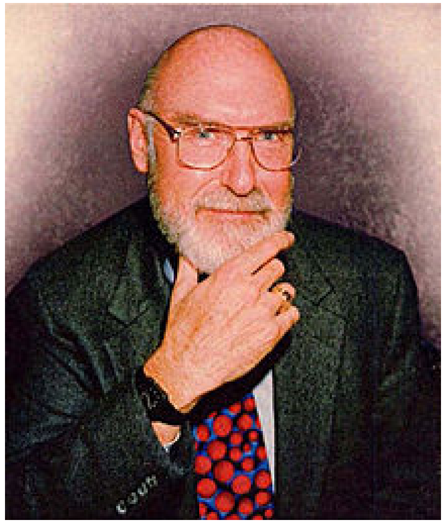
VIRTUALIZATION: IBM VM/370 AND XEN

CS6410

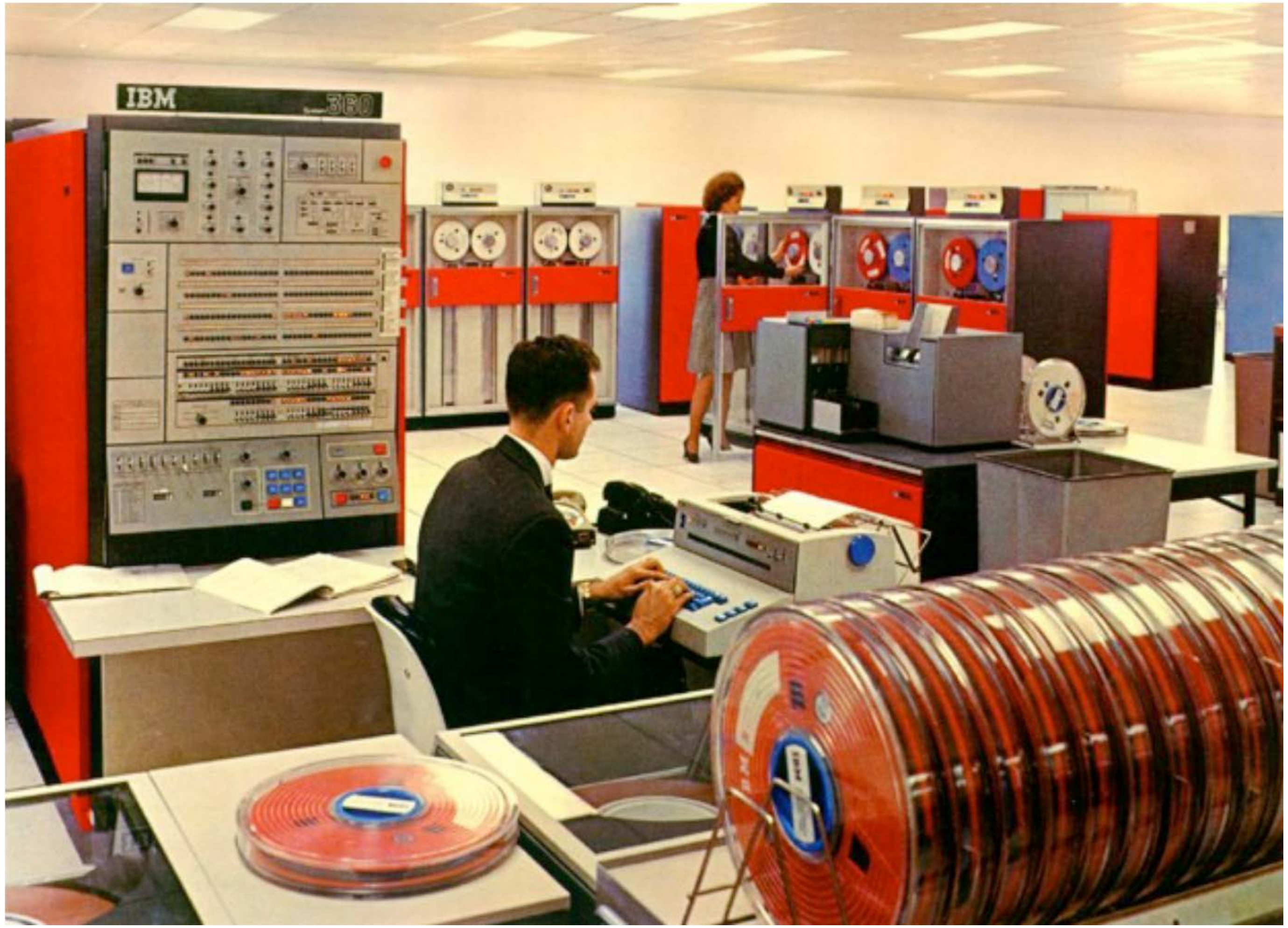
Hakim Weatherspoon

IBM VM/370

- Robert Jay Creasy (1939-2005)
  - Project leader of the first full virtualization hypervisor: IBM CP-40, a core component in the VM system
  - The first VM system: VM/370



Virtual Machine: Origin

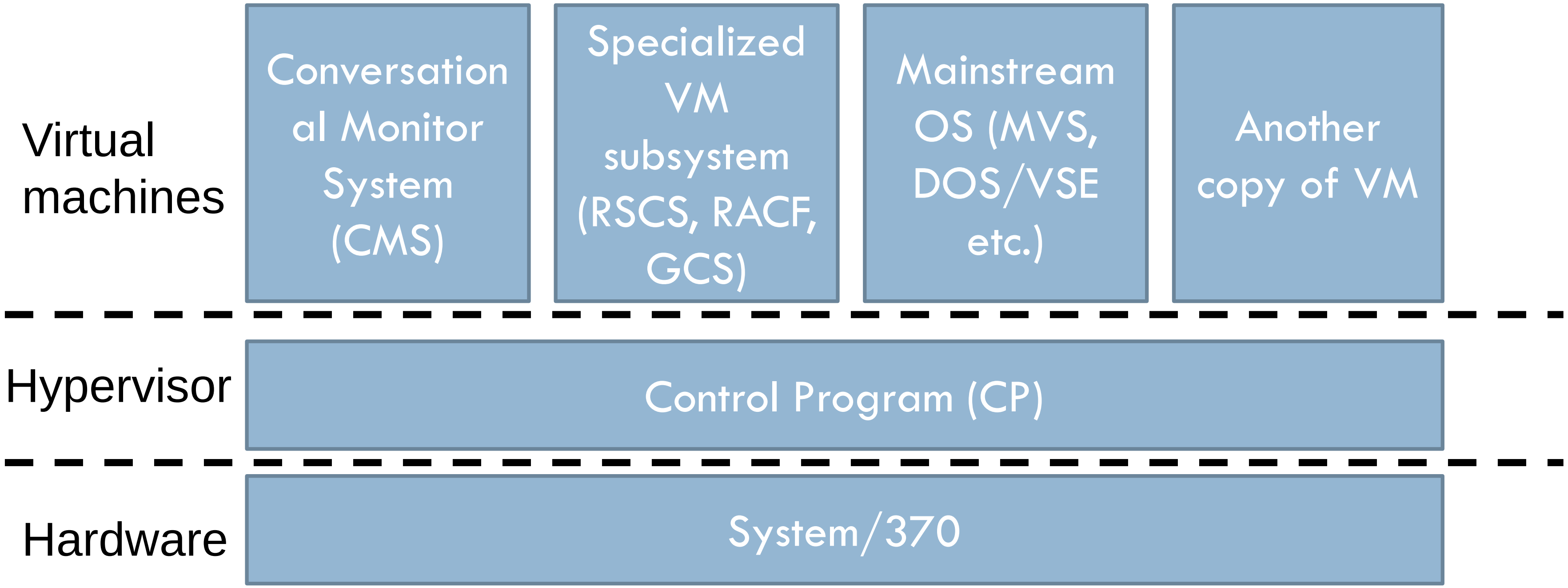


- IBM CP/CMS
  - CP-40
  - CP-67
  - VM/370

Why Virtualize

- Underutilized machines
- Easier to debug and monitor OS
- Portability
- Isolation
- The cloud (e.g. Amazon EC2, Google Compute Engine, Microsoft Azure)

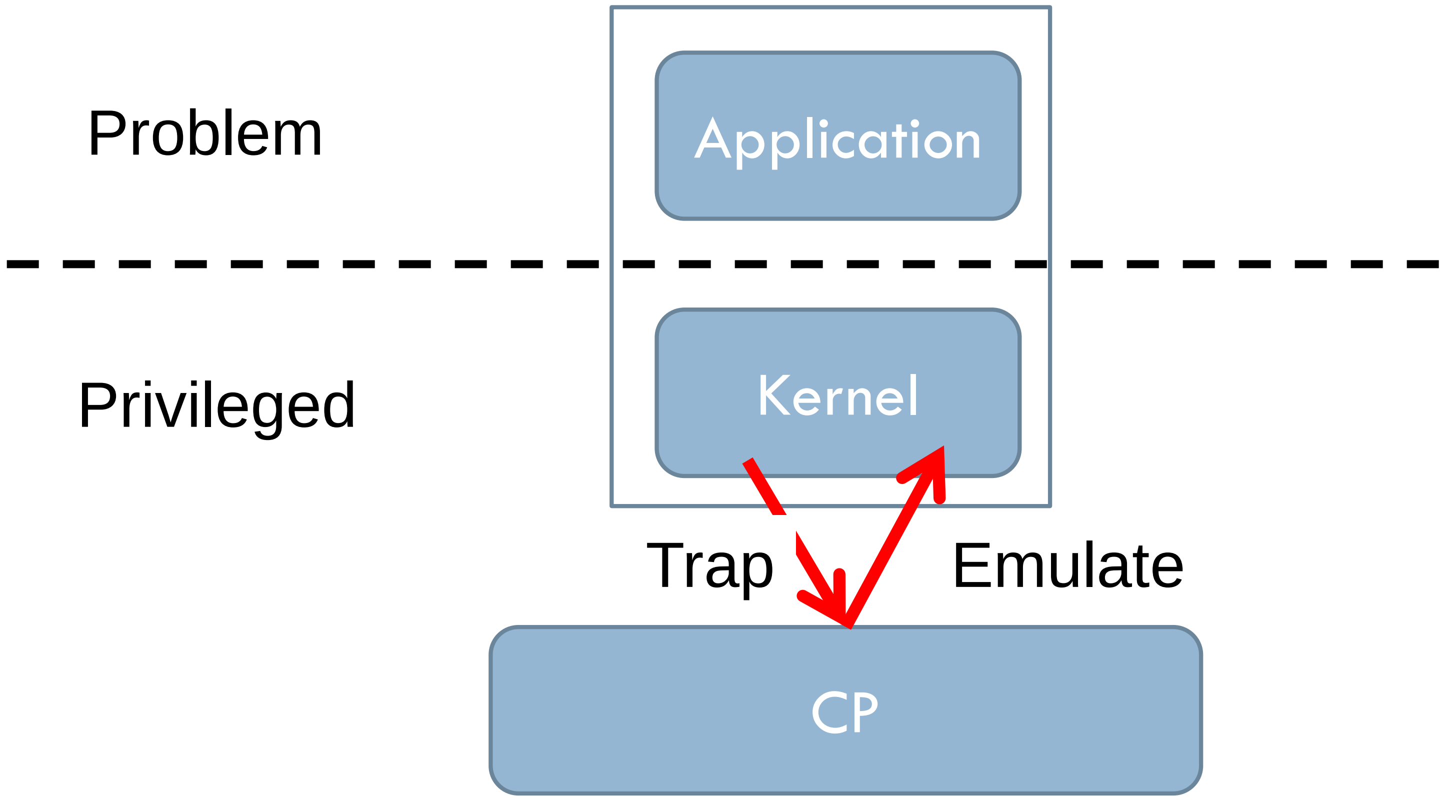
IBM VM/370



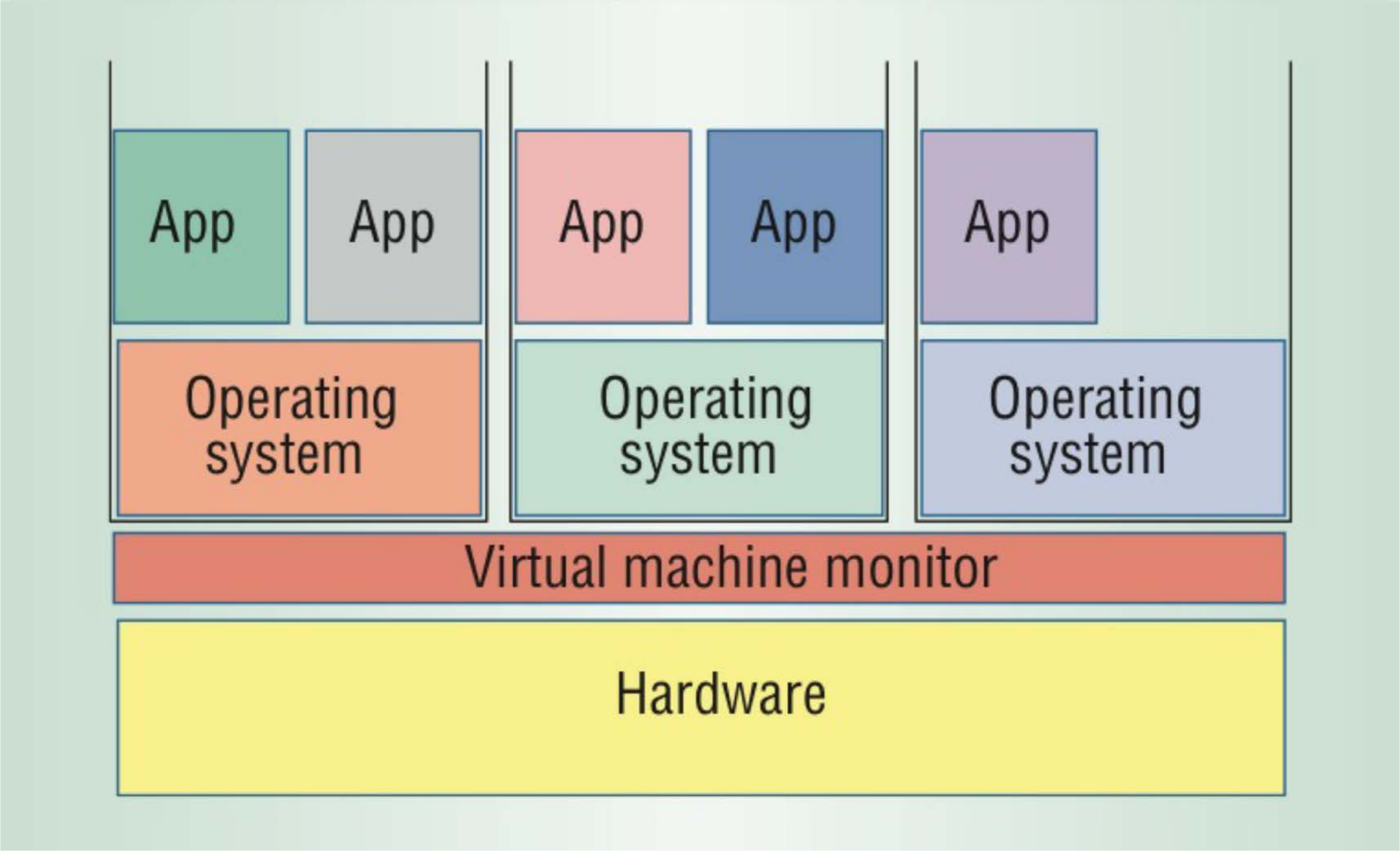
Virtual machines
Conversational Monitor System (CMS)
Specialized VM subsystem (RSCS, RACF, GCS)
Mainstream OS (MVS, DOS/VSE etc.)
Another copy of VM
Hypervisor
Control Program (CP)
Hardware
System/370

IBM VM/370

- Technology: trap-and-emulate



Classic Virtual Machine Monitor (VMM)



Virtualization: rejuvenation

- 1960's: first track of virtualization
  - Time and resource sharing on expensive mainframes
  - IBM VM/370
- Late 1970's and early 1980's: became unpopular
  - Cheap hardware and multiprocessing OS
- Late 1990's: became popular again
  - Wide variety of OS and hardware configurations
  - VMWare
- Since 2000: hot and important
  - Cloud computing
  - Docker containers

Full Virtualization

- Complete simulation of underlying hardware
- Unmodified guest OS
- Trap and simulate privileged instruction
- Was not supported by x86 (Not true anymore, Intel VT-x)
- Guest OS can't see real resources

Paravirtualization

- Similar but not identical to hardware
- Modifications to guest OS
- Hypercall
- Guest OS registers handlers
- Improved performance

VMware ESX Server

- Full virtualization
- Dynamically rewrite privileged instructions
- Ballooning
- Content-based page sharing

Denali

- Paravirtualization
- 1000s of VMs
- Security & performance isolation
- Did not support mainstream OSes
- VM uses single-user single address space

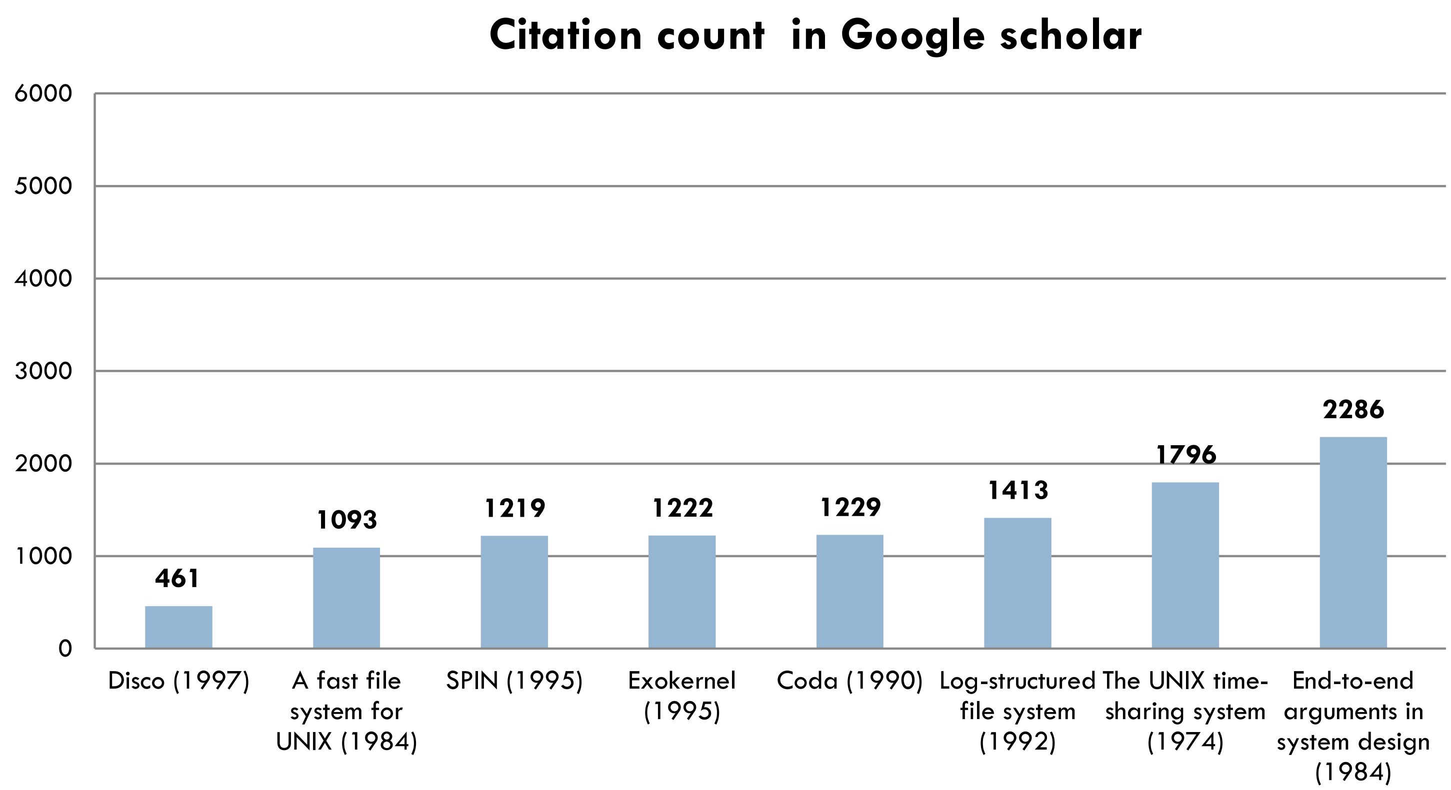
Xen and the Art of Virtualization

Xen

- University of Cambridge, MS Research Cambridge
- XenSource, Inc.
- Released in 2003 and published in SOSP 2003
- Acquired by Critix Systems in 2007 for $500M
- Now in RHEL5, Solaris, SUSE Linux Enterprise 10, EC2

Xen and the art of virtualization

- SOSP'03
- Very high impact (data collected in 2013)



Xen

- No changes to ABI (application binary interface)
- Full multi-application OS
- Paravirtualization
- Real and virtual resources
- Up to 100 VMs

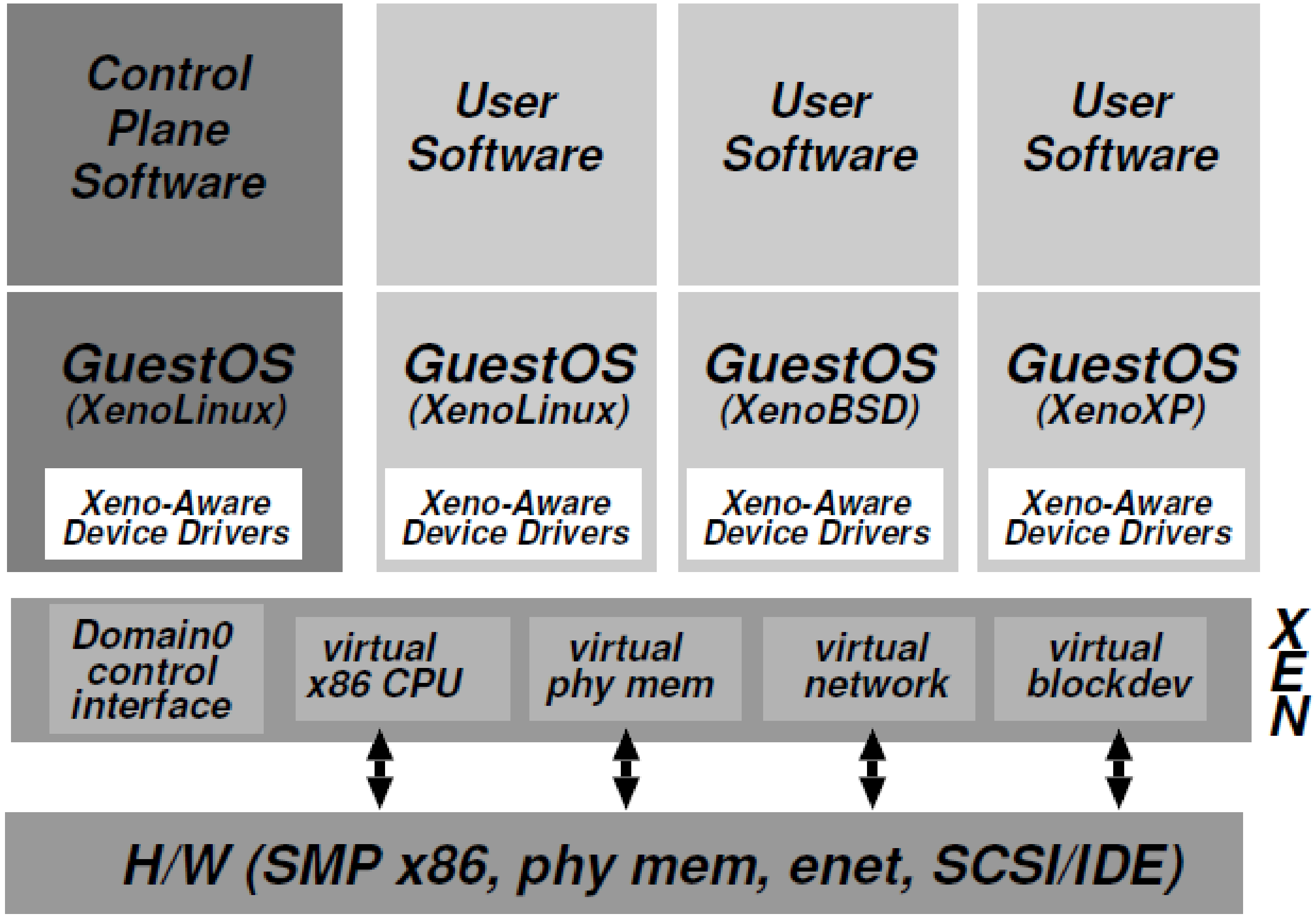
Virtualization on x86 architecture

- Challenges: Virtualization on x86 architecture
  - Correctness: not all privileged instructions produce traps!
    - Example: popf
  - Performance:
    - System calls: traps in both enter and exit (10X)
    - I/O performance: high CPU overhead
    - Virtual memory: no software-controlled TLB

Xen

- Xen 3.0 and up supports full virtualization with hardware support
- See backup slides

Xen architecture



Domain 0

- Management interface
- Created at boot time
- Policy from mechanism
- Privileged

Control Transfer

- Hypercalls
- Lightweight events

Interface: Memory Management

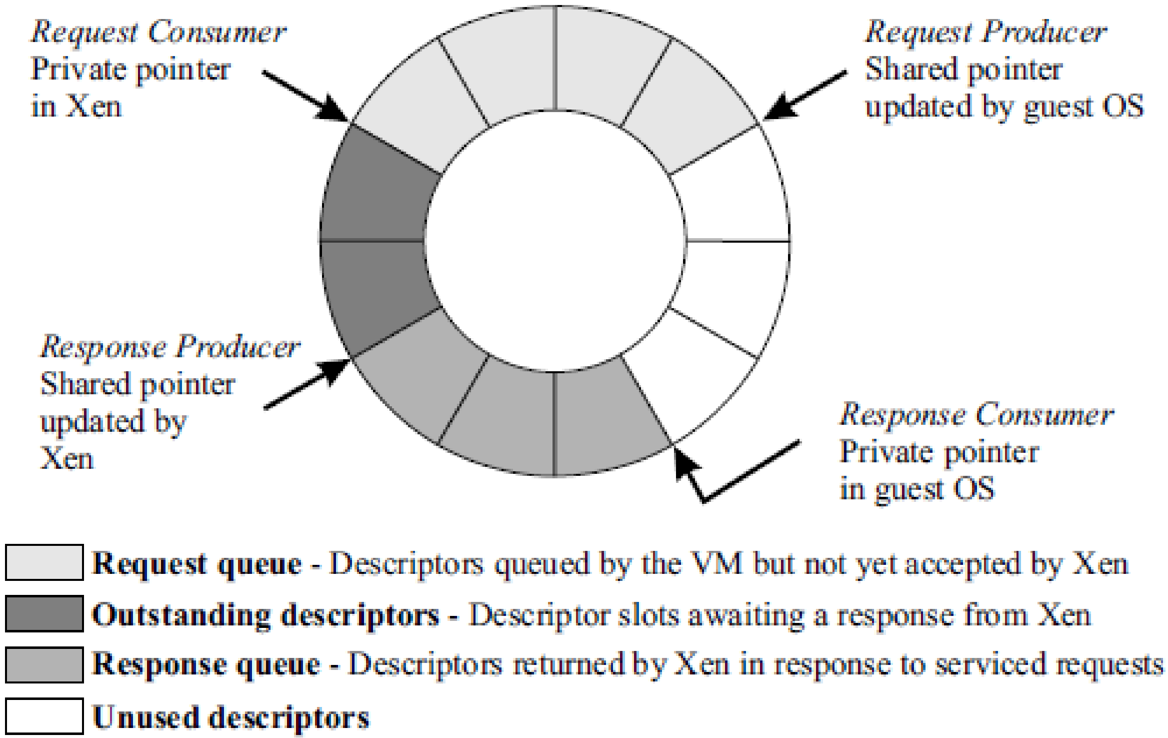
- Guest OSes manage their own page tables
- Register pages with Xen
- No direct write access
- Updates through Xen
- Hypervisor @ top 64MB of every address space
  - 2018: security issues with Meltdown/Spectre

Interface: CPU

- Xen in ring 0, OS in ring 1, everything else in ring 3
- “Fast” exception handler
- Xen handles page fault exceptions
- Double faulting

Interface: Device I/O

- Shared-memory, asynchronous buffer descriptor I/O rings



Subsystem Virtualization

- CPU Scheduling : Borrowed Virtual Time
- Real, virtual, and wall clock times
- Virtual address translation : updates through hyper call
- Physical memory : balloon driver, translation array
- Network : VFR, VIF
- Disk : VBD

Porting effort

OS subsection	# lines	
	Linux	XP
Architecture-independent	78	1299
Virtual network driver	484	–
Virtual block-device driver	1070	–
Xen-specific (non-driver)	1363	3321
Total	**2995**	**4620**
(Portion of total x86 code base	**1.36%**	**0.04%)**

Table 2: The simplicity of porting commodity OSes to Xen. The cost metric is the number of lines of reasonably commented and formatted code which are modified or added compared with the original x86 code base (excluding device drivers).

Evaluation: Relative Performance

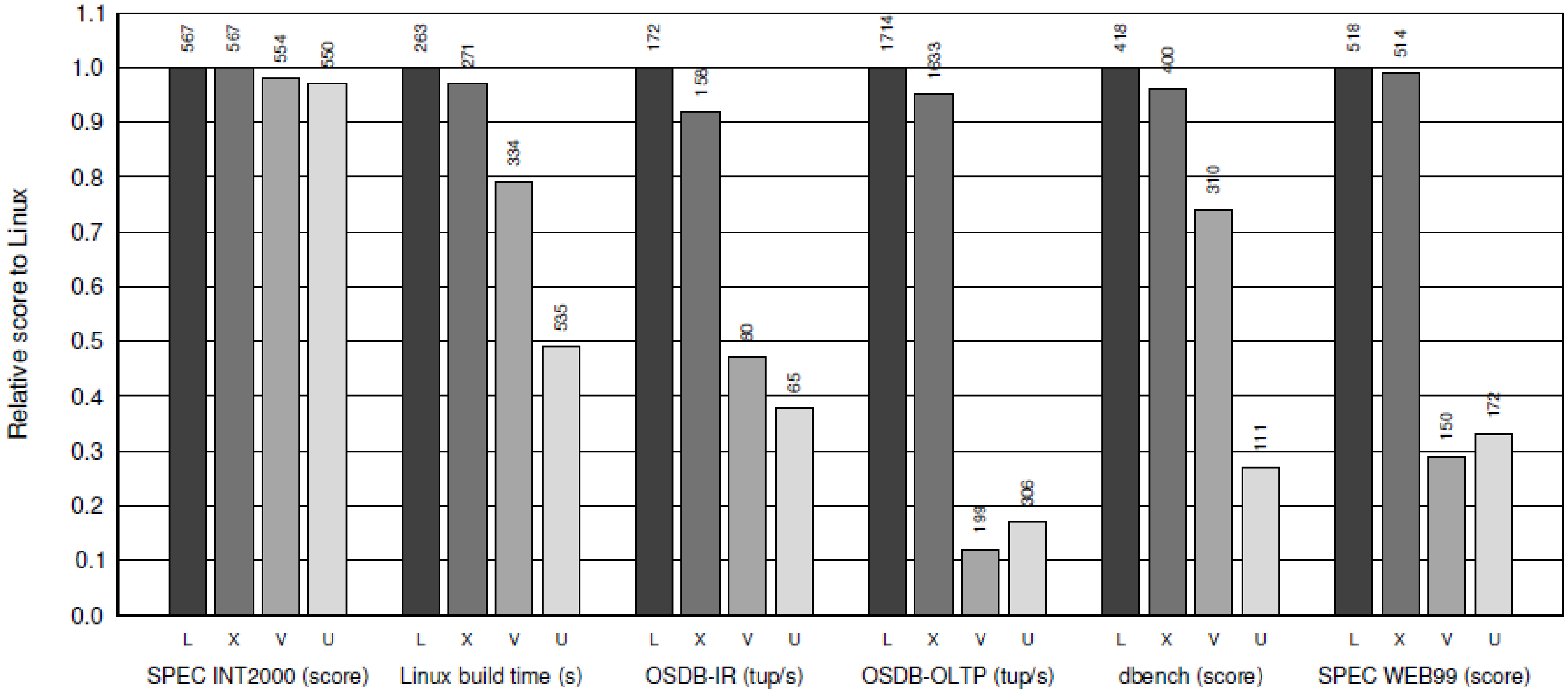


Figure 3: Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).

Evaluation: Concurrent Virtual Machines

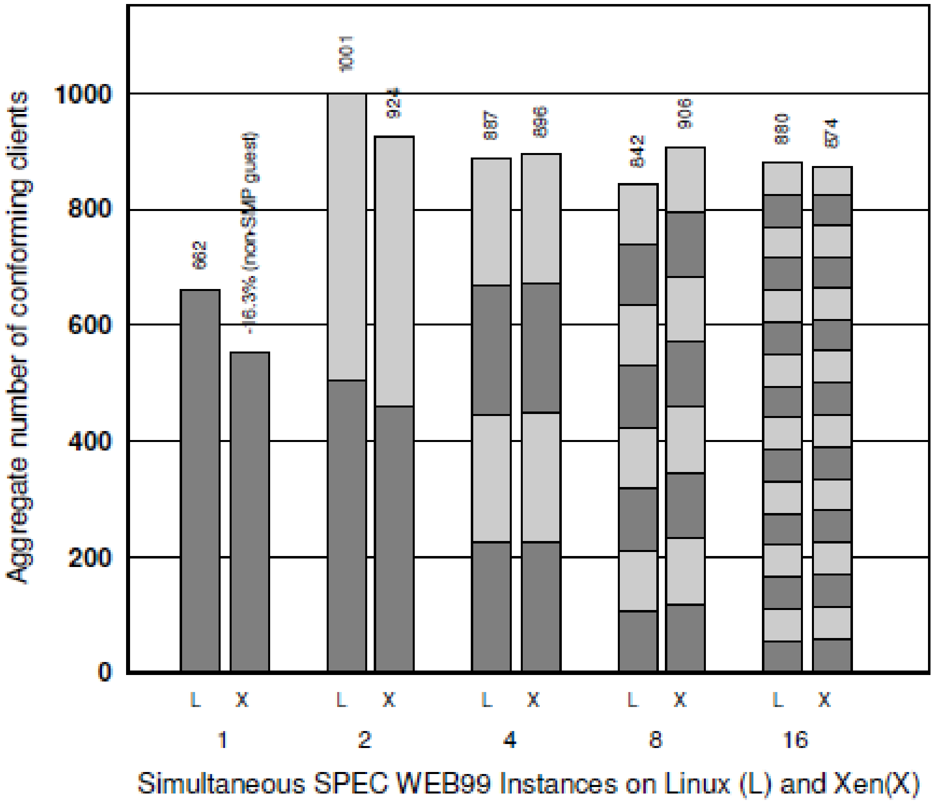


Figure 4: SPEC WEB99 for 1, 2, 4, 8 and 16 concurrent Apache servers: higher values are better.

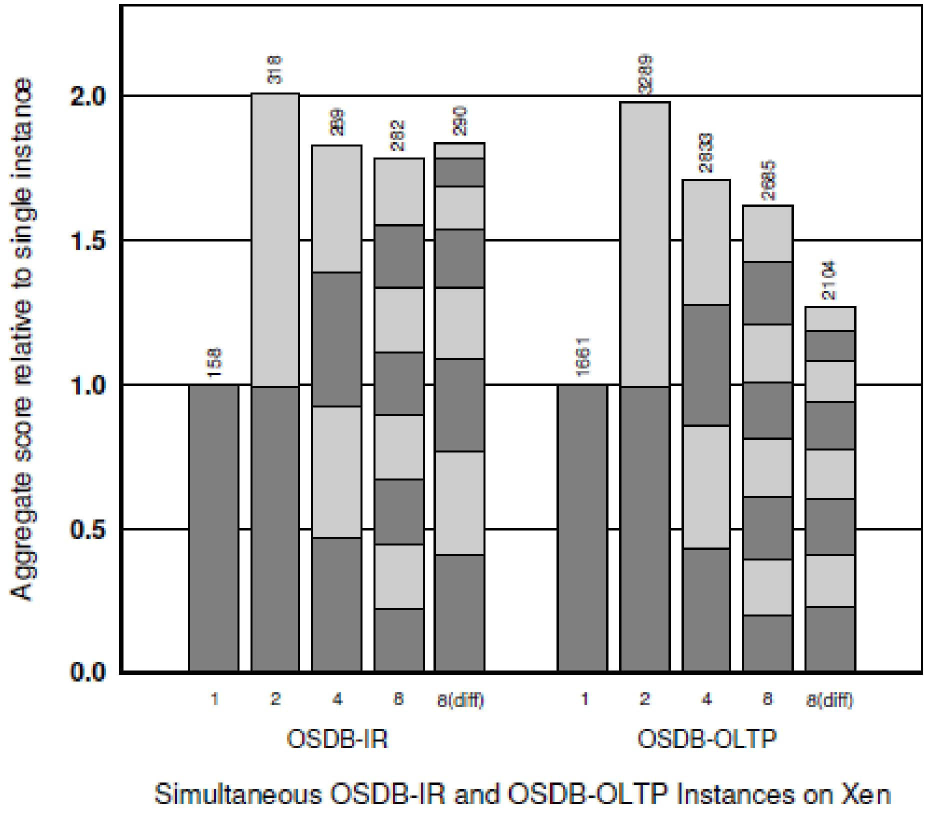
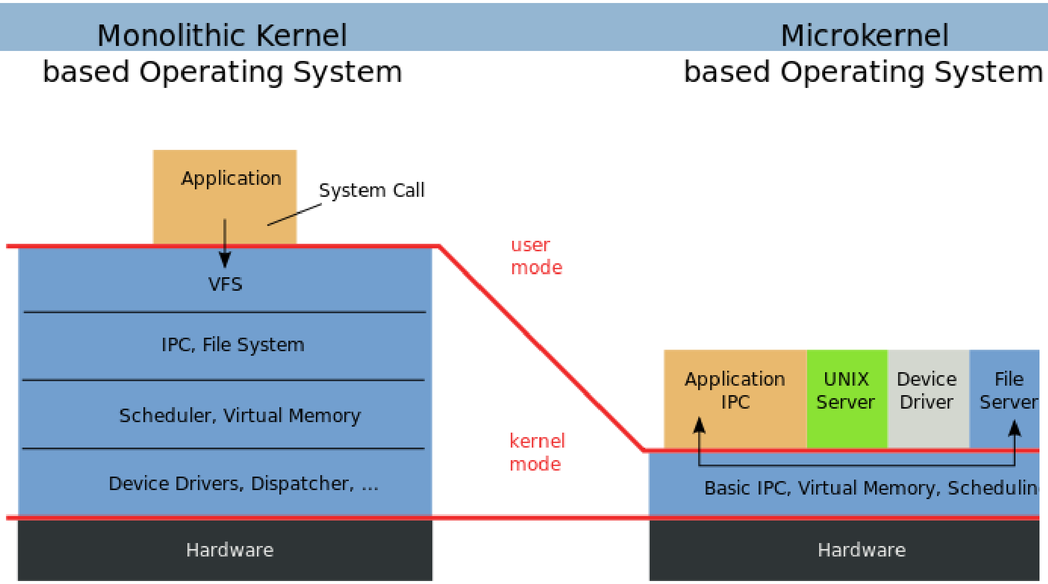


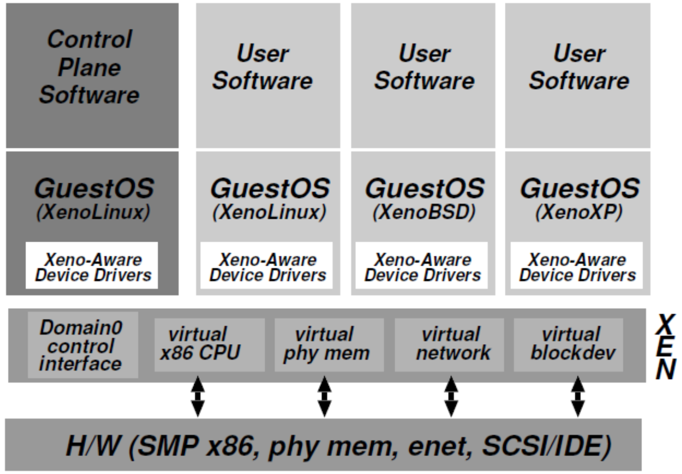
Figure 5: Performance of multiple instances of PostgreSQL running OSDB in separate Xen domains. 8(diff) bars show performance variation with different scheduler weights.

Conclusion

- x86 architecture makes virtualization challenging
- Full virtualization
  - unmodified guest OS; good isolation
  - Performance issue (especially I/O)
- Para virtualization:
  - Better performance (potentially)
  - Need to update guest kernel
- Full and para virtualization will keep evolving together

Microkernel vs. VMM(Xen)





Virtual Machine Monitor (VMM): *"… software which transforms the single machine interface into the illusion of many. Each of these interfaces (virtual machines) is an efficient replica of the original computer system, complete with all of the processor instructions …"*

-- Robert P. Goldberg. Survey of virtual machine research. 1974

Microkernel: *"... to minimize the kernel and to implement whatever possible outside of the kernel…"*

-- Jochen Liedtke. Towards real microkernels. 1996

Are Virtual Machine Monitors Microkernels Done Right?

Steven Hand, Andrew Wareld, Keir Fraser
HotOS'05

- VMMs (especially Xen) are microkernels done right
  - Avoid liability inversion:
    - Microkernels depend on some user level components
  - Make IPC performance irrelevant:
    - IPC performance is the key in microkernels
  - Treat the OS as a component
    - Hard for microkernels to support legacy applications

Are Virtual Machine Monitors Microkernels Done Right?

Gernot Heiser, Volkmar Uhlig, Joshua

A

- VMMs (especially Xen) are ... ne right. Really??
  - Avoid liability inversion:
    - Microkernels depend on some user level components
  - Make IPC performance irrelevant:
    - IPC performance is the key in microkernels
  - Treat the OS as a component
    - Hard for microkernels to support legacy applications

Xen also relies on Dom0!

Xen performs the same number of IPC!

Look at L4Linux!

Discussion

- What is the difference between VMMs and microkernels?
- Why do VMMs seem to be more successful than microkernels?

Perspective

- Virtualization: creating a illusion of something
- Virtualization is a principle approach in system design
  - OS is virtualizing CPU, memory, I/O …
  - VMM is virtualizing the whole architecture
  - What else? What next?

Next Time

- Project: next step is the Survey Paper due next Friday

- MP1 Milestone #1 due Today
- MP1 Milestone #2 due in two weeks

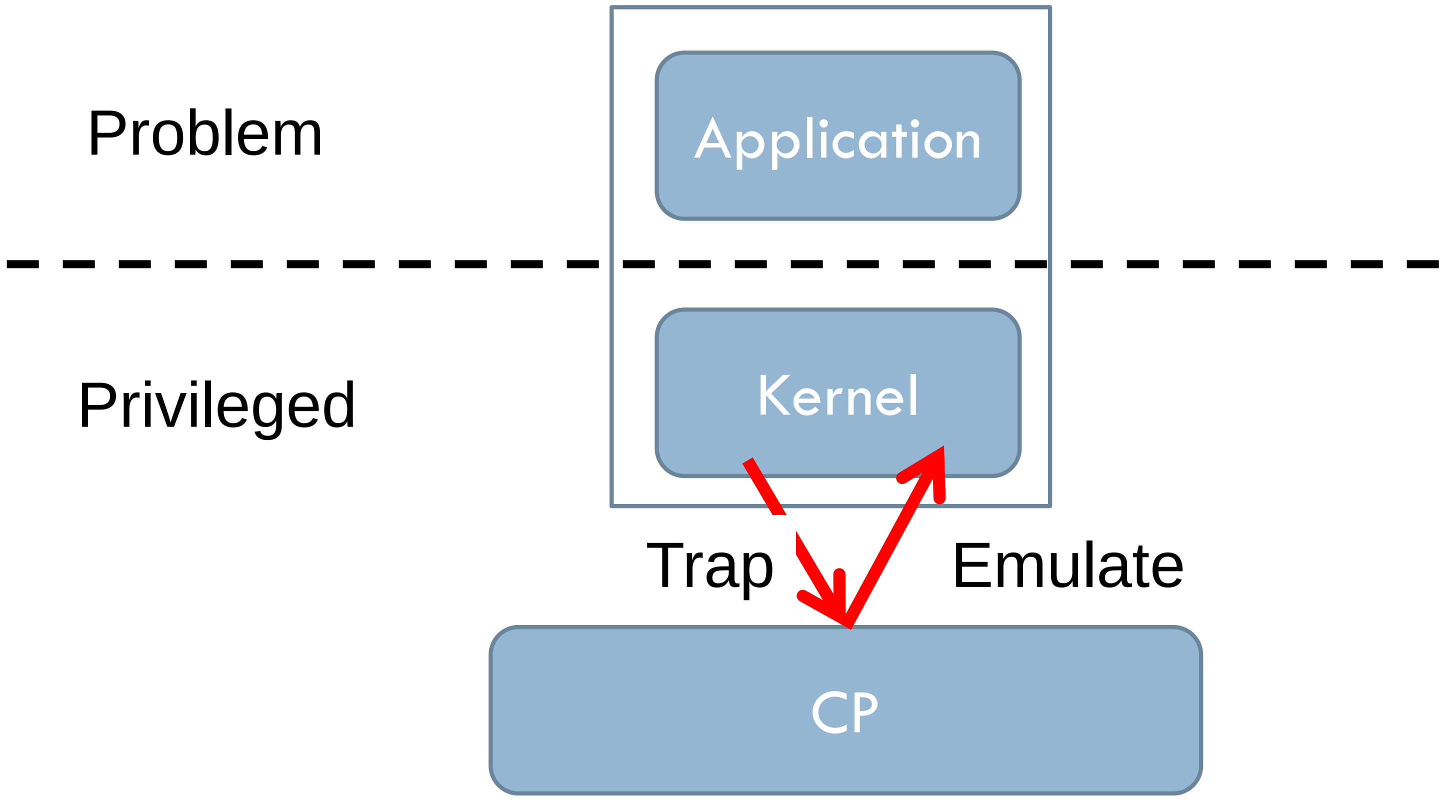
- Read and write a review:
  - ***Required*: Disco: Running Commodity Operating Systems on Scalable Multiprocessors,** Edouard Bugnion, Scott Devine, and Mendel Rosenblum. 16th ACM symposium on Operating systems principles (SOSP), October 1997, pages 143--156..

  - ***Optional*: The Multikernel: A new OS architecture for scalable multicore systems.** Andrew Baumann, Paul Barham, Pierre-Evariste Dagand, Tim Harrisy, Rebecca Isaacs, Simon Peter , Tim Roscoe, Adrian Sch�pbach, and Akhilesh Singhania . Proceedings of the Twenty-Second ACM Symposium on Operating Systems Principles (Austin, Texas, United States), ACM, 2009.

Backup

IBM VM/370

- Technology: trap-and-emulate



Virtualization on x86 architecture

- Challenges
  - Correctness: not all privileged instructions produce traps!
    - Example: popf
  - Performance:
    - System calls: traps in both enter and exit (10X)
    - I/O performance: high CPU overhead
    - Virtual memory: no software-controlled TLB

Virtualization on x86 architecture

- Solutions:
  - Dynamic binary translation & shadow page table
  - Hardware extension
  - Para-virtualization (Xen)

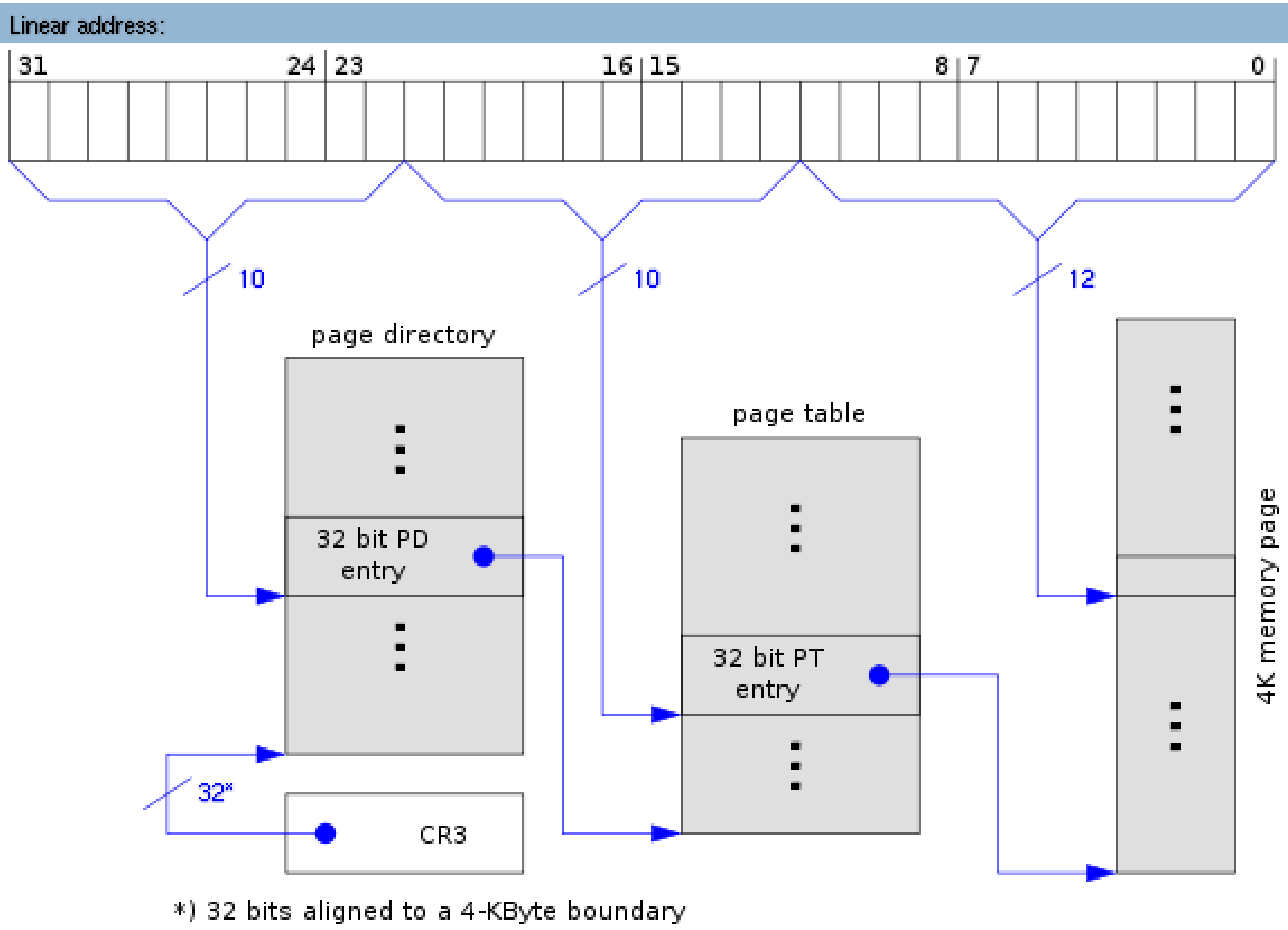
Dynamic binary translation

- Idea: intercept privileged instructions by changing the binary
- Cannot patch the guest kernel directly (would be visible to guests)
- Solution: make a copy, change it, and execute it from there
  - Use a cache to improve the performance

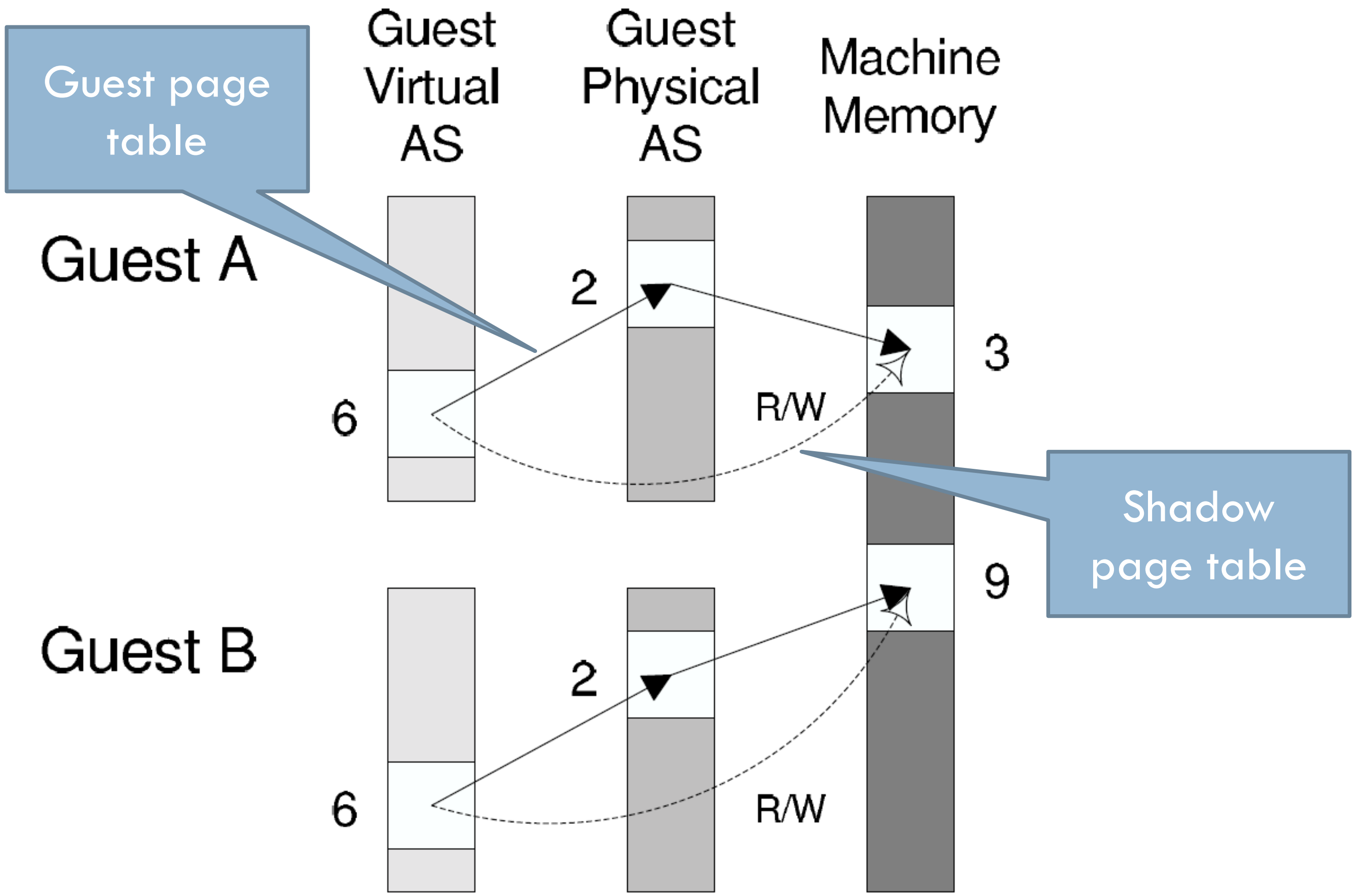
Dynamic binary translation

- Pros:
  - Make x86 virtualizable
  - Can reduce traps
- Cons:
  - Overhead
  - Hard to improve system calls, I/O operations
  - Hard to handle complex code

Shadow page table



Shadow page table



Shadow page table

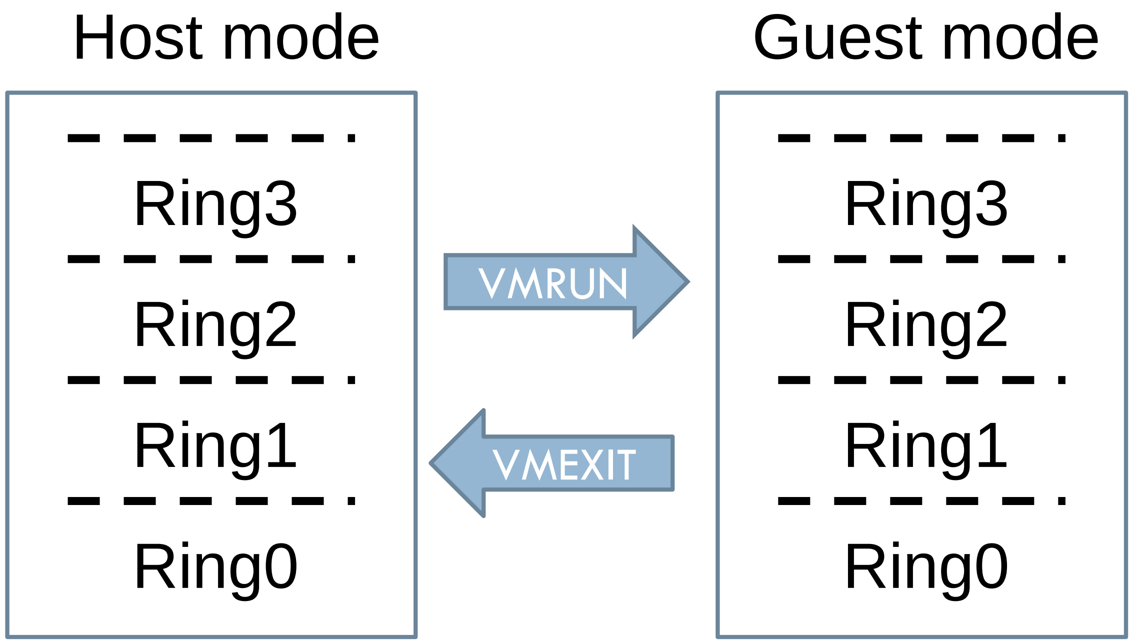
- Pros:
  - Transparent to guest VMs
  - Good performance when working set is stable
- Cons:
  - Big overhead of keeping two page tables consistent
  - Introducing more issues: hidden fault, double paging …

Hardware support

- First generation - processor
- Second generation - memory
- Third generation – I/O device

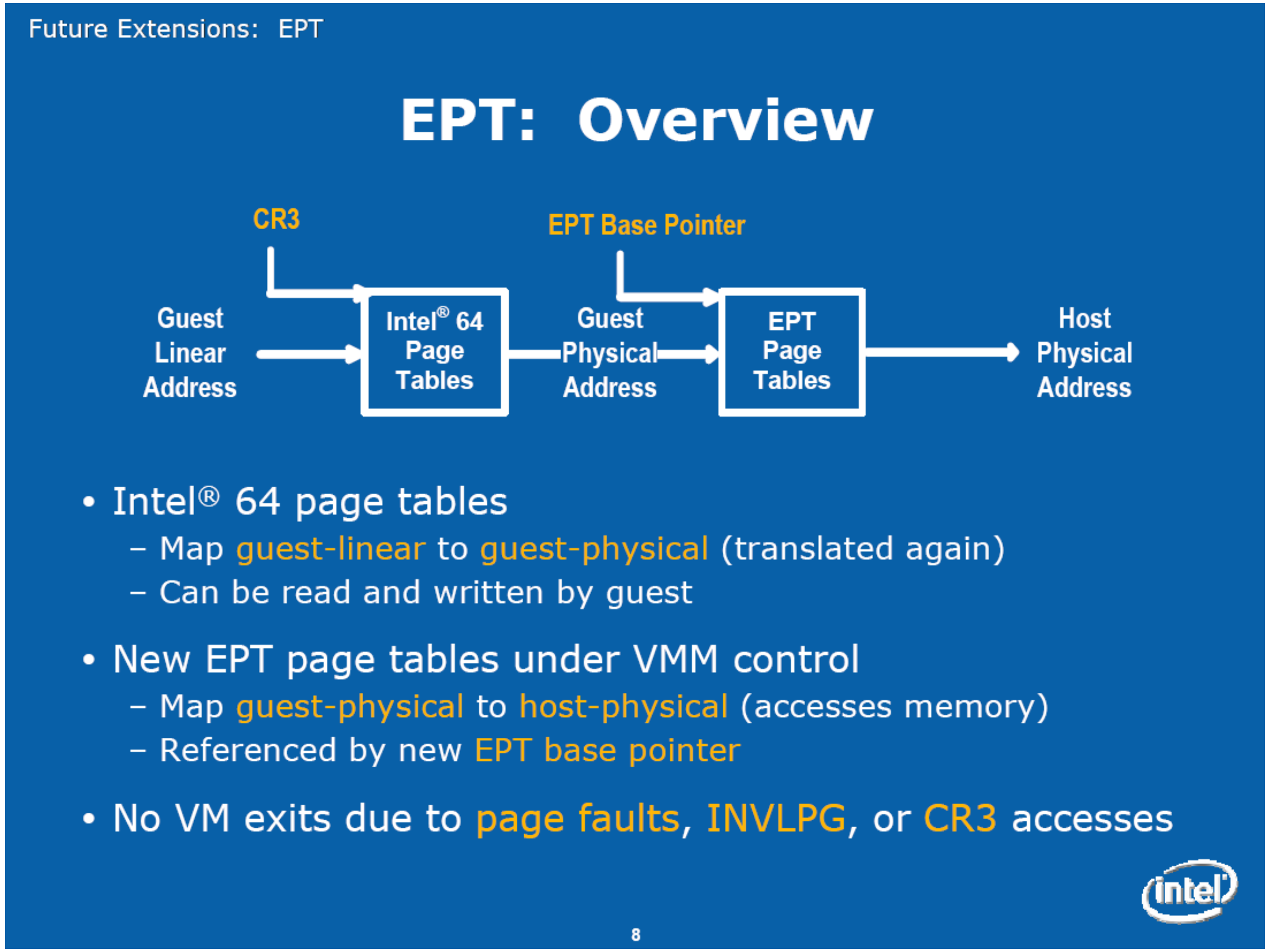
First generation: Intel VT-x & AMD SVM

- Eliminating the need of binary translation



Second generation: Intel EPT & AMD NPT

- Eliminating the need to shadow page table



Third generation: Intel VT-d & AMD IOMMU

- I/O device assignment
  - VM owns real device
- DMA remapping
  - Support address translation for DMA
- Interrupt remapping
  - Routing device interrupt

Para-virtualization

- Full vs. para virtualization