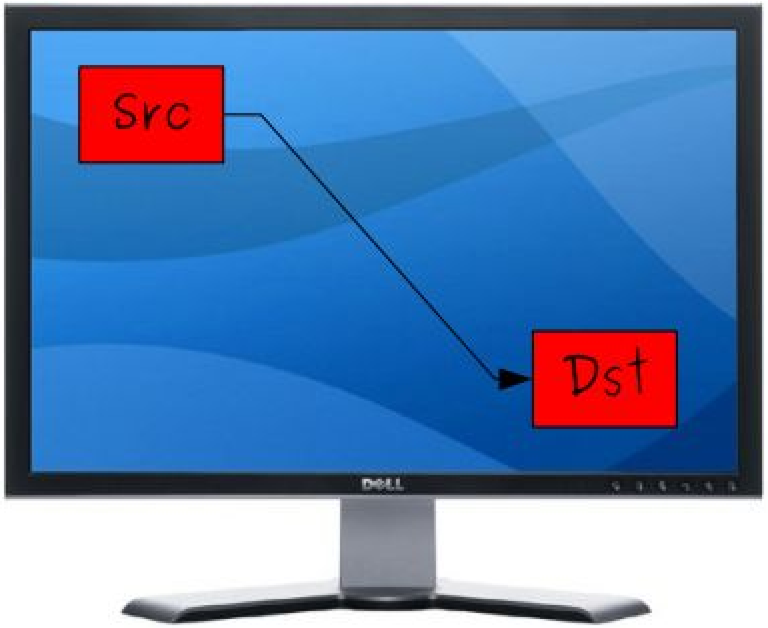
Virtual Video Card

Cloudburst



SVGA_CMD_INVALID_CMD
SVGA_CMD_UPDATE
SVGA_CMD_RECT_FILL
SVGA_CMD_RECT_COPY
~~SVGA_CMD_DEFINE_BITMAP~~
~~SVGA_CMD_DEFINE_BITMAP_SCANLINE~~
~~SVGA_CMD_DEFINE_PIXMAP~~
~~SVGA_CMD_DEFINE_PIXMAP_SCANLINE~~
~~SVGA_CMD_RECT_BITMAP_FILL~~
~~SVGA_CMD_RECT_PIXMAP_FILL~~
~~SVGA_CMD_RECT_BITMAP_COPY~~
~~SVGA_CMD_RECT_PIXMAP_COPY~~
~~SVGA_CMD_FREE_OBJECT~~
SVGA_CMD_RECT_ROP_FILL
SVGA_CMD_RECT_ROP_COPY
~~SVGA_CMD_RECT_ROP_BITMAP_FILL~~
~~SVGA_CMD_RECT_ROP_PIXMAP_FILL~~
~~SVGA_CMD_RECT_ROP_BITMAP_COPY~~
~~SVGA_CMD_RECT_ROP_PIXMAP_COPY~~
SVGA_CMD_DEFINE_CURSOR
~~SVGA_CMD_DISPLAY_CURSOR~~
~~SVGA_CMD_MOVE_CURSOR~~
SVGA_CMD_DEFINE_ALPHA_CURSOR
SVGA_CMD_DRAW_GLYPH
SVGA_CMD_DRAW_GLYPH_CLIPPED
SVGA_CMD_UPDATE_VERBOSE
SVGA_CMD_SURFACE_FILL
SVGA_CMD_SURFACE_COPY
SVGA_CMD_SURFACE_ALPHA_BLEND
SVGA_CMD_FRONT_ROP_FILL
SVGA_CMD_FENCE
SVGA_CMD_VIDEO_PLAY_OBSOLETE
SVGA_CMD_VIDEO_END_OBSOLETE
SVGA_CMD_ESCAPE

Cloudburst



Frame Buffer