10/8 CS240 - ESX

Announcements

For next class (Tuesday 10/14)

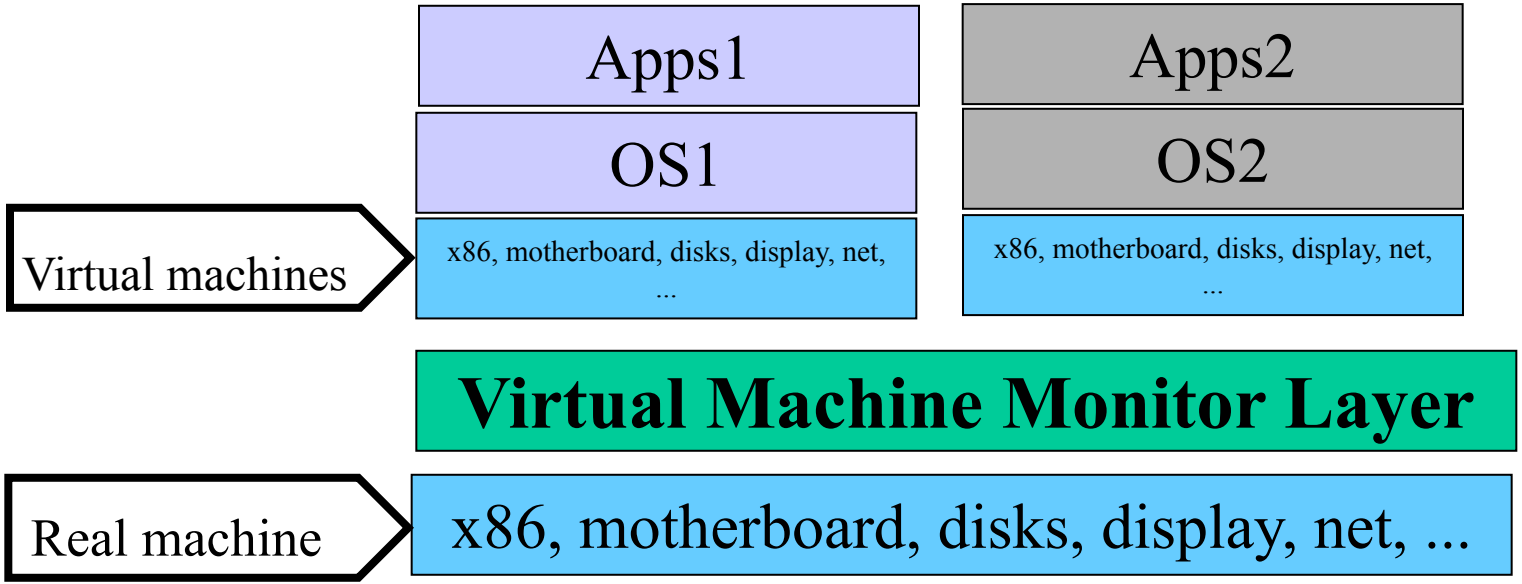
1. Read: Practical, Transparent Operating System Support for Superpages
2. Submit answers to reading questions (see course schedule) before class

Remember: Lab 1 is due end of day November 2nd.

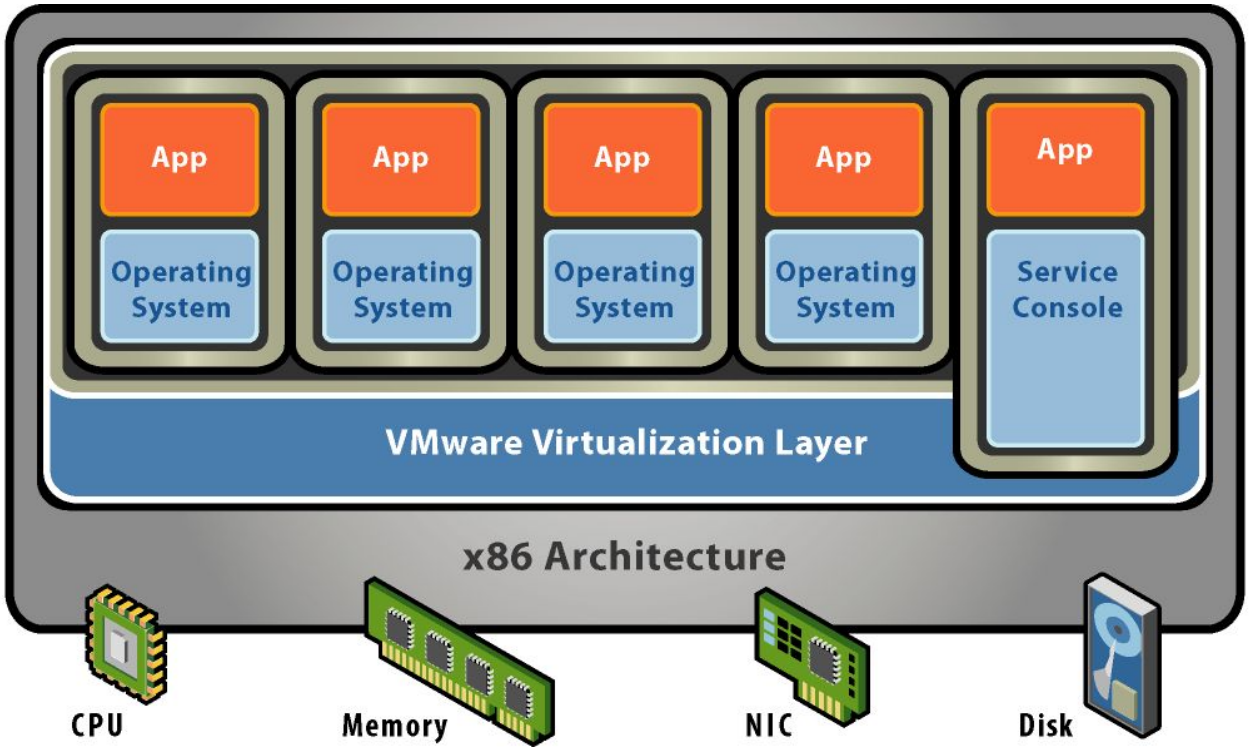
Paper background

- OSDI 2002 - 5th Symposium on Operating Systems Design and Implementation
  - Best paper award, SIGOPS Hall of Fame Award
- Carl - MIT -> DEC WRL -> VMware - Disco connection
- Example of an industry paper from engineer in product group

My virtualization slide 1999



VMware ESX Server



Why bring back virtual machine monitors?

- References from 1970s: “VM/370: A Study of Multiplicity and Usefulness”
  - 1979 IBM Systems Journal - Seawright and McKinnon
  - VMMs not carried forward with mainframe -> minicomputers -> workstations -> x86 computers
    - Interesting when machines were rare and expensive …
      - … not with good OSes on inexpensive computers
- Cloud computing? Dot com bust
  - 2002 Server Consolidation
    - Run many VMs (typically running Windows) on a beefy machine
      - Can't change guest OS and apps to run in VM
      - Too costly to just divvy up machine memory among VMs
  - Need VMM to be significantly more reliable than Windows
    - Eggs in one basket - isolation and fault containment needed to work

Intel IA32 memory management review

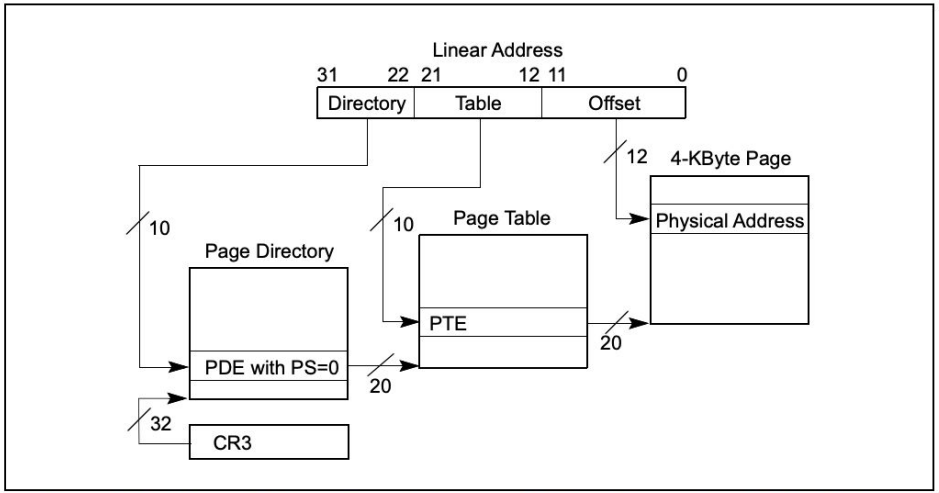


Figure 5-2. Linear-Address Translation to a 4-KByte Page using 32-Bit Paging

Intel MMU:

- 4K & 4M page sizes
- Hardware TLB with reload using 2-level page table walking

Virtual Memory concepts:

Copy-on-write (COW)

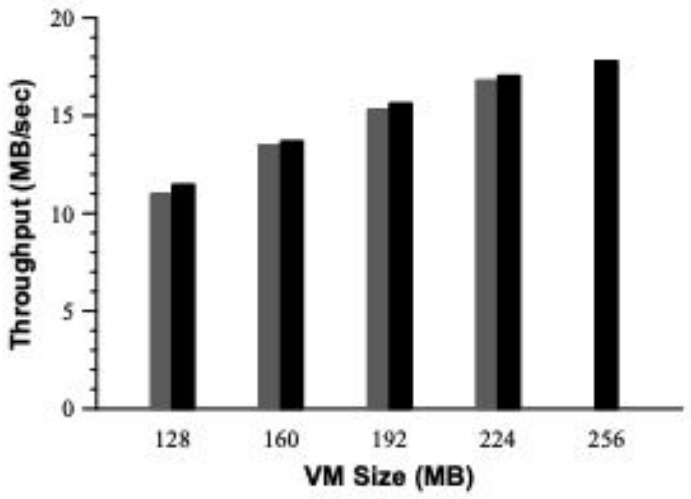
Page coloring

ESX memory virtualization

- *pmap* data structure: maintain map (VM,PPN -> MPN)
- Shadow page table approach
  - Required to use Intel MMU
- Powerful, transparent

Memory Reclamation

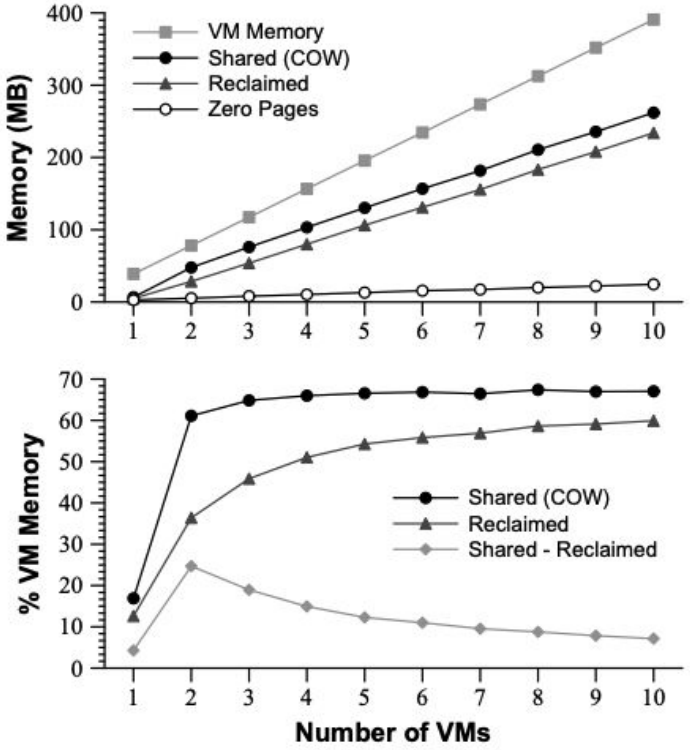
- Overcommit
  - *max size* per VM
  - Interfaces to add and remove memory from guest OSes would have been useful
- Double paging problem?
  - How did ESX avoid it?
- Balloon driver / kernel service in guest?



Content-based Page Sharing

- Why is this called transparent sharing?
- What mechanism is used to find pages to share?
  - Why don't they hash or enter every scanned page?

Best case sharing: Linux running SPEC95 benchmark



Real-world Page Sharing

		Total	Shared		Reclaimed	
	Guest Types	MB	MB	%	MB	%
A	10 WinNT	2048	880	42.9	673	32.9
B	9 Linux	1846	539	29.2	345	18.7
C	5 Linux	1658	165	10.0	120	7.2

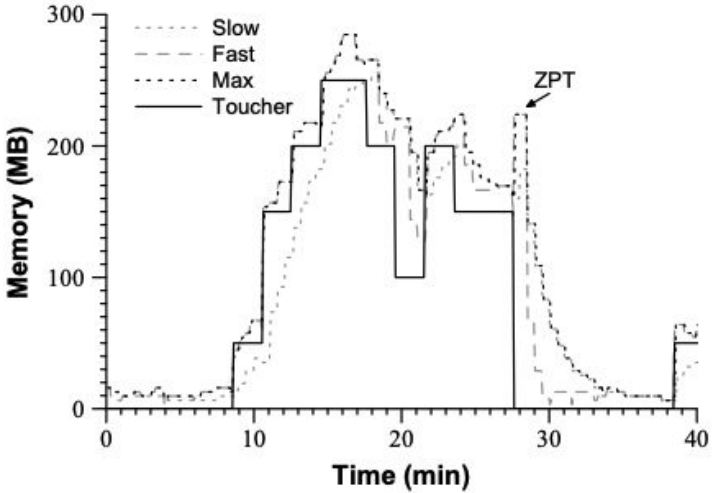
Share-based Allocation

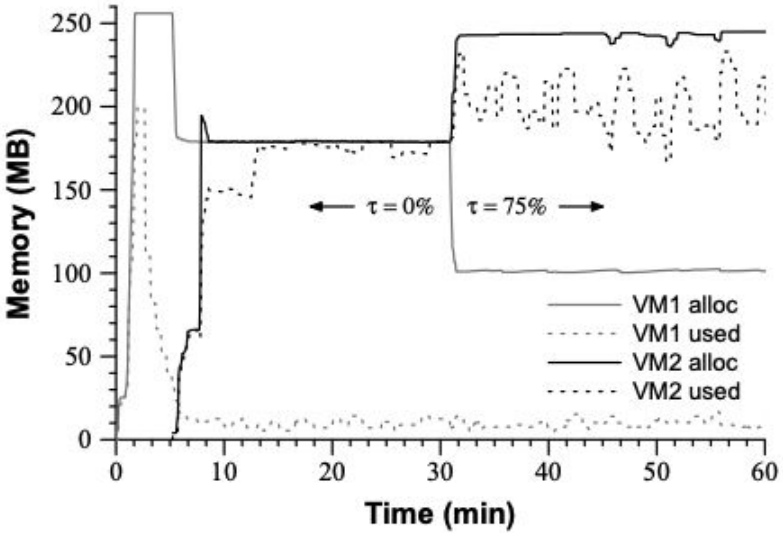
- Why not use same algorithm as in OS virtual memory systems?
  - Carl's PhD thesis?
- Min-funding revocation algorithm
- How can isolation and efficiency conflict?

Idle Memory Tax

$$\rho = \frac{S}{P \cdot (f + k \cdot (1 - f))} \qquad k = 1/(1 - \tau)$$

- What are ρ, *S*, *P*, *f* and τ here?
  - What does a tax of 75% (the default) do?
- How do they measure idle memory?





Allocation Policies

- Parameters?:
  - Min size
  - Max size
  - Shares
- Admission Control Policies
  - States
    - High 6%
    - Soft 4%
    - Hard 2%
    - Low 1%

I/O Page Remapping

- What's the problem here?

Today's environments

- OSes have been updated to know they are running in a VM
  - Support dynamic adding memory
  - Some removing memory supported (ballooning still being used)
- Hardware support for virtualization

Reading Questions

1. What happens if a virtual machine running under ESX reboots and starts accessing memory that belonged to the balloon driver?
2. Most OSes have some sort of `mpin(void* va, unsigned len)` syscall, which will pin a range of addresses in physical memory. Explain how to use this system call to replace the need to write a balloon driver. Make sure to describe how to communicate with ESX. What is the advantage/disadvantage of this approach?