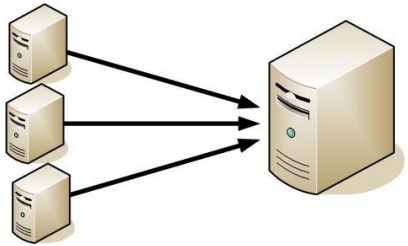
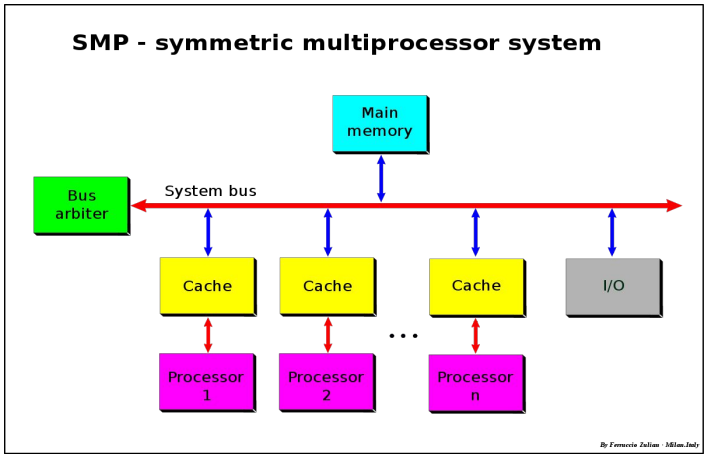
Memory Resource Management in VMware ESX Server

Surya Suresh

Background

- Industry trends have motivated the need for server virtualization
  - Server consolidation
  - Shared memory multiprocessors
- Servers are underutilized in many computing environments, can be expensive to maintain
- SMP architectures are difficult to leverage on an operating system (NUMA, cache coherency)



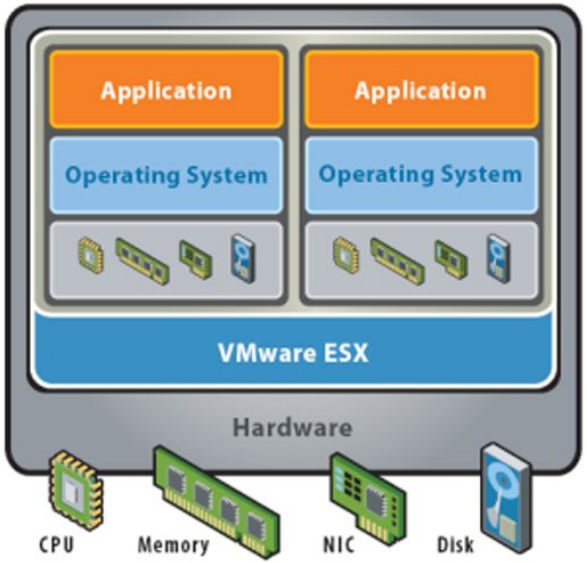


Motivation

- Run commodity operating systems without modifying them
- Manage hardware resources directly
- Make efficient use of memory and page reclamation

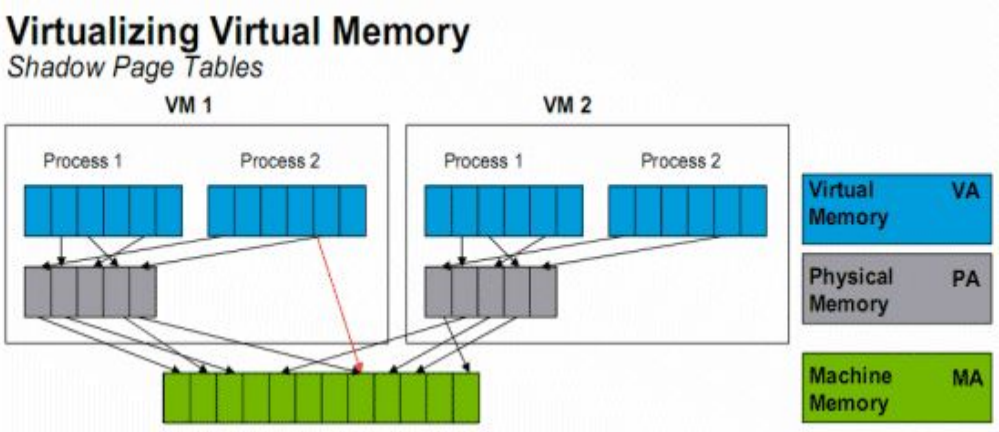
VMware ESX

- A virtual machine monitor (type 1 hypervisor)
- Very similar to Disco
- Idea of virtualization reused, novel memory management policies



Memory Virtualization

- Virtual, “physical”, machine memory addresses
- Pmap data structure
  - “Physical” to machine page number
- Layer of redirection allows for ease of remapping pages
- Guest OS page table and tlb modifications are intercepted, shadow page tables used for virtual address to machine translation



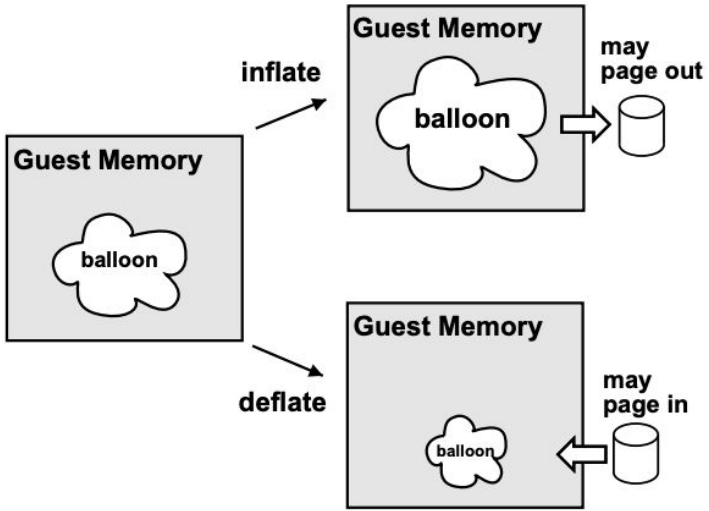
Reclamation Mechanisms

Page Replacement Issues

- Problem: Reclaim space when memory is overcommitted
  - Total virtual memory allocated exceeds physical memory
- Meta page table replacement policy not optimal
  - VMM doesn't know what pages to replace
  - Should aim to have guest O.S issue pages to replace
- Guest O.S and meta page replacement policies can lead to double paging problem
  - System chooses page to reclaim only to have the same page be used by the guest for virtual paging device

Ballooning

- Technique to get the guest O.S to cooperate with ESX
- Small balloon module loaded into the guest O.S
  - Inflates when ESX wants to reclaim pages
  - Defaulting allows the guest to gain back memory
- Guest O.S management policies are put to work under high memory pressure



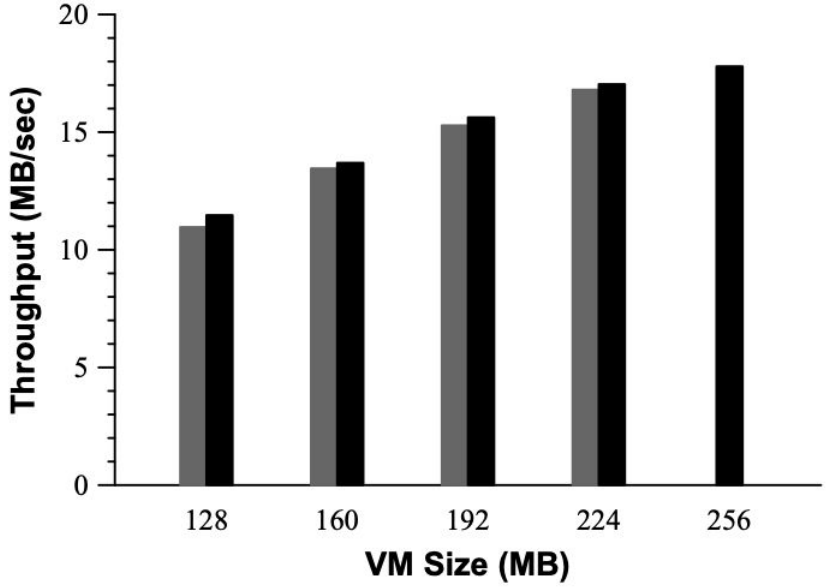
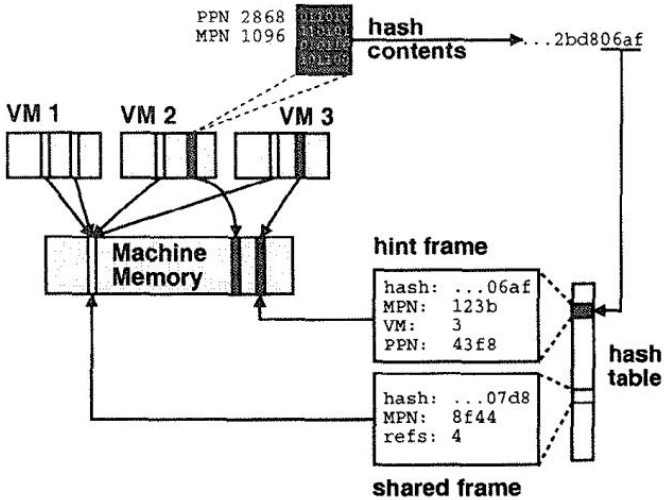


Figure 2: **Balloon Performance.** Throughput of single Linux VM running `dbench` with 40 clients. The black bars plot the performance when the VM is configured with main memory sizes ranging from 128 MB to 256 MB. The gray bars plot the performance of the same VM configured with 256 MB, ballooned down to the specified size.

Sharing Memory

Content Based Sharing

- Overcommitment can place memory stress on guest O.S's
- Share machine pages for VMs that use the same data (same O.S, same application)
- Copy on write when a VM modifies the page
- Utilize a hash function to find page content matches



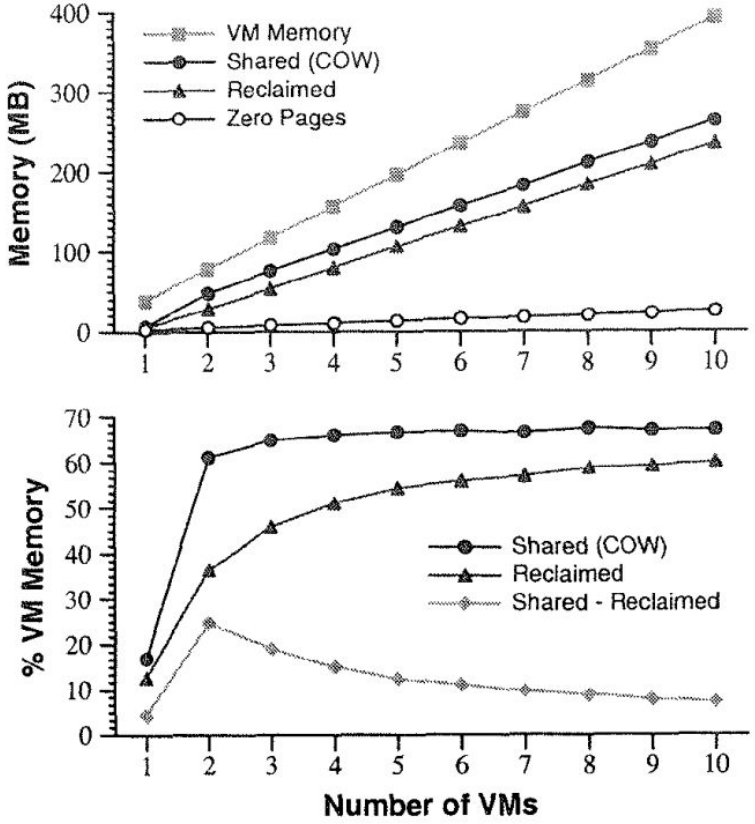


Figure 4: **Page Sharing Performance.** Sharing metrics for a series of experiments consisting of identical Linux VMs running SPEC95 benchmarks. The top graph indicates the absolute amounts of memory shared and saved increase smoothly with the number of concurrent VMs. The bottom graph plots these metrics as a percentage of aggregate VM memory. For large numbers of VMs, sharing approaches 67% and nearly 60% of all VM memory is reclaimed.

Share Based Allocation and Idle Memory

- Clients (VMs) are allocated resources based on the number of shares it has
- Revoke resource from a lower paying client to a higher paying client (based on shares)
- Pure shares per page ratios can lead to issues such as idle memory
- Idle memory tax: charge more for idle pages
- Give pages back when client starts using more

Min-funding revocation is extended to use an adjusted shares-per-page ratio. For a client with S shares and an allocation of P pages, of which a fraction f are active, the adjusted shares-per-page ratio ρ is

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

where the idle page cost $k = 1/(1 - \tau)$ for a given tax rate $0 \leq \tau < 1$.

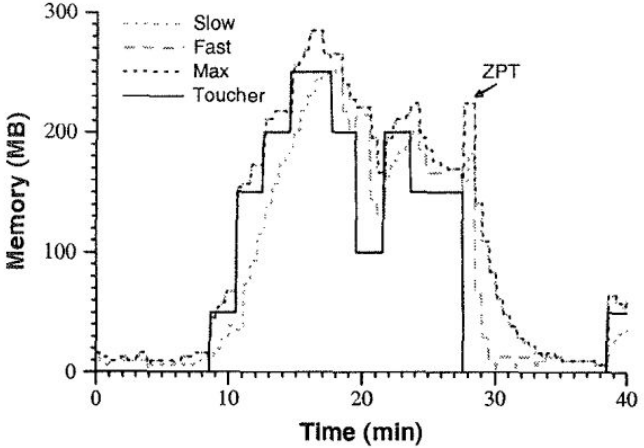


Figure 6: **Active Memory Sampling.** A Windows VM executes a simple memory *toucher* application. The solid black line indicates the amount of memory repeatedly touched, which is varied over time. The dotted black line is the sampling-based statistical estimate of overall VM memory usage, including background Windows activities. The estimate is computed as the *max* of *fast* (gray dashed line) and *slow* (gray dotted line) moving averages. The spike labelled *ZPT* is due to the Windows "zero page thread."

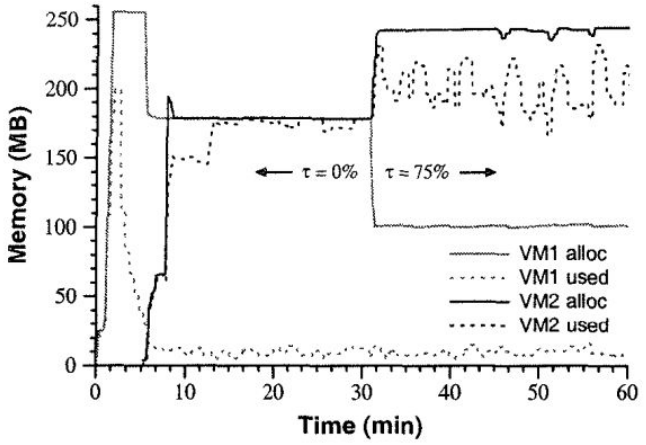


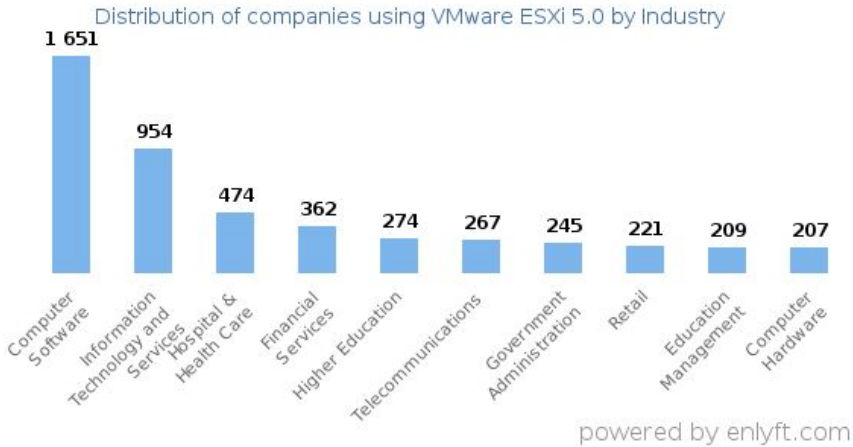
Figure 7: **Idle Memory Tax.** Two VMs with identical share allocations are each configured with 256 MB in an overcommitted system. VM1 (gray) runs Windows, and remains idle after booting. VM2 (black) executes a memory-intensive Linux workload. For each VM, ESX Server allocations are plotted as solid lines, and estimated memory usage is indicated by dotted lines. With an initial tax rate of 0%, the VMs each converge on the same 179 MB allocation. When the tax rate is increased to 75%, idle memory is reclaimed from VM1 and reallocated to VM2, boosting its performance by over 30%.

Allocation Policies

- Target memory allocation computed via parameters
  - Min size: guaranteed lower bound for memory
  - Max size: amount of “physical” memory allocated
  - Shares: metric for how much memory VM is obligated to receive
- Dynamic reallocation
  - Uses thresholds for different reclamation states high, soft, hard, low
  - No reclamation, ballooning, forcibly remove pages, remove and block execution

Impact

- ESX inspired ESXi design
- VMware ESXi widely used today, 8860 companies!



Conclusion

- ESX server is a virtual machine monitor with efficient memory management techniques
- Ballooning
- Page sharing
- Page shares and memory idle tax
- Dynamic allocation for page reclamation

Discussion

- Is the min-funding revocation algorithm a fair way to reclaim pages for all the clients? Is it better to have homogeneous workloads in the same ESX server?
- What happens if all workloads are utilizing their full memory capacity? Should all workloads do page swapping to virtual disk?
- How does VMware ESX compare with Xen? Do you think paravirtualization is better or worse than full virtualization?