VMware's Virtual Platform™

A Virtual Machine Monitor for Commodity PCs

Mendel Rosenblum

Chief Scientist & Founder VMware Inc.

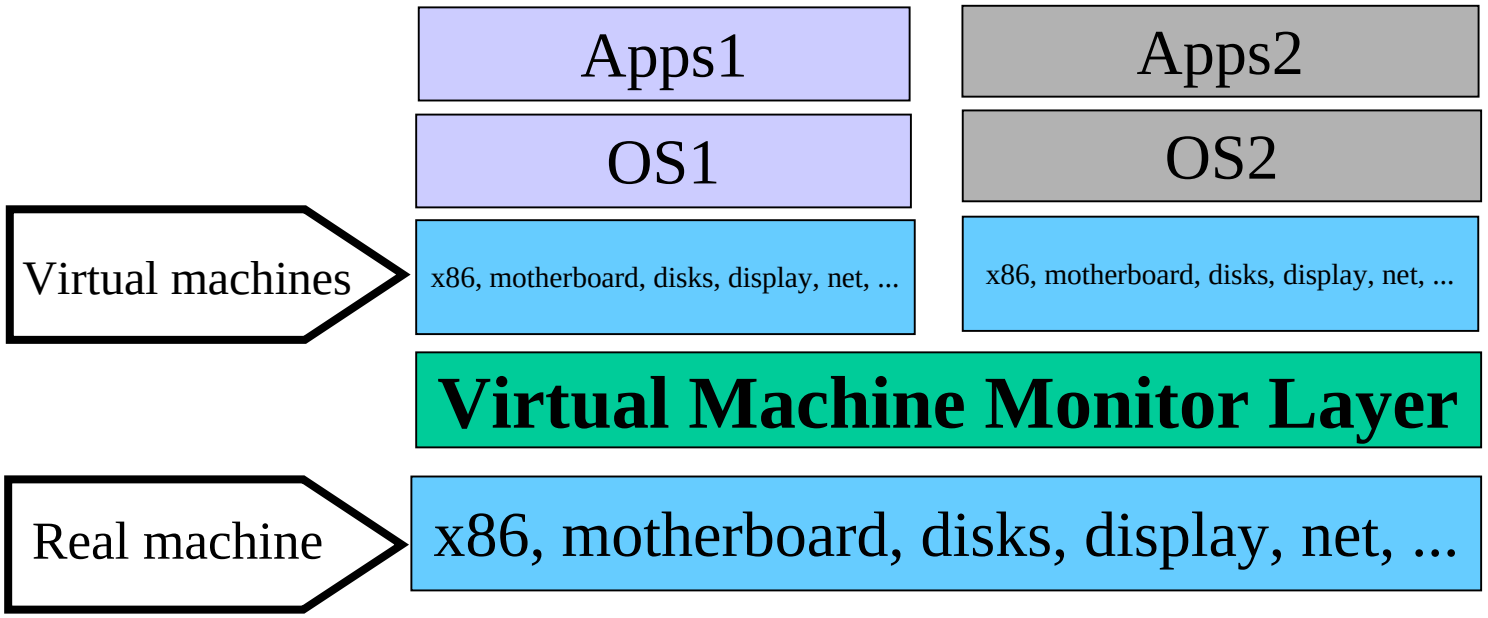
mendel@vmware.com

http://www.vmware.com

Talk Outline

- What is a virtual machine monitor (VMM)?
- Why would you want one on your PC?
- What are the problems of doing it for a PC?
- How do we solve these problems?
- What products are available?
- Conclusions

What's a virtual machine monitor



- VMMs popular during 1970s
  - Multiplex expensive hardware (e.g. IBM's VM)

Why on a Commodity x86 PC?

- Hardware now cheap, software expensive!
  - Multiplex expensive software on cheap HW
- Low-level general-purpose capability
  - Many different uses, problems being solved
    - Software development & testing
    - OS migration
    - Security
    - Many more

PC VMM Usage Examples

- SW development - develop, test, etc.

  Example: Run 95,98,NT,2000

- OS Migration - Perfect legacy app. support

  Example: Unix & Windows, Win98 and Win2000

- Security - Isolated environment

  Example: Fault and security isolation

- Other usage: Help desks, Multi-lingual, teaching, general freedom of choice, etc.

Challenges for PC VMMs

- Traditional VMM techniques won't work
  - x86 architecture not strictly virtualizable
- Large hardware diversity in PC marketplace
  - Want to run on **any** PC not just **a** PC
- Need ease of installation and use
  - Can't force user to reinstall all software, etc.

Traditional approach to VMMs

- Virtualize all resources
  - CPU, Memory, I/O devices
- Run all VM code non-privileged
  - Trap and emulate privileged operations
- Simulated virtual I/O devices by multiplexing access to real devices

x86 CPU problems

- Same opcode have different semantics in different protection rings
  - Example: POPF
- Privileged level visible to software
  - Example: MOV AX,CS or LMSW AX
- Some MMU “features” problematic
  - Cached segment descriptors; big real mode

Hardware Diversity Problem

- VMM must “understand” hardware
  - Which video card is in your PC?

    ATI, Matrox, Intel, Trident, S3, nVidia, Compaq, Dell, Diamond, Number Nine, Orchid, STB, ..
  - What SCSI card?

    BusLogic, Qlogic, Adaptec, NCR, UltraStor, ..
  - What LAN card?

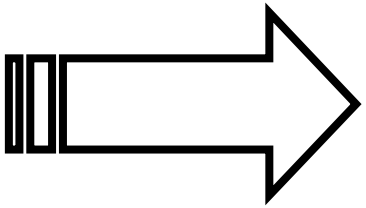
    3COM, Intel, Digital, AMD, SMD, National, IBM,…
- Problem: Too many drivers to write

Virtualizing the x86 architecture

- x86 is “somewhat” virtualizable
- Some CPU modes can be virtualized
  - If conditions are right, can use direct execution
    - Most user-level code and V8086 mode code
- Only a few instruction types are problematic
  - Examples: PUSHF/POPF, privileged insts
  - Need to interpose and emulate these

Privilege code patching

```
pushfd
cli
mov   eax,(0x824)
cmp   eax,1
je      5
mov   (0x900),edx
popfd
add     edx,eax
```



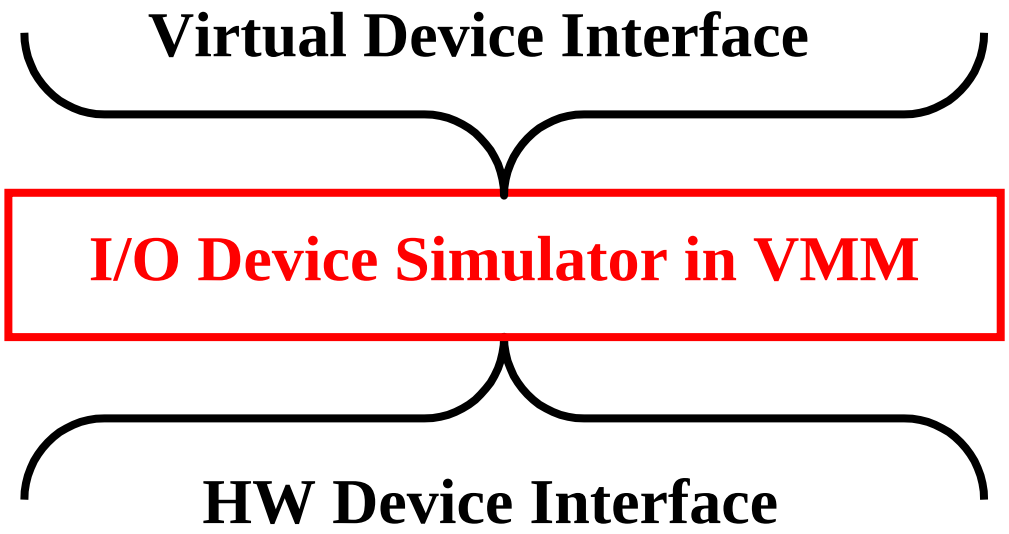
```
<pushfd sim insts>
<cli sim insts>
mov   eax,(0x824)
cmp   eax,1
je      5
mov   (0x900),edx
<popfd sim insts>
add     edx,eax
```

Challenges for patching x86 inst

- Heavy use of self-modifying code in the x86 software world
- Semantics of privileged instructions are some of the less well documented ones
- Need to hide and protect the patching code in the x86 linear address space

Virtual I/O Devices

- VMM must simulate virtual I/O devices

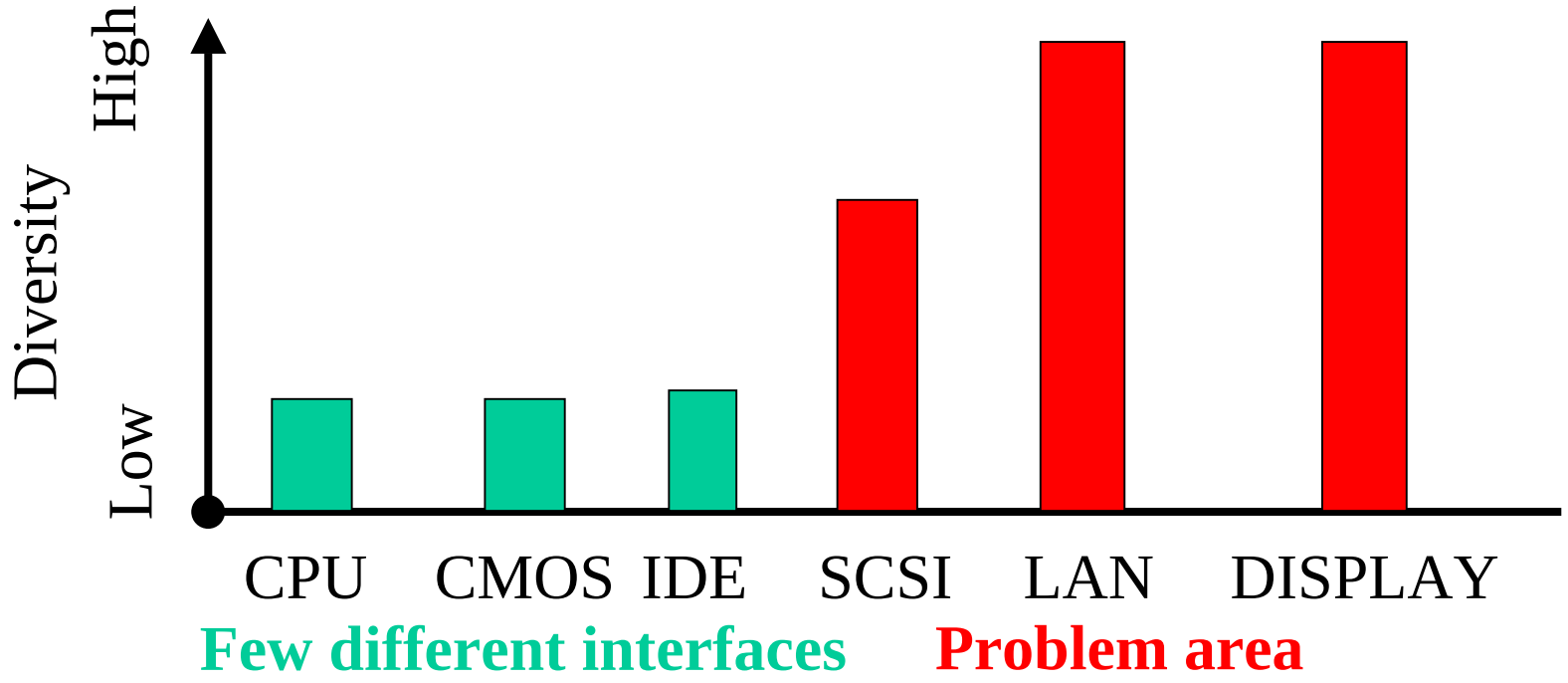


E.g. IDE

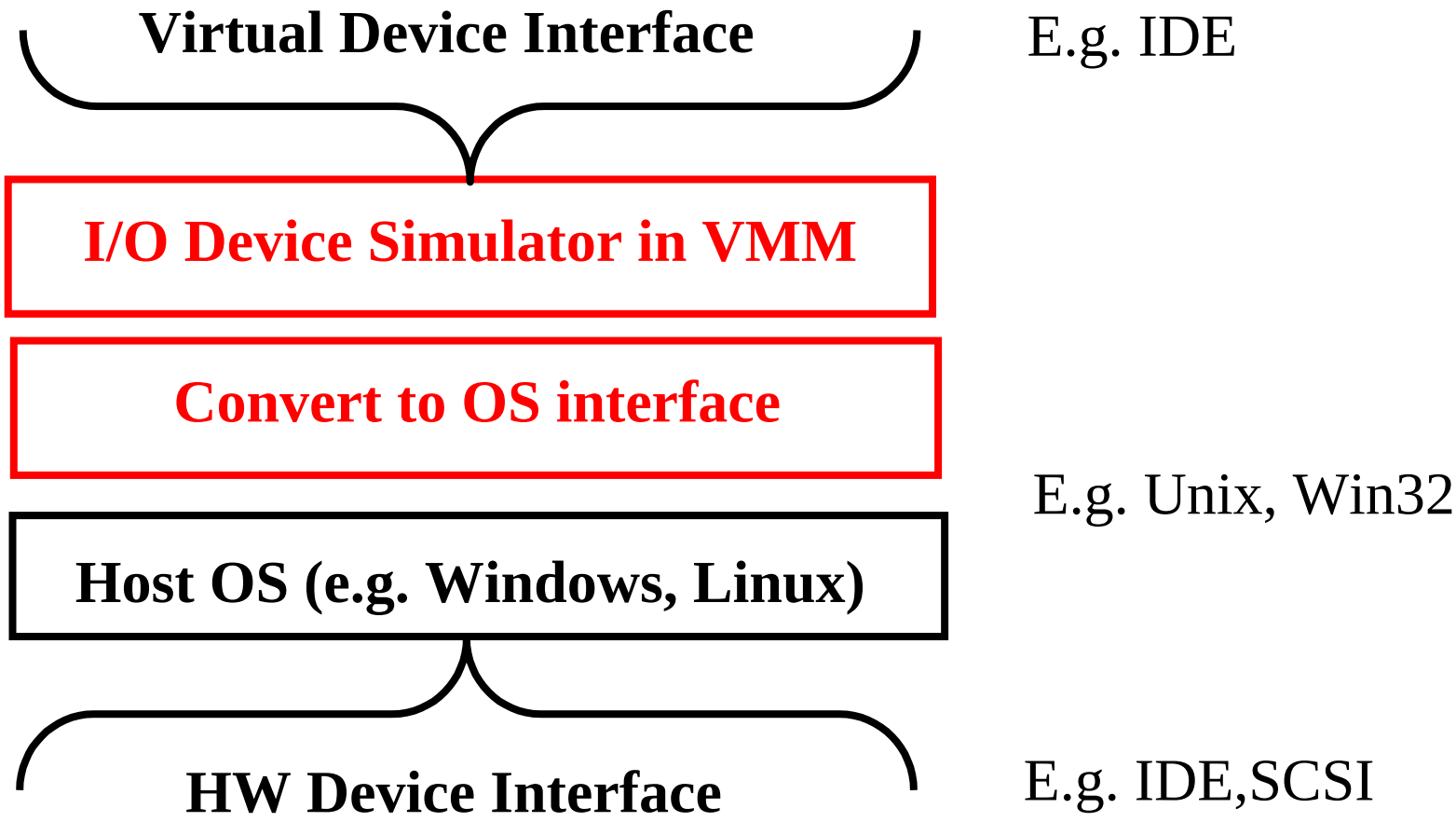
E.g. IDE,SCSI

Hardware Diversity Problem

- Some HW is “standard” others are not

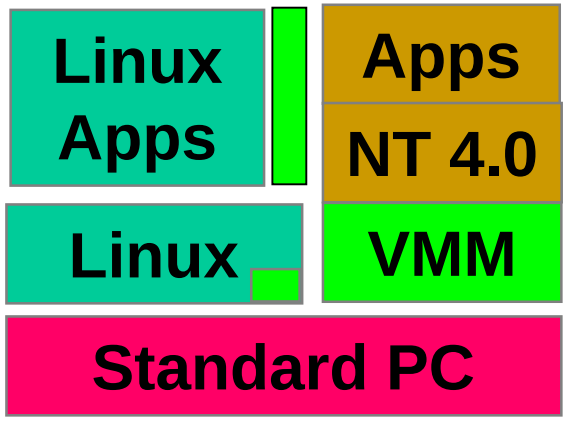


Our Approach to virtual devices

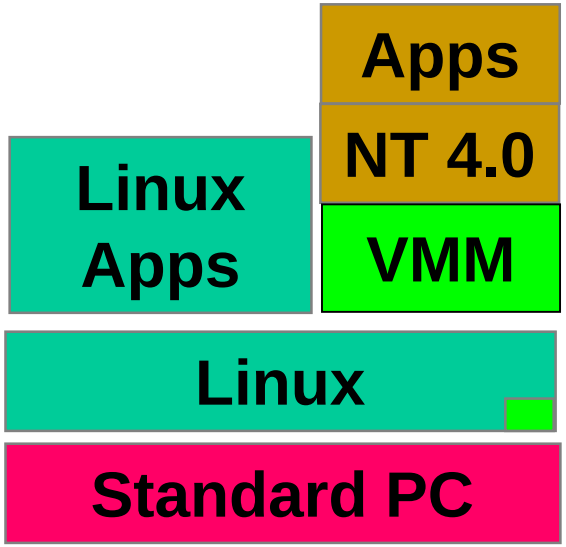


Dual-Mode Personality

Can masquerade as either a VMM or application



VMM - Direct to HW



App - Through Host OS

Dual-Mode Advantages

- Portability - Can run any OS on any HW
- VMM can use Host OS services
  - VMM easily installs like an app
  - Disk can be stored in file system, etc.
- Low impact on Host OS
  - Host OS runs at speed
  - Lowers the barriers to running multiple OSes

Other Resource Virtualization

- Virtual Physical Memory
  - Uses memory pages assigned by Host OS
  - Demand paged by the monitor
  - Active working set locked by driver
    - Driver acts like a device doing DMA
- Virtual I/O Devices
  - Some devices map to Host OS devices:

    Floppy, CDROM, sound, serial ports, parallel ports

Virtual Device Management

- Disks
  - Raw disk partition or file in Host OS
  - COW: Undoable or nonpersistent disks
- UI Devs: Video card, keyboard, mouse
  - X window or direct framebuffer access
  - With VMware tools, cut-and-paste.

Virtual Network Management

- Virtual Ethernet bridged to Host and VMs
- Can be used to share resources
  - File, printer access with Samba or NFS
- Assign an IP address to VM:
  - Access any remote service:

    Printers, file servers, etc.

Performance - Current Status

- CPU-bound workloads pretty good

Benchmark	Slowdown	Comment
CPUmark32	8%	All Direct
Norton SI32	30%	All Patched
SPECint95	< 10% Est.	All Direct
SPECfp95	< 10% Est.	All Direct
Intel Media	2x Est.	Direct + Graphics

- Graphics-intensive a problem area
  - Can't get at all of video card accelerations
  - Need to pass-through card to VM

Virtual Platform™ Products

- VMware for Linux™ version 1.0
  - Available from online store

    http://www.vmware.com
  - Introductory pricing

    Commercial: $199 Non-commercial: $75
- VMware for Windows NT™
  - In private beta now

Conclusion

- Virtual Platform™: VMM for a PC
  - Solves unique problems for PC VMMs:
    - x86 lack of virtualizability
    - Large I/O device diversity
    - Easy deployment
- Near future features for Virtual Platform™
  - Better Performance and Completeness
  - Checkpoint/restore

http://www.vmware.com