Memory Resource Management in VMware ESX Server

一十百千萬億組
김도형, 이승수

MAY 7, 2019

Real-Time Ubiquitous Systems Lab.

Table of Contents

Motivation-Virtual Machine

- Interest in virtualization techniques ↑
- Server Consolidation
- Utilization ↑
- Run Isolated VMs in Same Physical Machine
- VMware workstation, Disco, IBM System/370…

VMware ESX Server

- **VMware ESX Server**

  - Type 1 VMM – No host OS, Directly Implemented

  - Higher I/O performance

- **Challenge**

  - Run OS without Modification

  - Unable to Influence the Design of OS

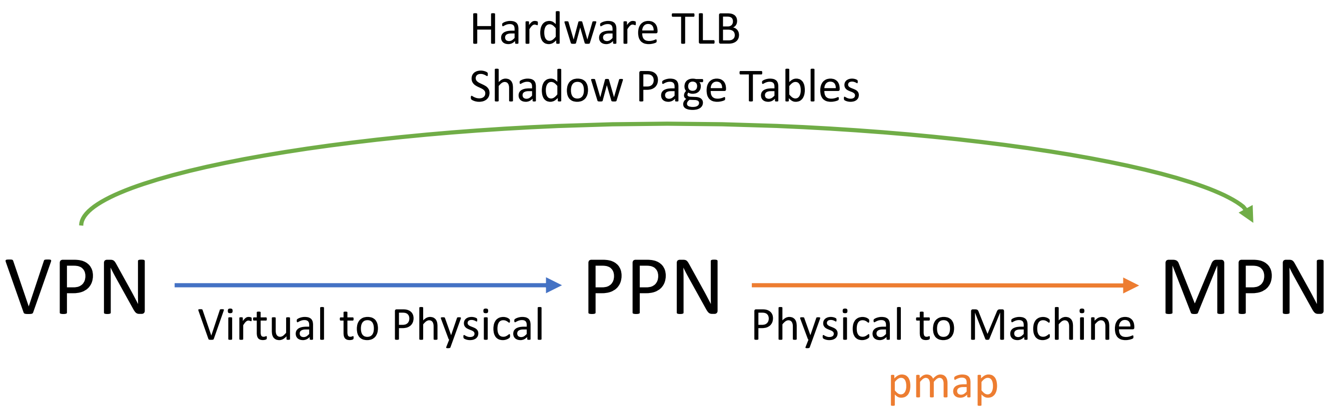
- **Goal**

  - Efficient Memory Management on running existing OS without modification

Contribution

- Ballooning – Reclamation Mechanisms
- Content-based page Sharing – Sharing Memory
- Idle memory tax – Sharing Memory
- Hot I/O page remapping – Page Remapping

Memory Virtualization



Reclaiming Memory

- **Memory Overcommitment**

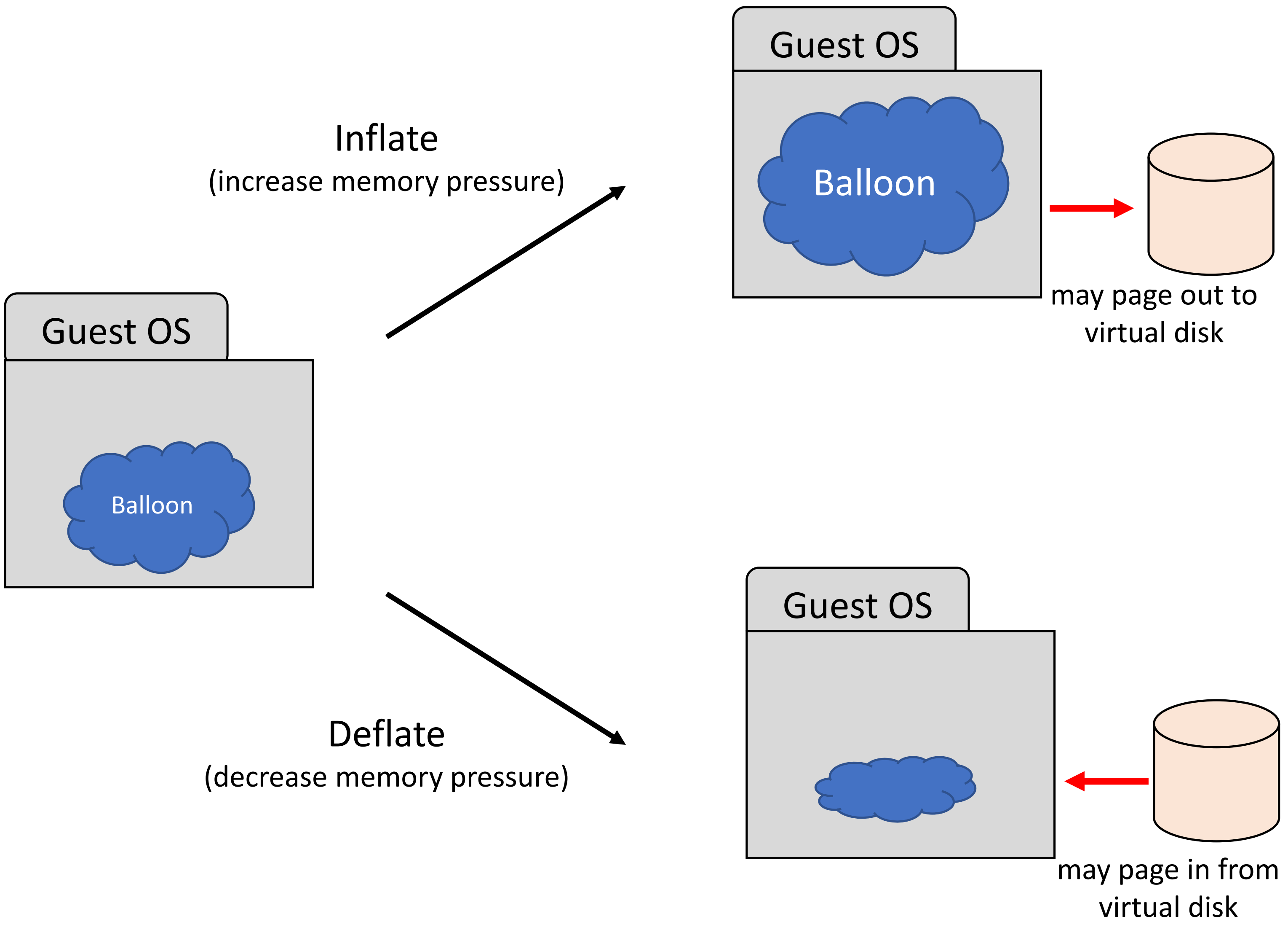
  - ($\sum$ memory allocated for each VM) $>$ (Total Machine Memory)

- **Traditional Method**

  - Add Transparent Swap Layer
  - Meta-Level Page Replacement Decisions (only known by Guest OS)
  - Guest OS and Meta-Level Policies may clash

    Ex) Double Paging

Ballooning


Guest OS
Balloon
Inflate
(increase memory pressure)
Guest OS
Balloon
may page out to
virtual disk
Deflate
(decrease memory pressure)
Guest OS
may page in from
virtual disk

Ballooning

- **Performance**

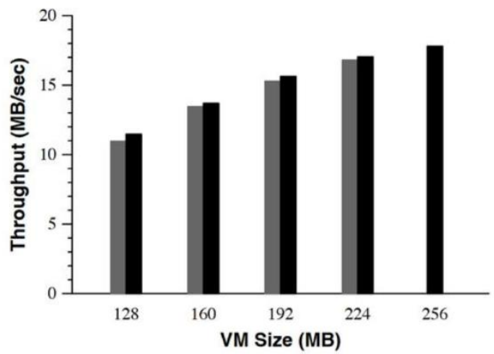


Figure 2: **Balloon Performance.** Throughput of single Linux VM running dbench with 40 clients.

- Compare i) & ii)
  - i) Grey Bars
    - 256MB with balloon sized 32 – 128 MB accordingly
  - ii) Black Bars
    - Static Virtual Machines

- Result
  - Overhead 1.4 % (32MB) - 4.4% (128MB)

- **Limitation**
  - Not available all the time : OS boot time, driver explicitly disabled
  - Not fast enough to satisfy current system demand
  - Guest OS might have limitations to upper bound on balloon size

⇨ ESX Sever Swap Daemon / Randomized Page Replacement

Sharing Memory

- Same OS, Same Apps, Same Data
- Support Overcommitment ↑↑

- Traditional Method : Transparent Page Sharing(Disco)

  - Several Physical Page -> Same Machine Page

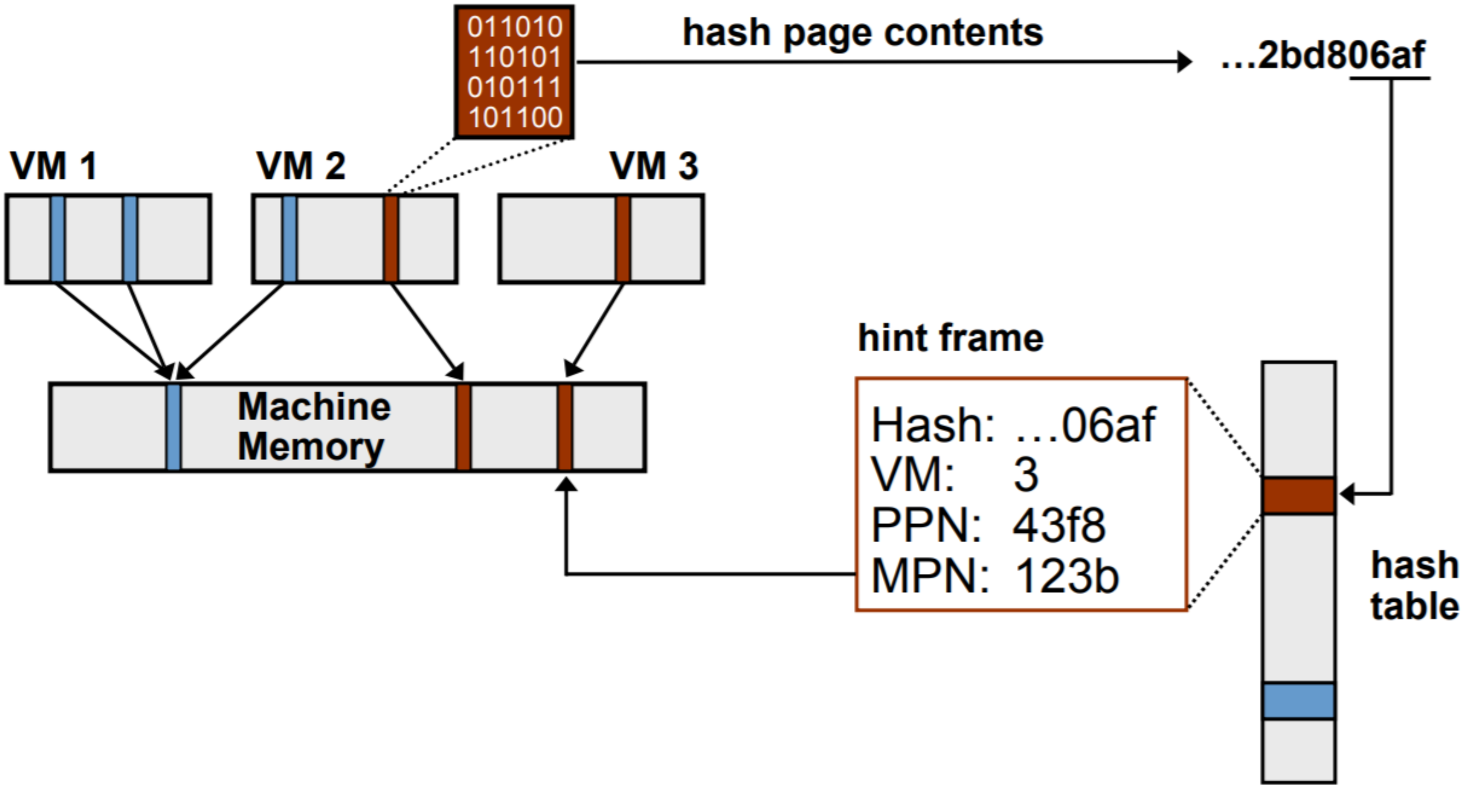
  - Copy-on-write

  - Guest OS Modifying Needed

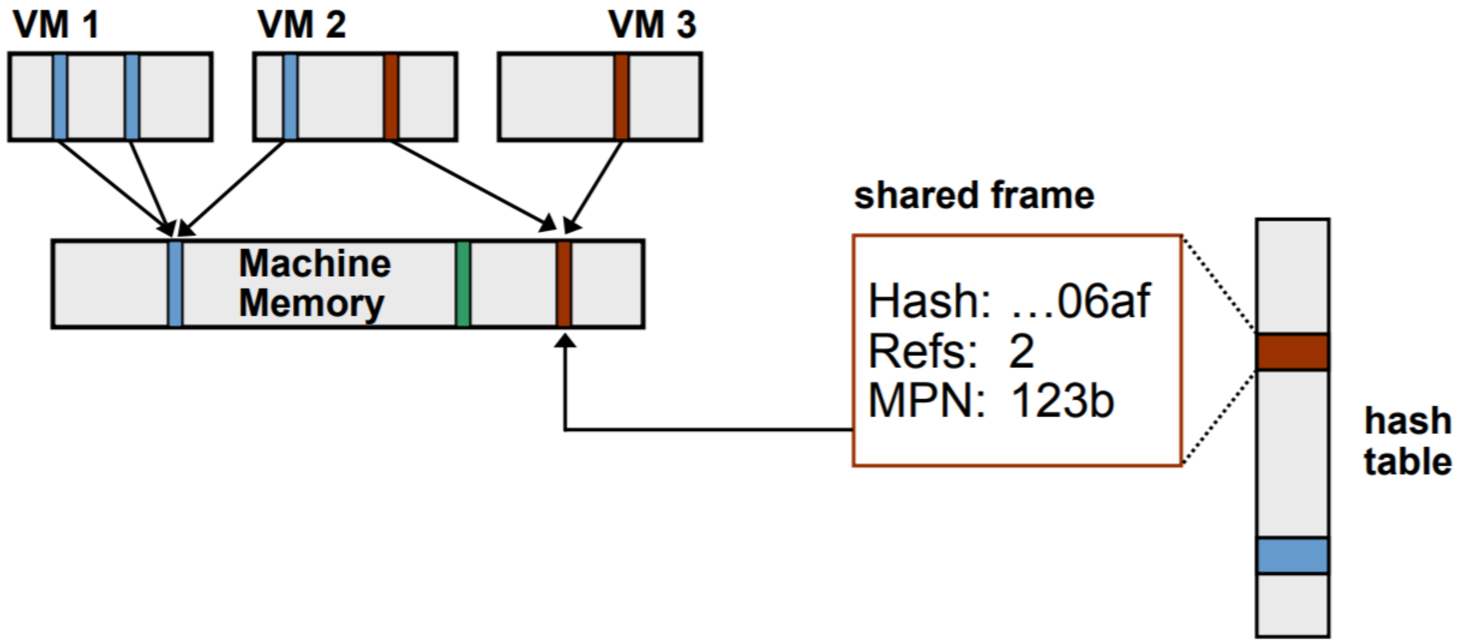
  - Restricted Interfaces

Content-Based Page Sharing

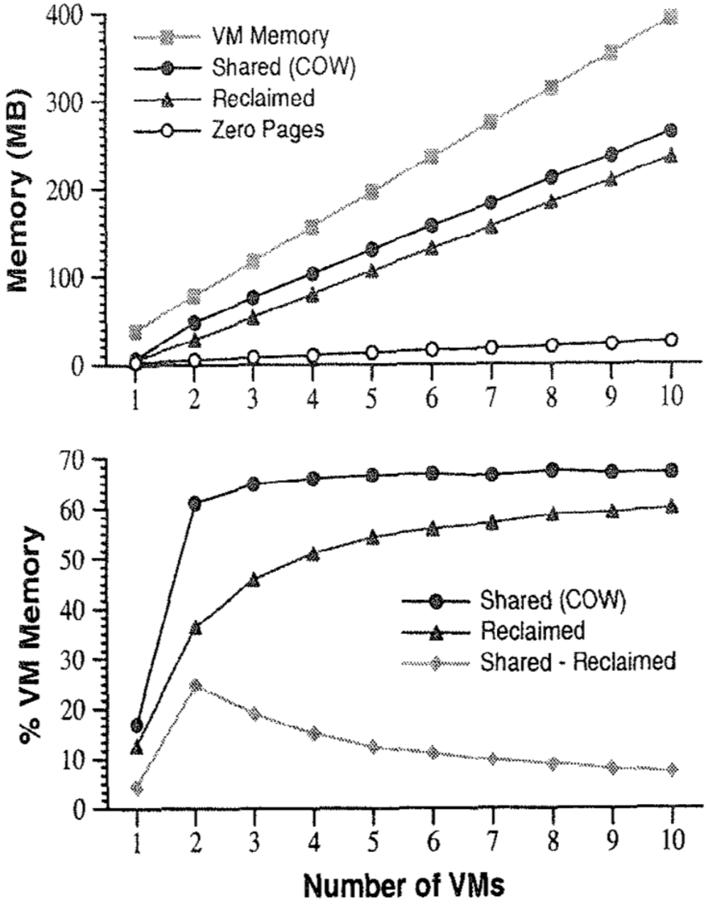
Idea : Same Contents, Same Memory



Content-Based Page Sharing



Content-Based Page Sharing



- “Best Case”

  - Same OS, Same Contents

- Linearly Increase

  - Jump between 1~2

Content-Based Page Sharing

- Worked Well in Different Guest Types

		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

Content-Based Page Sharing

- Advantage

  - No Guest OS Changes Even No Understand Code

  - General Purpose

- Overhead

  - Scan Pages Randomly

  - Negligible CPU Overhead

Shares vs. Working Set

- **Shared-based Allocation**
  - Resource rights are distributed to clients through *shares*
  - Resources are allocated proportional to the share
  - **Limitations**

    Not incorporate any information about active memory usage or working set

    Idle Client with more shares

    Active Cl

Idle Memory Tax

- **Idea**
  - To charge a client more for an idle page than for on it is actively using
  - Tax-Rate(τ) : maximum fraction of idle pages that may be reclaimed from a client

    a) τ = 0 : pure share-based isolation

    b) τ ≈ 1 : all of a client's idle memory are reclaimed

    c) ESX Server default tax rate : 75%

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

S : shares

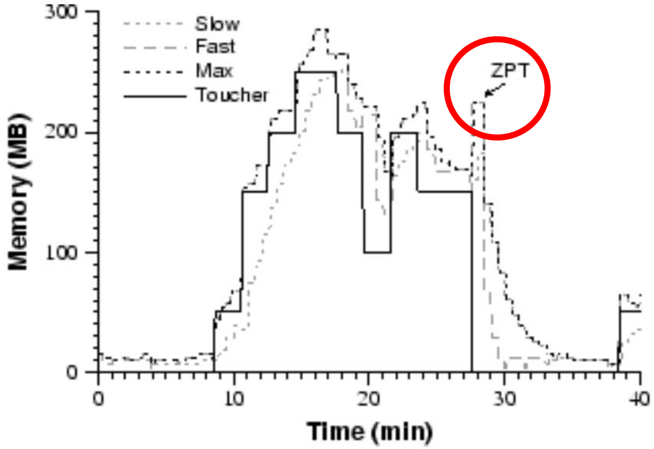
P : allocation of pages

f : active fraction on allocated pages

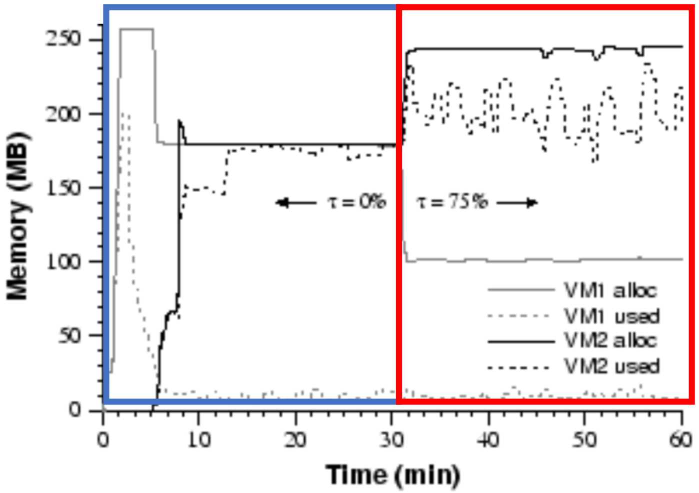
ρ : adjusted shares-per-page ratio

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

Idle Memory Tax



Statistical Sampling Approach



VM1 : Window + idle

VM2 : Linux + dbench

Allocation Policy

- **Parameters**
  - Min size

    Guaranteed, even when overcommitted
  - Max size

    Amount of physical memory

    Unless overcommitted, Allocated Max size
  - Shares

    Based on Proportional-share allocation policy

Allocation Policy

- **Admission Control Policy**

  - When VM is allowed to power on…

  - Ensure (sufficient memory + swap space)

Min+Overhead

Max - Min

VM Graphics Frame Buffer +
Virtualization Data Structures

Pmap
Shadow page tables

Allocation Policy

- **Dynamic Reallocation**

  - ESX Reallocation Memory in Various Events

- **Reclamation Sates**

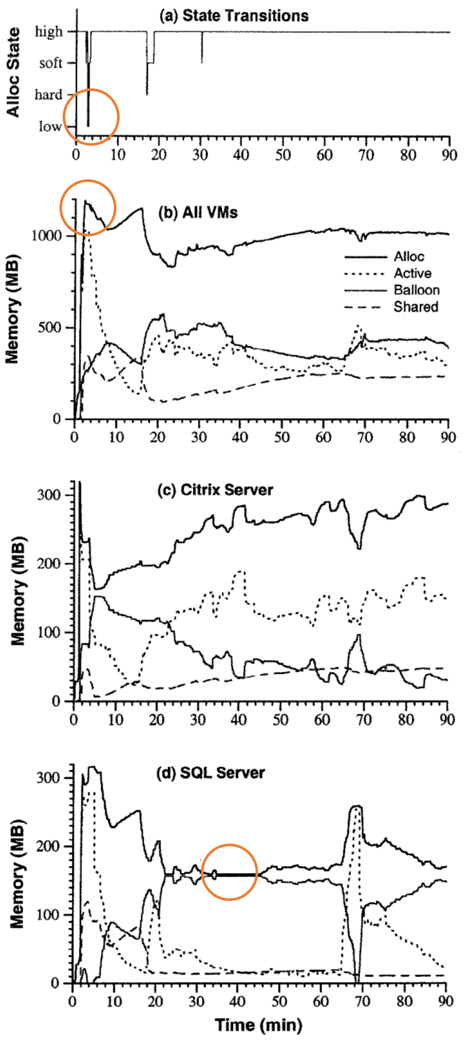
  - High 6% - No reclamation

  - Soft 4% - Ballooning + Paging(Only if Ballooning is not possible)

  - Hard 2% - Paging

  - Low 1% - Block VMs Target Allocation

Allocation Policy



- Experiment Setting

  Machine Memory : 1 GB

  Aggregate VM Workload : 1472 MB

  Additional Overhead : 160 MB

  Overcommitted : 60% ↑

- Nearly All time in high and soft states
- Exceeding the total amount of machine memory
- Drops to the lower bound, “Min” size

I/O Page Remapping

- IA-32 Processors : 36-bit (up to 64 GB) “High” memory
- I/O devise : 32-bit (low 4 GB only) “Low” memory
- Copy data “High” to “Low”

- “Hot” pages : Recently Reference a lot
- Hot I/O Page Remapping

  “Hot” pages in “High” memory → “Low” memory

Conclusions

- **Novel Techniques**

  - Ballooning – Reclamation Mechanisms

  - Content-based page Sharing – Sharing Memory

  - Idle memory tax – Sharing Memory

  - Hot I/O page remapping – Page Remapping

- Currently Exploring

  VMware ESXi ~ (still updating)

Further Work

- Barham, Paul, et al. "Xen and the art of virtualization." *ACM SIGOPS operating systems review*. Vol. 37. No. 5. ACM, 2003.

- Garfinkel, Tal, and Mendel Rosenblum. "A Virtual Machine Introspection Based Architecture for Intrusion Detection." *Ndss*. Vol. 3. No. 2003. 2003

- Garfinkel, Tal, et al. "Terra: A virtual machine-based platform for trusted computing." *ACM SIGOPS Operating Systems Review*. Vol. 37. No. 5. ACM, 2003.

Thank You

Reference

- Waldspurger, Carl A. "Memory resource management in VMware ESX server." *ACM SIGOPS Operating Systems Review* 36.SI (2002): 181-194.
- https://docplayer.net/19029430-Memory-resource-management-in-vmware-esx-server.html