# ESX Server

Emmett Witchel

CS380L

# Virtualization faux quiz (pick 2, 5 min)

- What is the “double paging” problem?
- How does a memory balloon work? What happens when a guest OS writes to memory owned by the balloon driver?
- How do Xen memory virtualization techniques differ from ESX?
- Compare and contrast interposition techniques in ESX, Xen, Arrakis

# ESX Memory management

**Goal:**

- support memory overcommit

**Constraints:**

- full virtualization

**The problems:**

- multiple resource managers/visibility
- VM page replacement doesn’t know what’s important to guest OS
  - How does this manifest?
- Double paging problem:
  - VM page out followed by OS page out → VM reclaim.
- Challenge: OSes can’t dynamically change physical memory size

# ESX Memory management

What are the main contributions?

**Goal:**

- support memory overcommit

**Constraints:**

- full virtualization

**The problems:**

- multiple resource managers/visibility
- VM page replacement doesn't know what's important to guest OS
  - How does this manifest?
- Double paging problem:
  - VM page out followed by OS page out → VM reclaim.
- Challenge: OSes can't dynamically change physical memory size

# ESX Memory management

What are the main contributions?

Ballooning
Content-based sharing
Idle memory taxation
Hot I/O Page remapping

**Goal:**

- support memory overcommit

**Constraints:**

- full virtualization

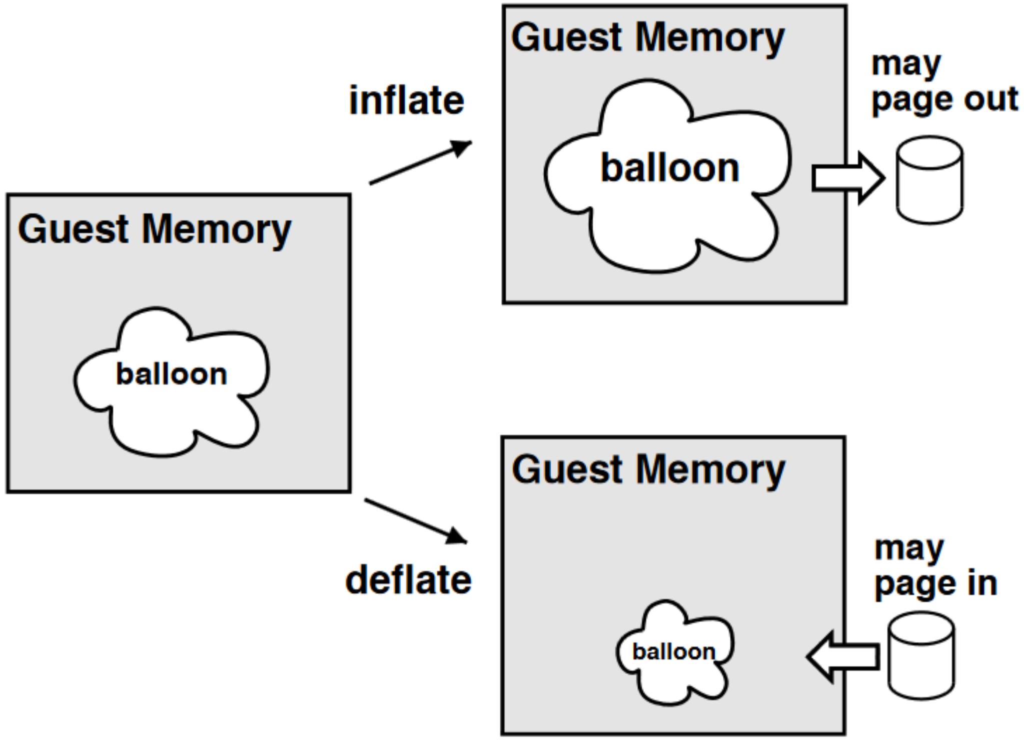
**The problems:**

- multiple resource managers/visibility
- VM page replacement doesn’t know what’s important to guest OS
  - How does this manifest?
- Double paging problem:
  - VM page out followed by OS page out → VM reclaim.
- Challenge: OSes can’t dynamically change physical memory size

# Ballooning!

Force guest OS to use its own page replacement algorithm

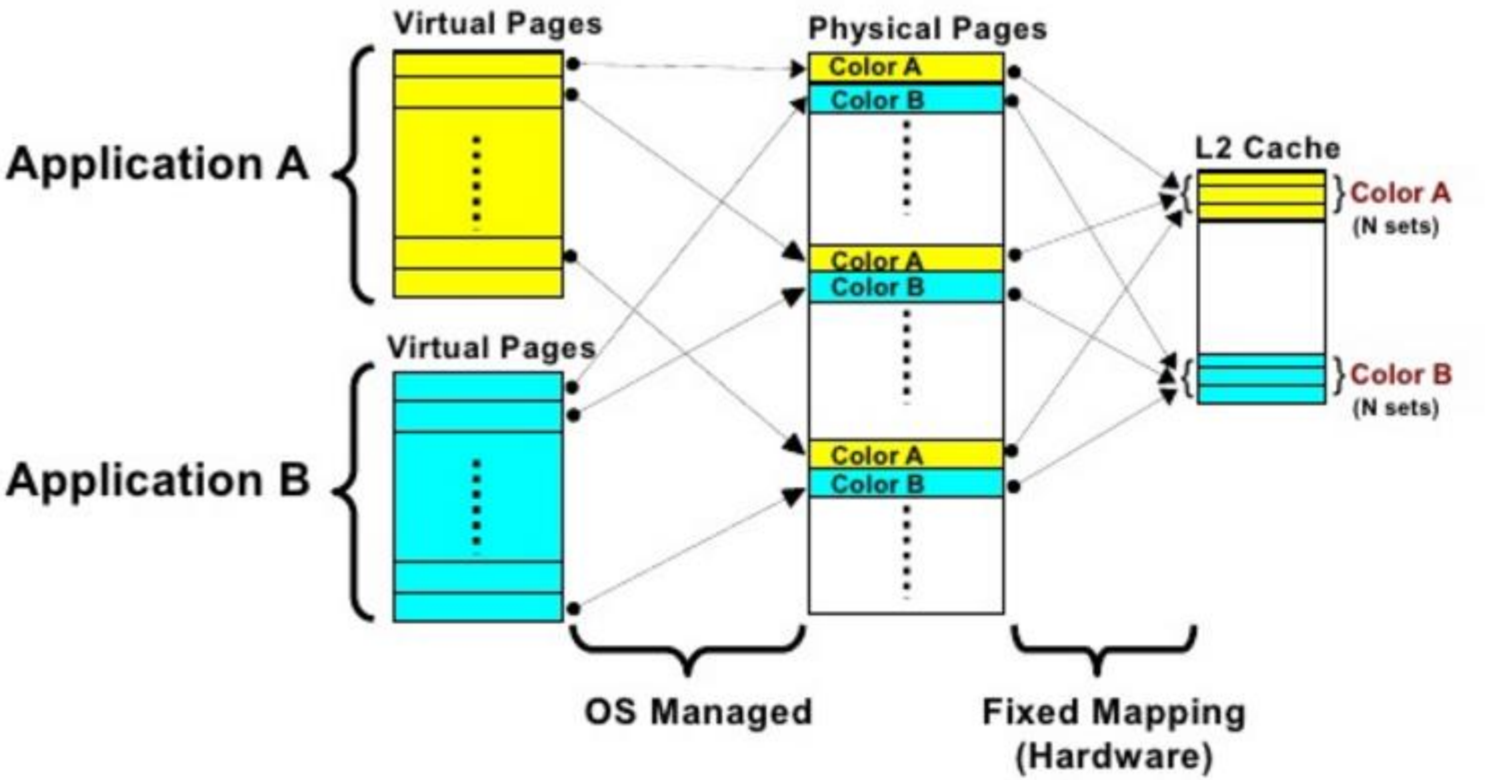
- VMware balloon module in guest OS
- Inflate balloon to get OS to free memory
- Deflate to get OS to use more memory
- Pages allocated to balloon are marked in the pmap, can be realloc'ed
- What happens if a guest touches a balloon page?
- How does ESX control the balloon?
- Is this still full virtualization?



# Balloon problems

- Might not be able to reclaim memory fast enough (VMware rate-limits allocation)
- Guest OS can refuse memory allocation request or limit the driver
  - Can always resort to paging.
  - How to avoid pathologically bad cases of paging out what guest needs?
- VMware PFN to MPN mapping preserves page coloring.
  - What is page coloring?

# Page coloring



Can be used to reduce cache misses (trad OS goal) or partition resources (cache partitioning)

*Can anyone see any downsides to this?*

# Content-based Sharing

How does it work?

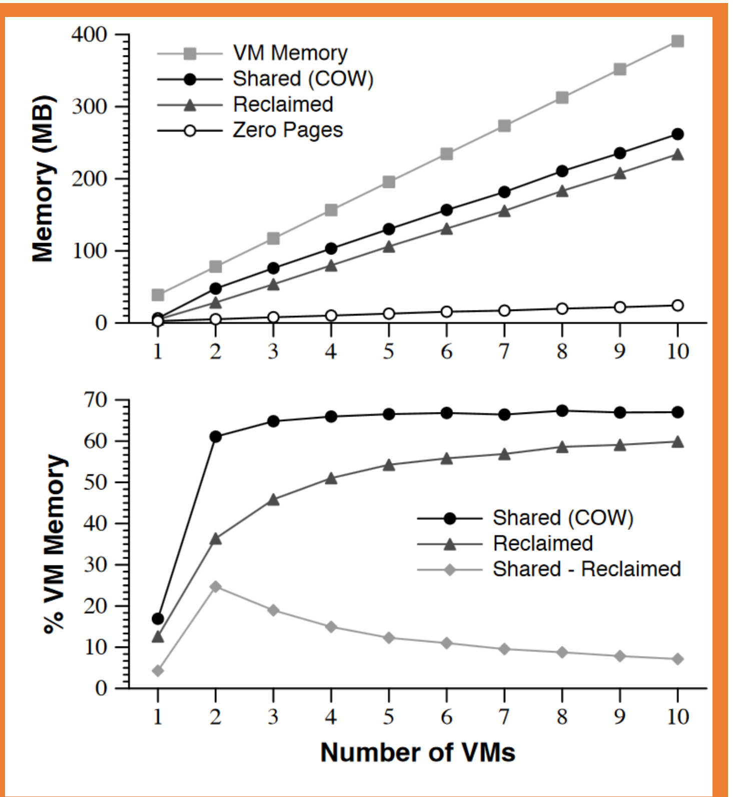
- Hash every page, store hashes in a hash table.
- On collision, check if pages are identical (why?)
  - If they are, share copy-on-write (what’s CoW?).
  - With no collision, store hash as hint.
- On future collision: hint still valid? (page contents unchanged).
  - If it is, share page.
- Data structure details:
  - 16 byte records (0.5% of system memory) : [hash value, MPN, ref count, chain link]
  - store ref count for shared pages with overflow table for widely shared pages (ref count?)
- How/when to do this? Does it use too much CPU time? What policy is used? Can you think of better policies?

# Content-based Sharing

Identical linux

VMs running

SPEC95



How does it work?

- Hash every page, store hashes in a hash tab
- On collision, check if pages are identical (wh
  - If they are, share copy-on-write (what's CoW?).
  - With no collision, store hash as hint.
- On future collision: hint still valid? (page contents unchanged).
  - If it is, share page.
- Data structure details:
  - 16 byte records (0.5% of system memory) : [hash value, MPN, ref count, chain link]
  - store ref count for shared pages with overflow table for widely shared pages (ref count?)
- How/when to do this? Does it use too much CPU time? What policy is used? Can you think of better policies?

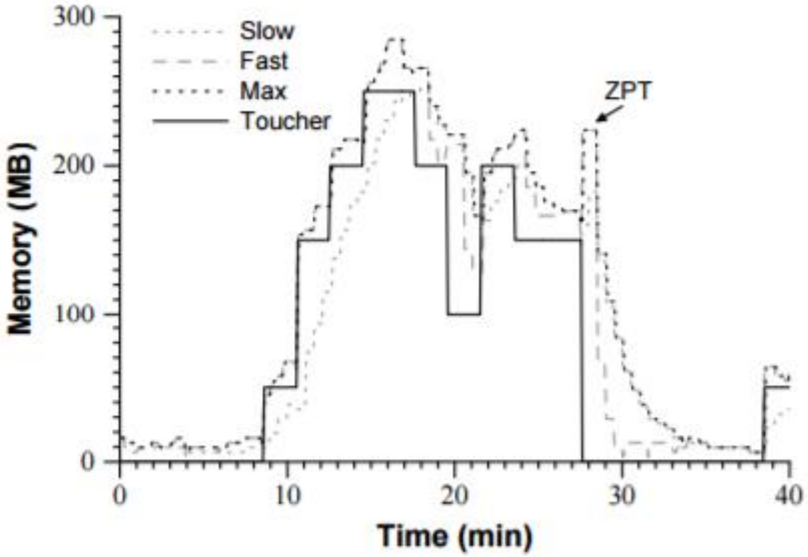
# Managing Memory with Taxes

- What’s wrong with proportional share allocation?
- How does ESX fix the problem?
- Idle memory tax!
    - Inflate cost of idle memory by tax rate
    - Charge a client more for its idle pages than for the ones it is actually using
    - When the system needs memory, pages reclaimed first from clients that are not actively using their full allocated memory.
    - Allow 25% idle memory for fast-growing working set increase.
        - →Max fraction that can be reclaimed: 75%
    - Only need percentage of idle memory: measure by random sampling

# Idle Memory Tax

- Rho = shares per page
- S = # shares, P = # pages
- f = active fraction, k = idle page cost (encodes tax rate)
- For f, multiple moving averages
- Four states, with hysteresis:
  - high: No reclamation
  - soft: Balloon, page if needed
  - hard: Page
  - low: Suspend VM

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



# Idle Memory Tax Evaluation

Two VMs with same share allocations (256 MB on overcommitted system.

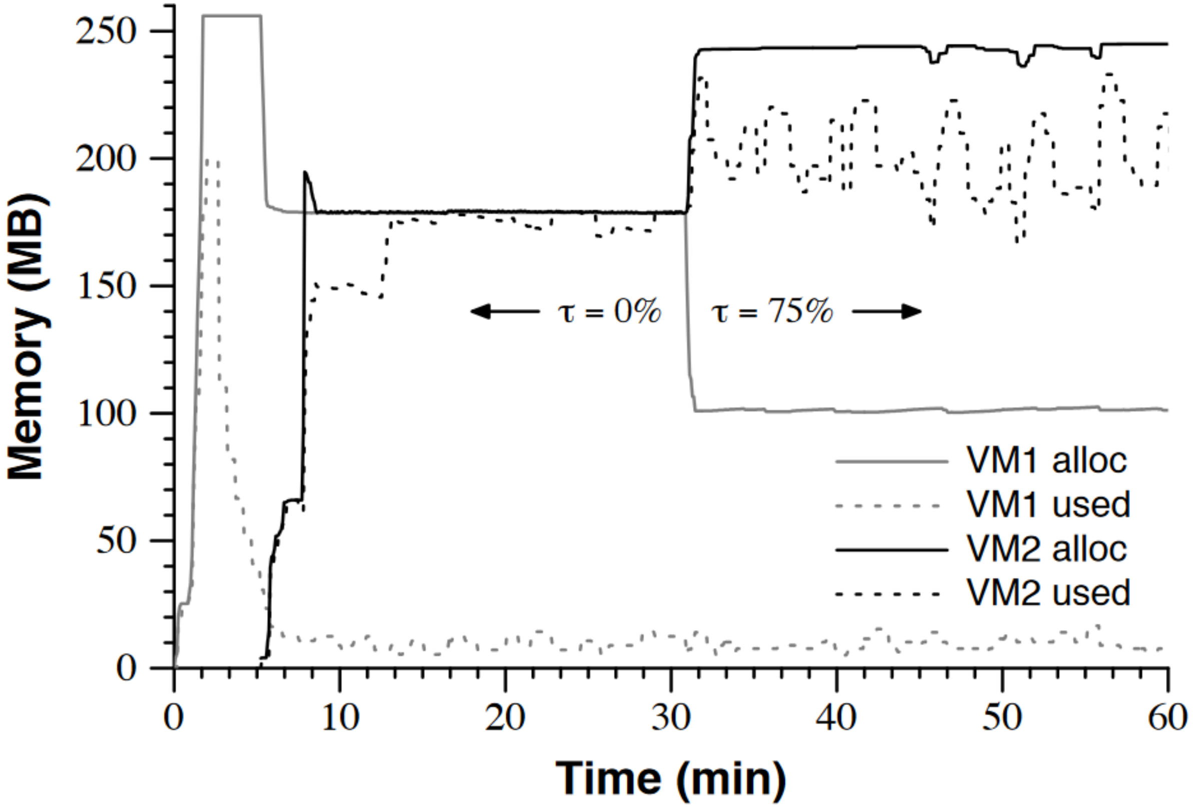


Figure 7: **Idle Memory Tax.**

# Hot I/O Page Remapping

- “Modern” processors can address up to 64GB of memory.
- Devices that use DMA for IO can only address up to 4 GB of memory.
- Traditional approach: copy “high” memory into a buffer in “low” memory.
    - Expensive, and even worse in the case of VMs since VMs that think they have “low” memory might actually be mapped to high memory!
- ESX server tracks ‘hot’ pages that are involved in a lot of I/O, and when it reaches a certain threshold, the page is transparently remapped to low memory.