Memory Resource Management in VMware ESX Server

Carl A. Waldspurger

VMware Inc.

OSDI, Dec 2002

Introduction - VMware

Goals

- virtualization and fault containment
- efficiency and scalability

Mechanisms

- **memory sharing**
- over-commitment and **memory reclamation**

Allocation policy for guests

- *min* - guaranteed size
- *max* - maximum size possible
- *share* - **adaptive share value**
- additional overhead for virtualization
- admission control and dynamic reallocation

Some Basics

Memory Virtualization

- virtualizing physical memory
- adding extra level of translation
- *physical address*: guest physical address
- *machine address*: host physical address
- shadow page tables (translate virtual to guest address)

Memory Sharing (1)

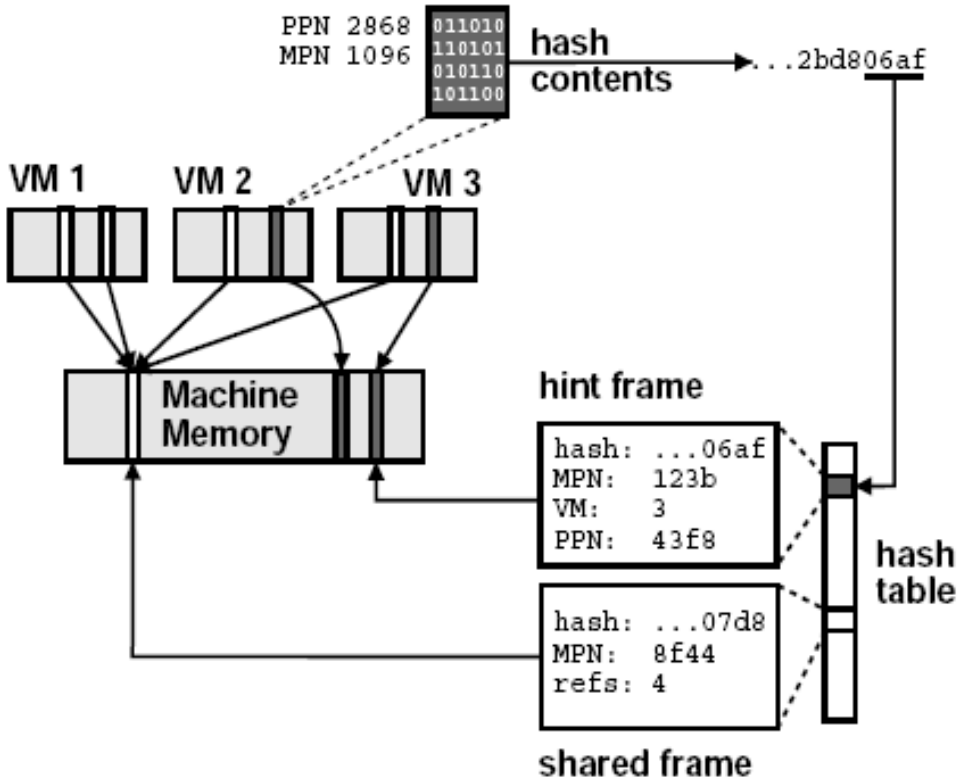
Transparent Page Sharing

- introduced with Disco
- requires guest OS modifications
- non-standard/restricted interfaces

Content-Based Page Sharing

- scanning of page contents
- no modifications of guest OS
- more opportunities for sharing
- naive matching requires O(n2) complexity

Memory Sharing (2)



Memory Sharing (3)

Basic Algorithm

1. mark page as copy-on-write
2. compute hash value
3. lookup into hash table
4. if key in hash table, check for false match
5. if identical reclaim copy

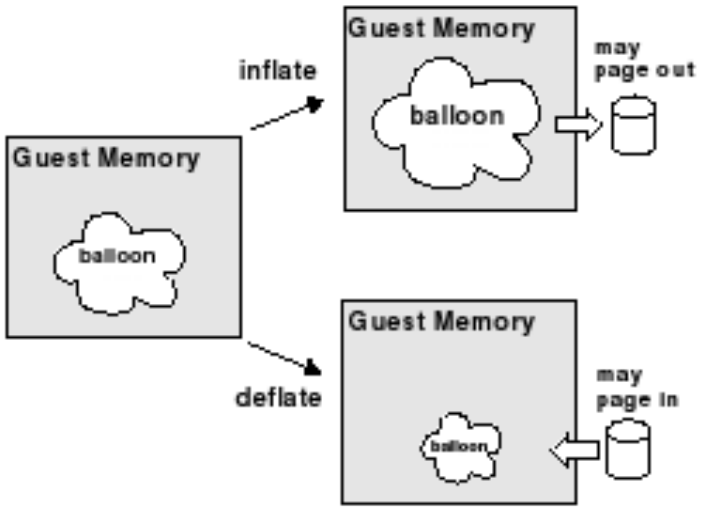
Optimization

- undesirable side-effect: every scanned page is marked read-only
- overhead on subsequent writes
- mark with special hint-bit
- on match rehash the page

Reclamation Mechanisms (1)

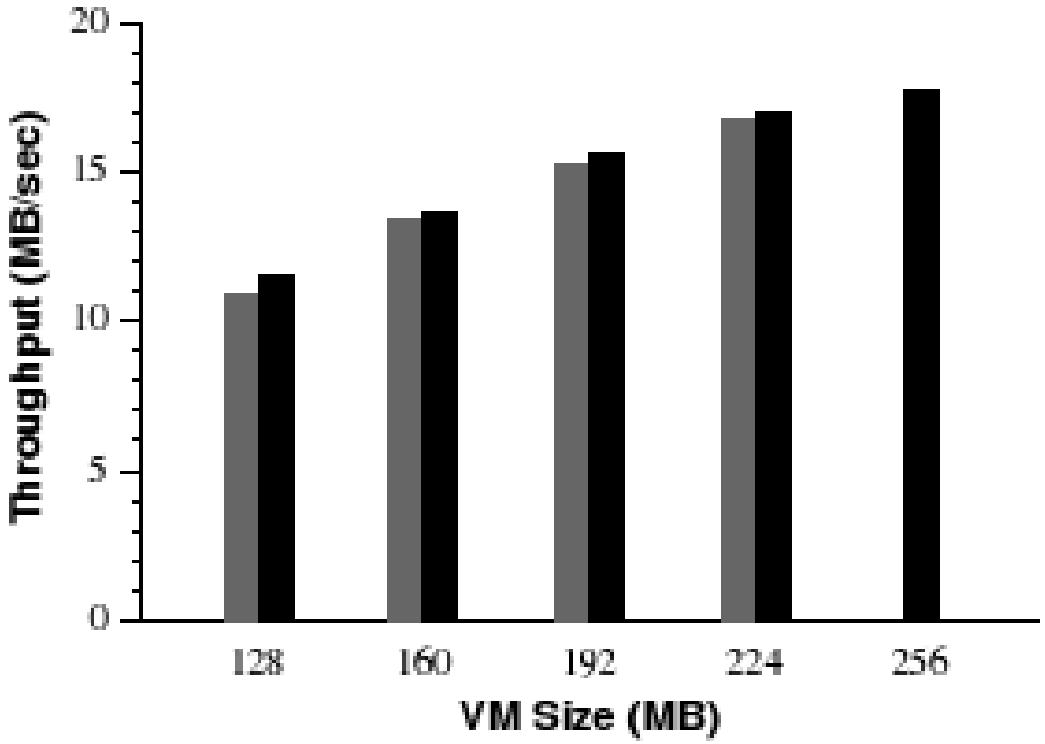
Ballooning

- avoid double paging
- minimal driver
- no external interface within the guest



Reclamation Mechanism (2)

Example



Share Adaption (1)

Calculation of the Shares-per-page-ratio

- idle memory tax $\tau : 0 \leq \tau < 1$
- idle page cost $k : k = 1/(1-\tau)$
- active fraction of pages f
- share S

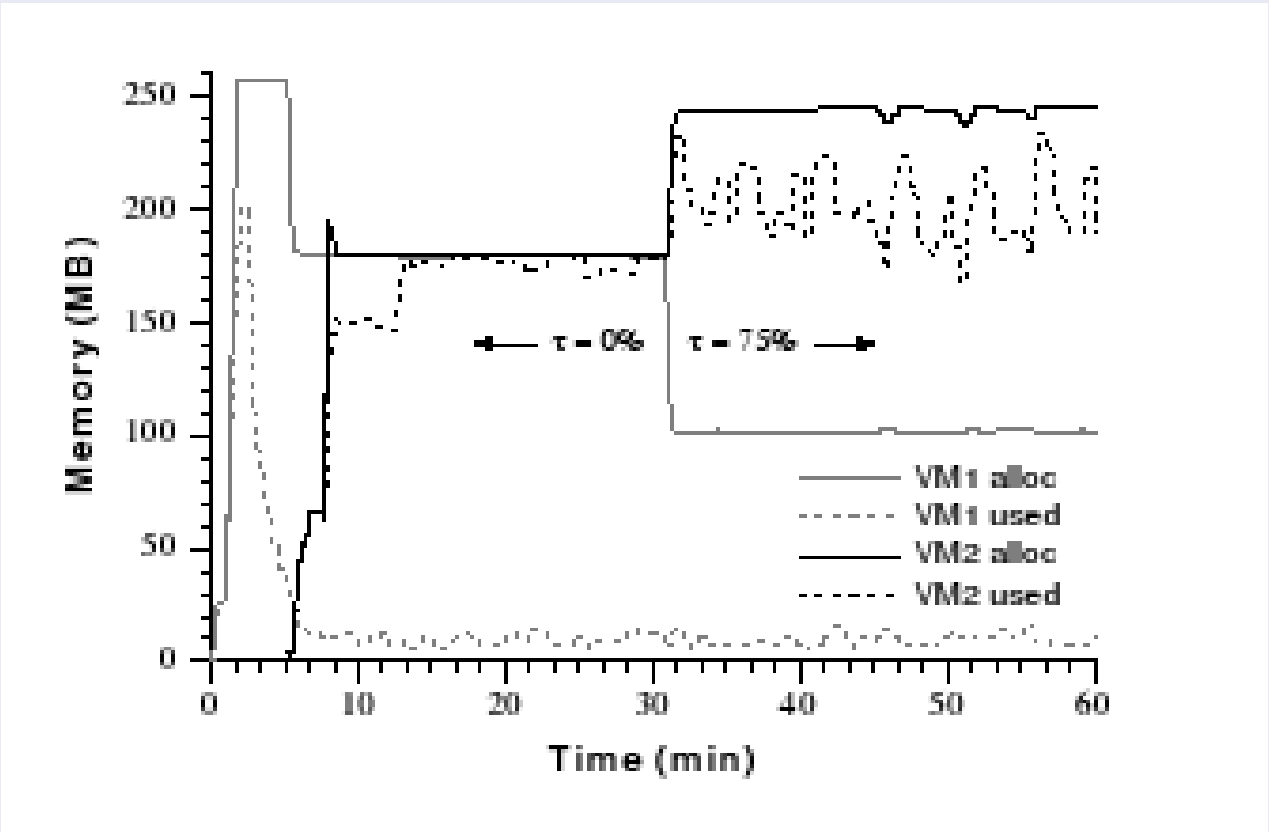
$$\rho = \frac{S}{P*(f+k*(1-f))}$$

Interpretation of the idle memory tax

- $\tau = 0 \rightarrow$ pure share based policy
- $\tau \approx 1 \rightarrow$ reclaim all idle memory

Share Adaption (2)

Example



Dynamic Reallocation

State transitions

- addition or removal of a vm
- change of allocation parameter
- periodic rebalancing

States of memory pressure

- *high(6%)* - no reclamation
- *soft(4%)* - ballooning
- *hard(2%)* - forcible paging
- *low(1%)* - additionally blocks execution

Conclusions

Questions

- What about a compromized ballon driver?
- Is context-based memory sharing really efficient? What about guest hints?
- How to specify a useful value for idle memory tax?