Memory Resource Management in VMware ESX Server

Carl A. Waldspurger

OSDI' 02

Presenter: Jin Zhang

2021/2/27

Brief for Memory Management of Virtualization

- Assume a VM with 4GB memory:

- Step 1:
  - `void* ptr = malloc(4GB)`
- Step 2:
  - Setup SPT/EPT
- Step 3:
  - A number of corner cases, such as MMIO (memory-mapped I/O), SMM (system management mode), MTRR (memory type range register), etc.

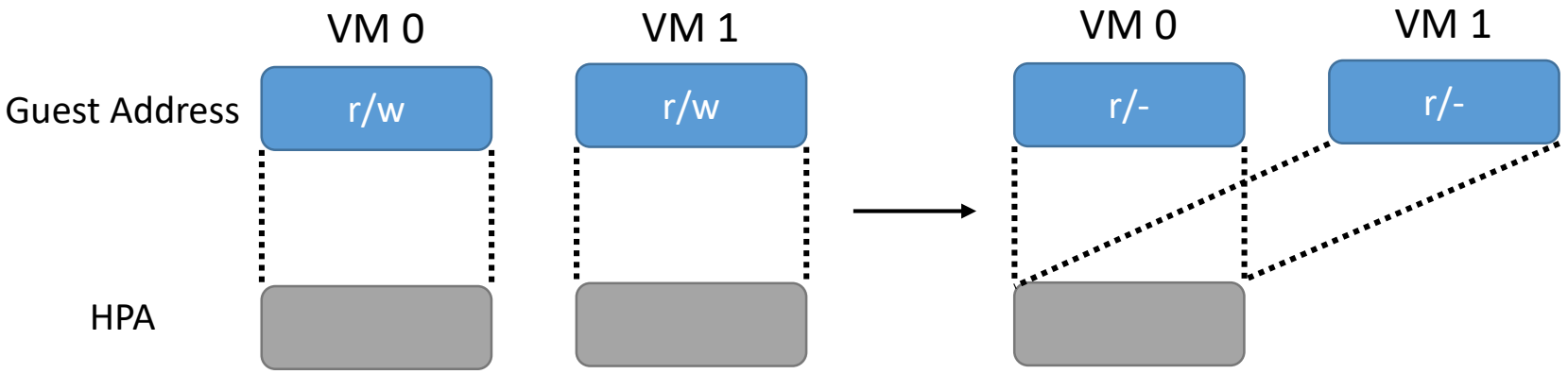
Outline

- **Introduction**

- Memory Deduplication
- Page Reclamation
- Allocation Policy
- I/O Page Remapping
- Summary

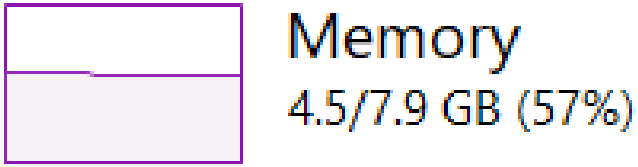
Memory Deduplication

- It is possible that multiple VMs own the pages with the same content.
- We can map two different virtual addresses to the same physical address.

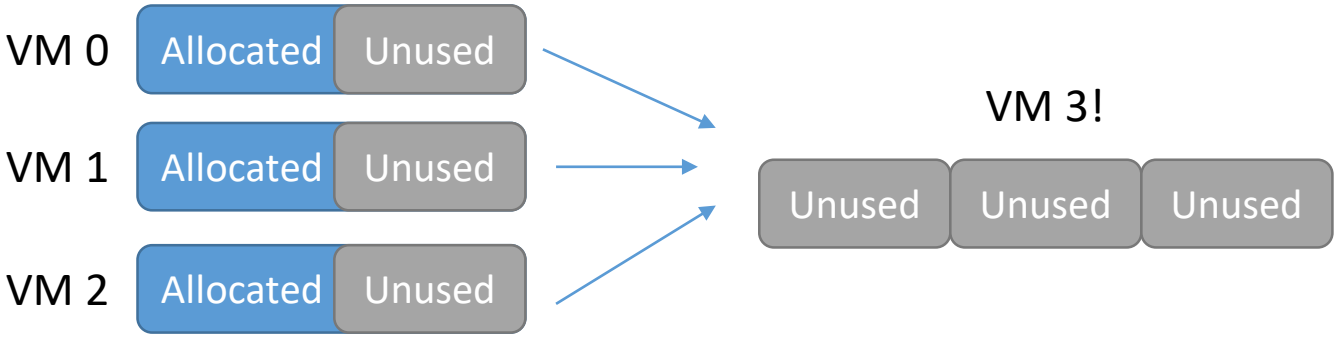


Memory Overcommitment

- In most cases, one doesn't run out of its memory.



- For a physical machine hosting multiple virtual machines, the accumulated unused memory might be significant.



- Swap to the disk

Business Concern in Data Centers (Including Cloud)

- Users: I want to maximize my applications' performance.

(I want to acquire all the *promised* physical resources)

- Cloud vendors: I want to maximize my profits.

(I want to gather the unused resources to sell more resources)

Outline

- Introduction
- **Memory Deduplication**

- Page Reclamation
- Allocation Policy
- I/O Page Remapping
- Summary

Transparent Page Sharing

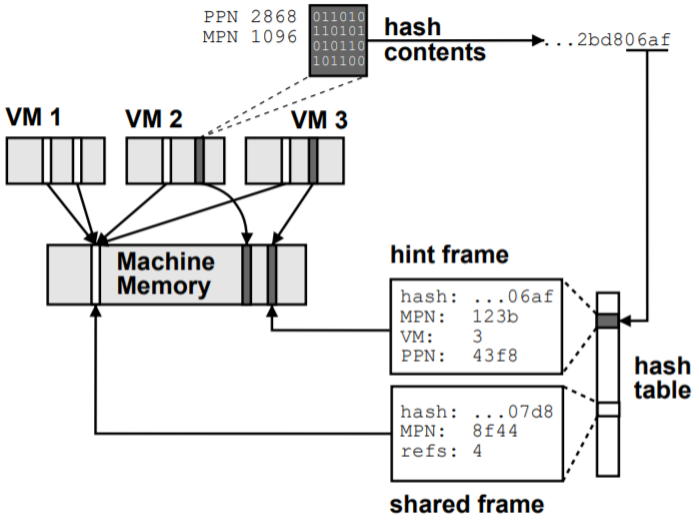
General idea: Pages may be the same. The same pages can be merged/shared.

- Problem: how to find the same pages?
- Technical idea I: capture all the write to the guest OS
  - All PTEs can be set as read-only. It's too slow.
- Technical idea II: compare the content of one page to all the others
  - The time complexity is $O(N^2)$, where N is the number of pages.

Transparent Page Sharing

Sol.

- The hash values of scanned pages are generated.
- The hash value of an incoming page is compared to the hash values in a global table.
  - Upon a hash match, both PTEs are set as COW.
- What if a mismatch?
  - The page is marked with a hint.
  - Next time the page will be rehashed.



Kernel Shared/Samepage Memory

- Courtesy by Redhat. KSM has been integrated to Linux kernel.
- Use: `madvise(VM_MERGEABLE)`
- Each page's hash value is calculated by `jhash2`.
- Two rbtrees:
  - Stable tree: pages that have been merged and their content won't be modified.
  - Unstable tree: pages that are not merged.
  - Time complexity: $O(N\log N)$, where N is the number of pages.

Kernel Shared/Samepage Memory

- `ksmd` consists of *pass*es.
- For each pass:
  - Whether there is a matched page in the stable tree?
  - Does hash value change?
    - If so, this page might be updated frequently and should ignore at least for this pass.
  - Does there is a matched page in the unstable tree?
    - If so, the page's hash value should be recalculated since it's "unstable".

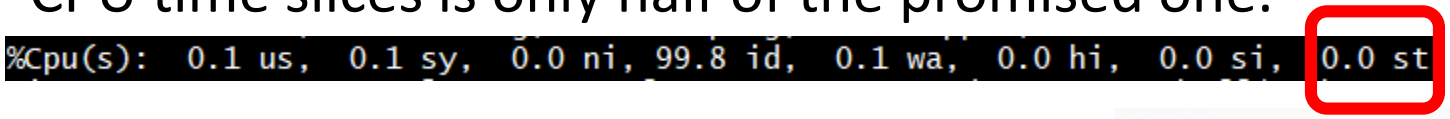
Outline

- Introduction
- Memory Deduplication
- **Page Reclamation**

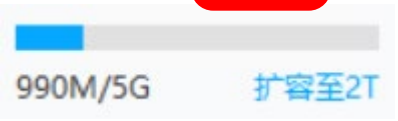
- Allocation Policy
- I/O Page Remapping
- Summary

Overcommitment

- Overcommitment is quite common in computer system, not limited to memory or virtualization.
  - E.g., a VM is allocated with 8GB memory. The actual received memory may be only 6GB.
  - A VM is guaranteed with 1 vCPU. The actual received CPU time slices is only half of the promised one.



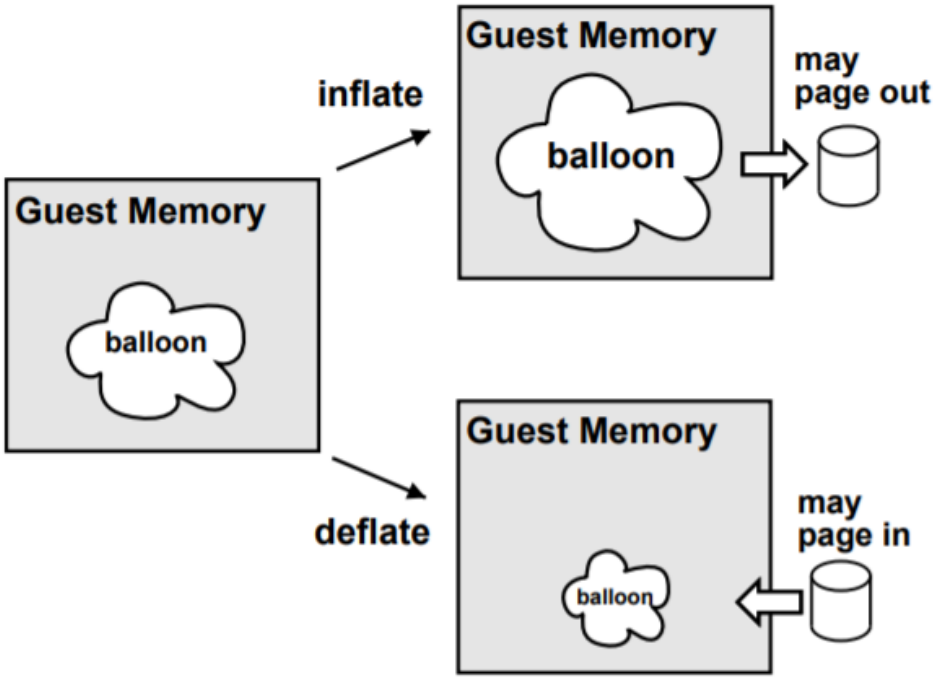
  - Baidu Cloud Disk claims a 5GB free space.



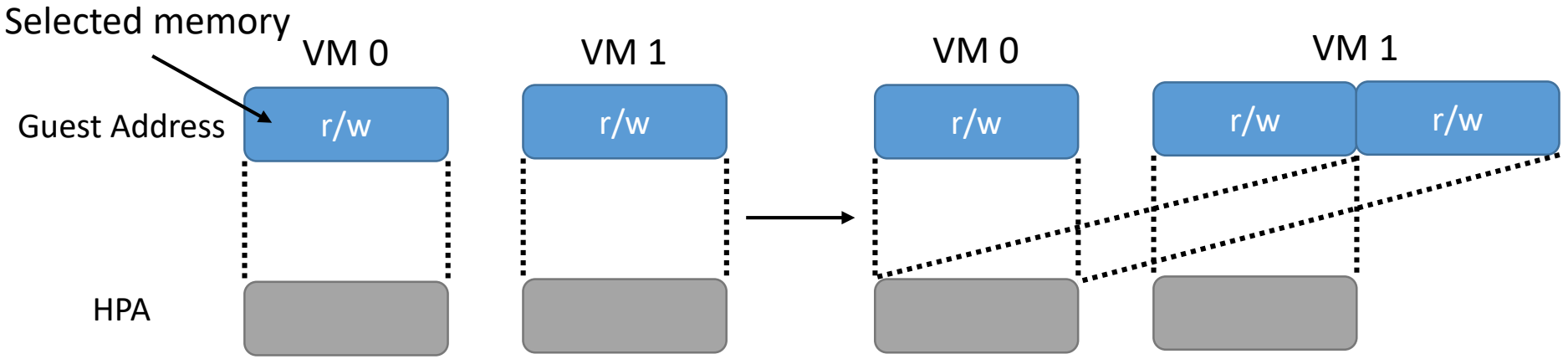
- What if the actual memory use of a VM approaches the promised one?
  - Memory should be reclaimed from other VMs to this one.

Ballooning

- Balloon is a *guest-aware* device driver that can take additional memory from one guest to another.

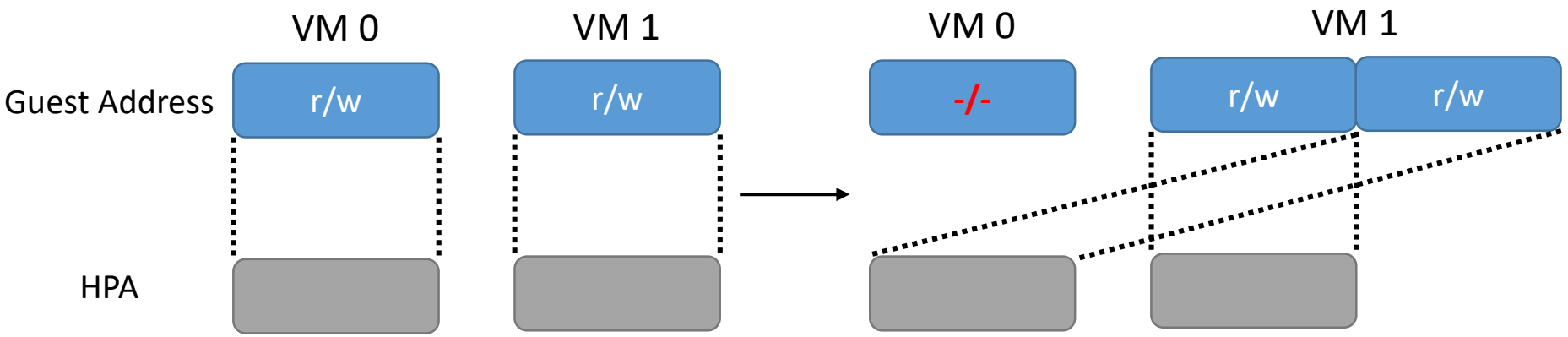


A Security Concern for Ballooning



- Oops! VM 0 now can access VM 1's memory, and vice versa.

A Security Concern for Ballooning



- Oops! VM 0 now can access VM 1's memory, and vice versa.
- Solution. PTEs of ballooned memory are set as -/-.

Swapping

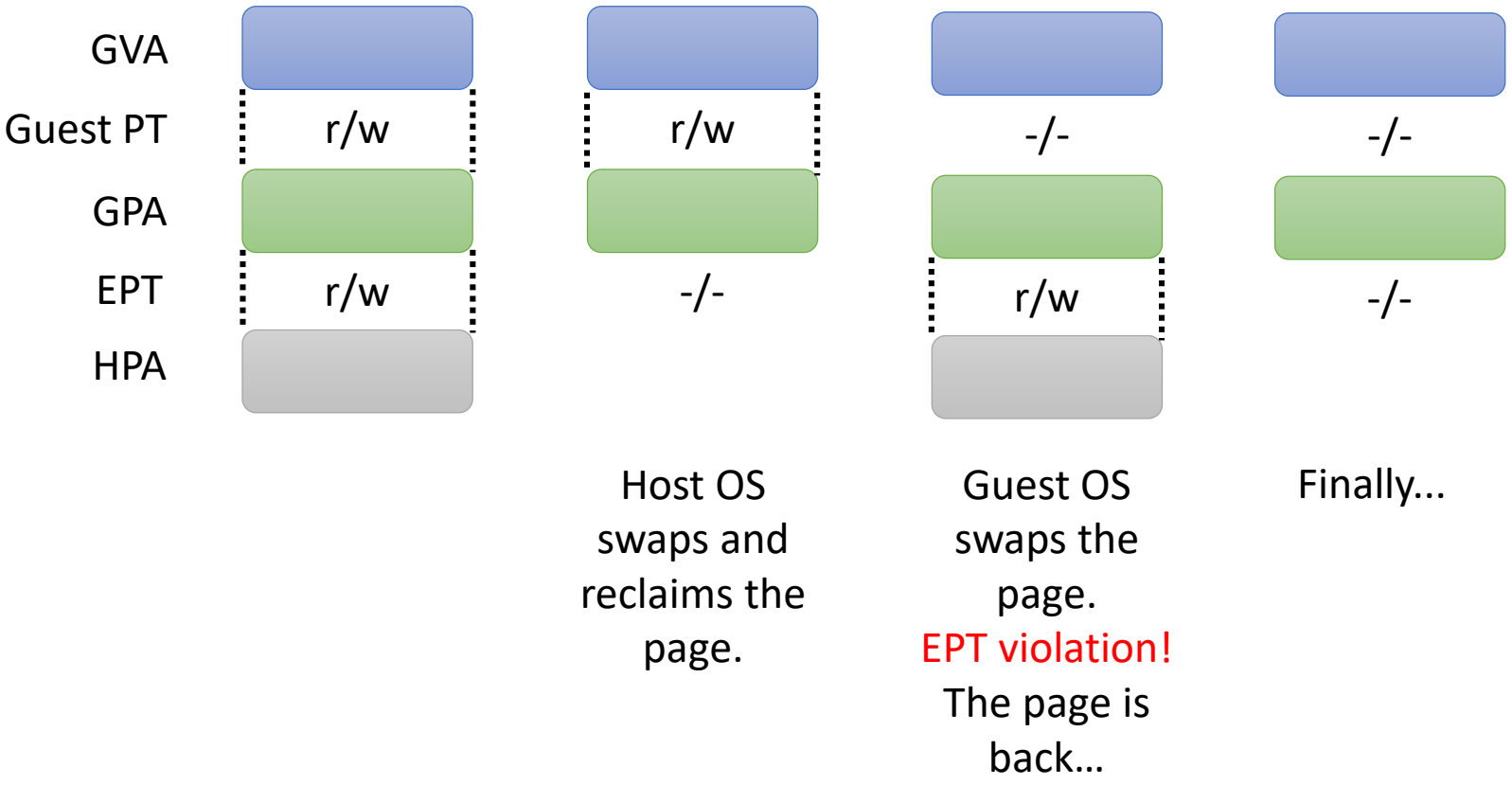
- *Swapping* means paging to the disk.
  - Nowadays it becomes popular to swap pages to a remote DRAM connected via RDMA.
- The last resort in case of insufficient ballooning.
- Which pages are candidates for swapping?
  - RANDOM ones.

Why VMWare Chooses such a Trivial Swapping Method?

- The conventional swapping method (e.g., `swapd`) is based on identifying hot/cold pages.
  - Hot pages in DRAM; cold pages on disk

- Such information is gathered in the guest OS, modification to guest OS is thus required, which is troublesome.
  - So ballooning is not troublesome.

Double Paging Problem



Evaluation

		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

A: Oracle, SQL Server, IIS, Websphere, Java, VB

B: Apache, Majordomo, Postfix, POP/IMAP, MailArmor

C: Squid, Postfix, RAV, ssh

Outline

- Introduction
- Memory Deduplication
- Page Reclamation
- **Allocation Policy**

- I/O Page Remapping
- Summary

Memory Allocation Policy

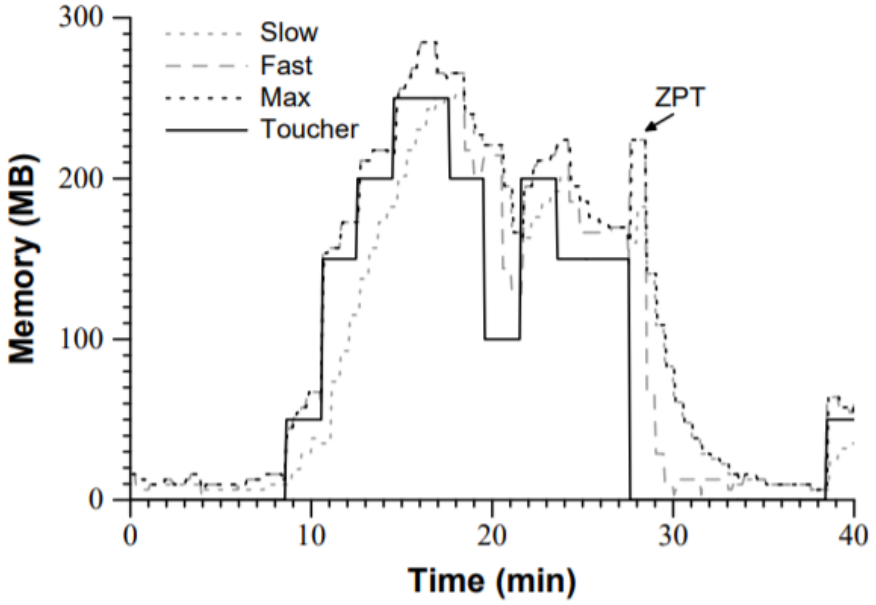
- A classic problem: how much memory should be allocated to a client (thread, process, container, VM, etc.)?

- A client is assigned with *shares*.
  - Priority?

- Problem: an idle client with many shares waste the memory.
  - Idle pages should contribute negative shares-idle memory tax

How to Detect Idle Pages?

- Solution I: info provided by the guest OS
  - Cons: per-process metrics; access bits are bypassed by DMA accesses
- Solution II: guest-agnostic sampling
  - Random pages are set as read-only by SPTe/EPTe.
  - Record all guest accesses during a period.
  - The period can be long(*slow*)/medium(*fast*)/short (*max*)

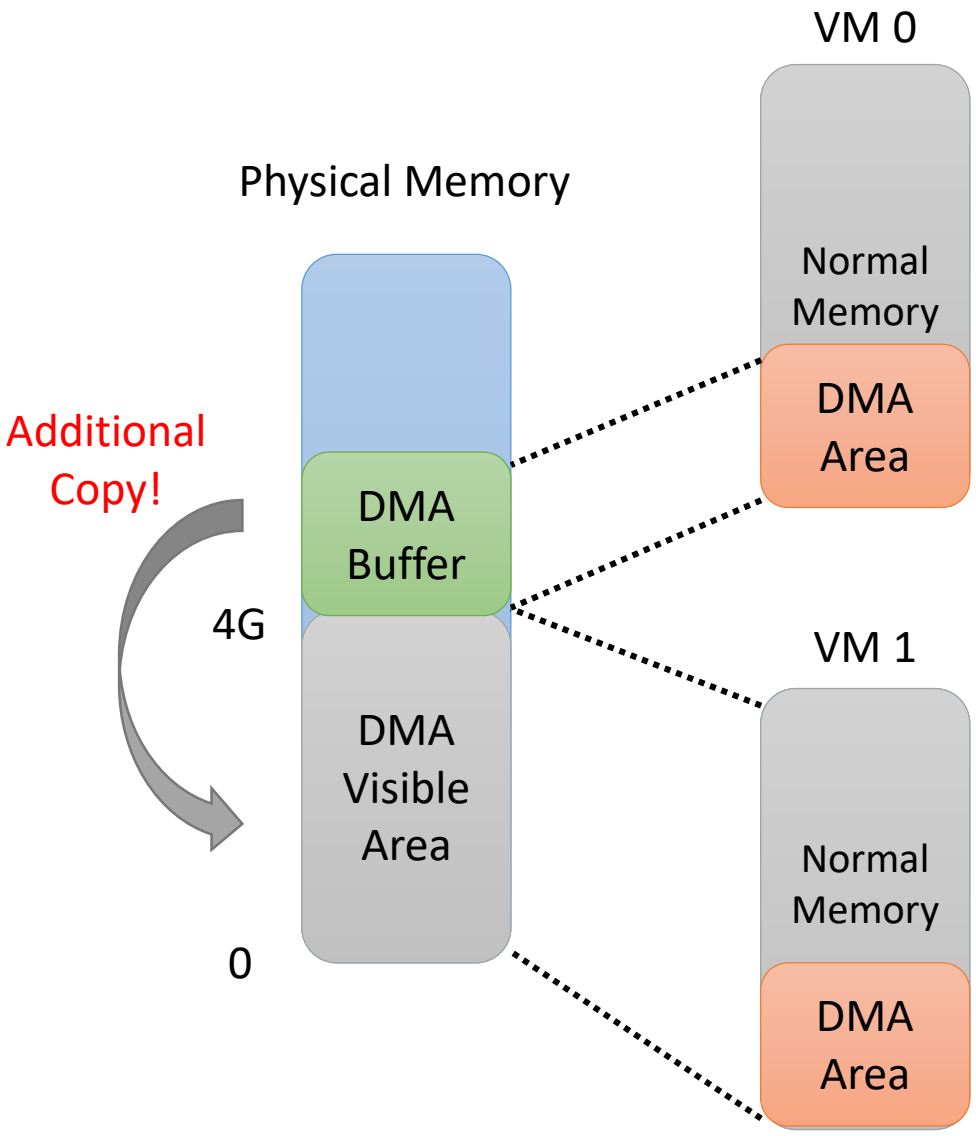


Outline

- Introduction
- Memory Deduplication
- Page Reclamation
- Allocation Policy
- **I/O Page Remapping**

- Summary

I/O Page Remapping



- Devices supporting 32-bit addressing can only access [0, 4GB] memory space (low memory).
- Low memory may run out, which causes additional copies.
- Sol: hot pages in low memory; cold pages in high memory.

Outline

- Introduction
- Memory Deduplication
- Page Reclamation
- Allocation Policy
- I/O Page Remapping
- **Summary**

Summary

- Several techniques to improve the memory utilization (profit):
  - Transparent Page Sharing
  - Ballooning
  - Swapping
  - Allocation Policy
  - I/O Page Remapping

Q & A

Live Migration of Virtual Machines

Christopher Clark, Keir Fraser, Steven Hand

NSDI ‘05

Jin Zhang

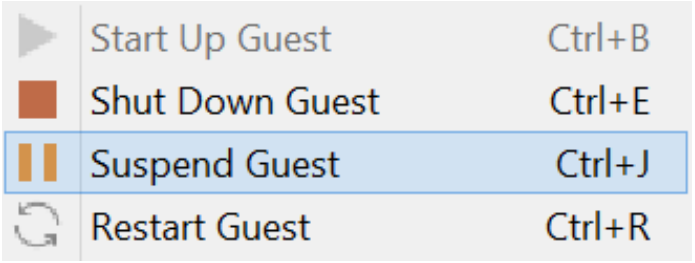
2021/2/27

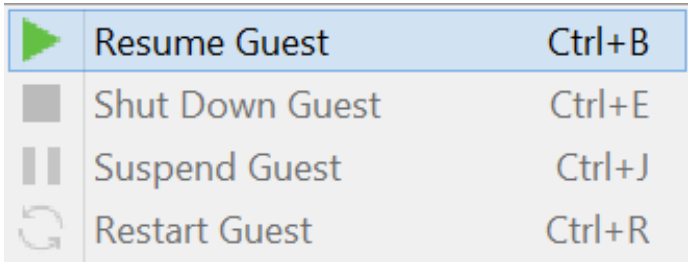
A Common Scenario: Change HDD to SSD in a VPS

- Solution I: Wild management
  - Install SSD by hand
  - Cons: People can make mistake

A Common Scenario: Change HDD to SSD in a VPS

- Solution I: Wild management
  - Install SSD by hand
  - Cons: People can make mistake
- Solution II: Suspend and resume
  - Suspend to get an image of current VM, copy it to the destination (with SSD) and resume execution





A Common Scenario: Change HDD to SSD in a VPS

- Solution I: Wild management
  - Install SSD by hand
  - Cons: People can make mistake
- Solution II: Suspend and resume
  - Suspend to get an image of current VM, copy it to the destination (with SSD) and resume execution
  - Cons: The applications stop for a long time

A Common Scenario: Change HDD to SSD in a VPS

- Solution I: Wild management
  - Install SSD by hand
  - Cons: People can make mistake
- Solution II: Suspend and resume
  - Suspend to get an image of current VM, copy it to the destination (with SSD) and resume execution
  - Cons: The applications stop for a long time
- Solution III: Live migration
  - “Live” migrate VM to destination, without stopping the applications

Outline

- **Introduction**

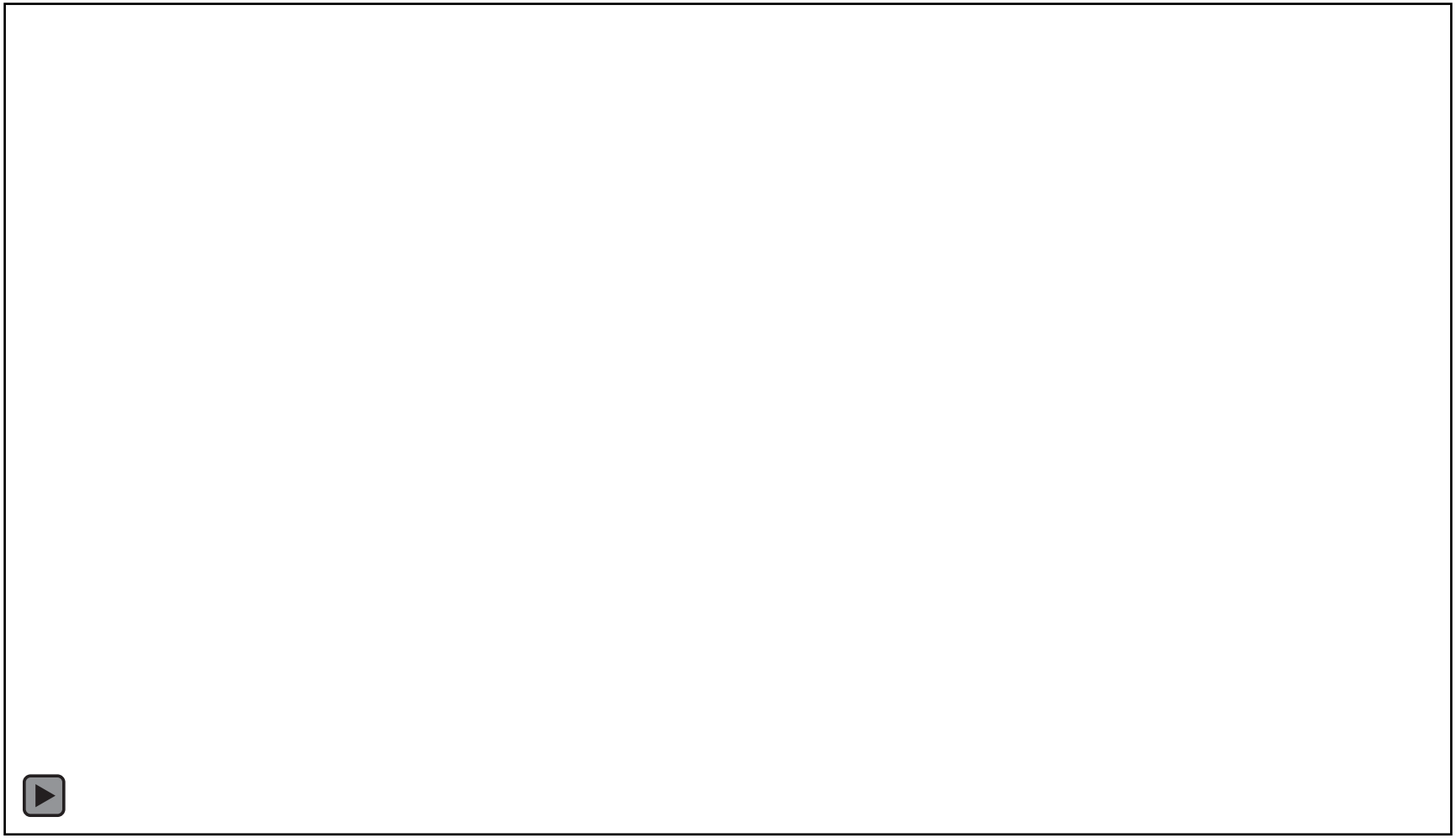
- Design
- Implementation
- Optimization
- Evaluation
- Summary

Introduction

What is live migration?

The process of moving a running virtual machine or application between different physical machines without disconnecting the client or application.

Video Demo



Introduction

- What is live migration?

The process of moving a running virtual machine or application between different physical machines without disconnecting the client or application.

- Consideration
  - CPU, I/O
    - Easy to handle, because of little state data
  - Memory
    - The key point. 512MB in this paper, up to hundred GBs nowadays

Outline

- Introduction
- **Design**

- Implementation
- Optimization
- Evaluation
- Summary

Memory Migration

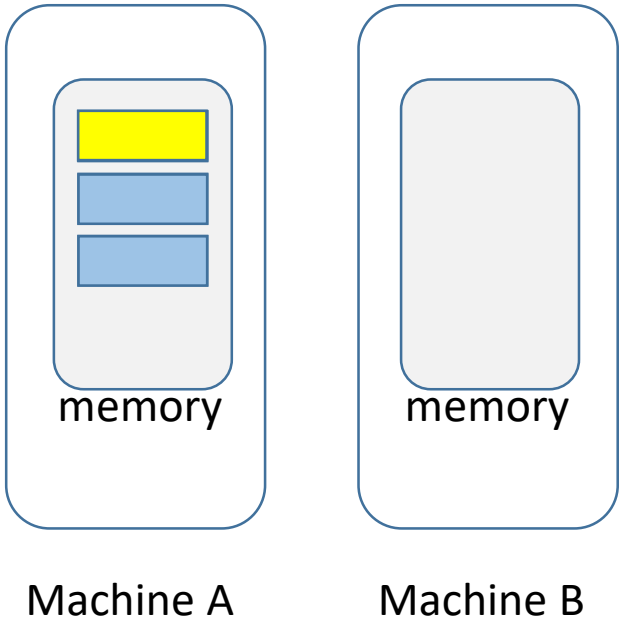
Three methods when migrating memory from machine A to machine B:

- Pre-copy phase
- Stop-and-copy phase
- Post-copy phase

Memory Migration

Three methods when migrating memory from machine A to machine B:

- **Pre-copy phase**
  - A pushes pages to B
  - Resend dirty pages
  - Many iterations
- Stop-and-copy phase
- Post-copy phase



How to Identify Dirty Pages?

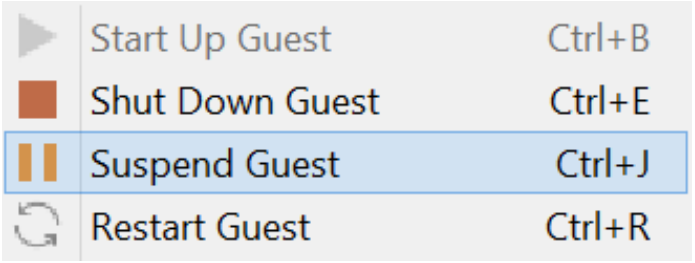
- Software-based (Legacy) approach:
  - Set page table as write-protected
  - The following write #PF can record the dirtied pages.
  - Cons: one trap for each #PF.

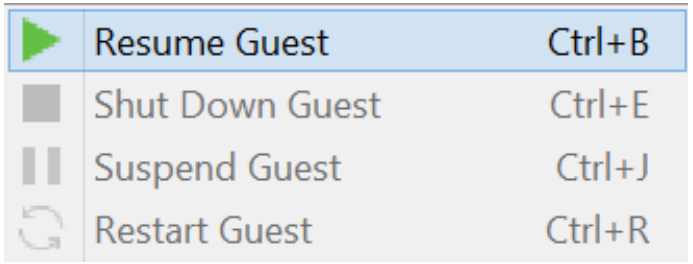
- Hardware-based approach-Page Modification Logging:
  - Dirtied pages are batched.
  - Batched dirtied pages are notified to the hypervisor via one VMExit.

Memory Migration

Three methods when migrating memory from machine A to machine B:

- Pre-copy phase
- **Stop-and-copy phase**
  - Looks like suspend-resume way
  - Usually cooperate with other two ways
- Post-copy phase

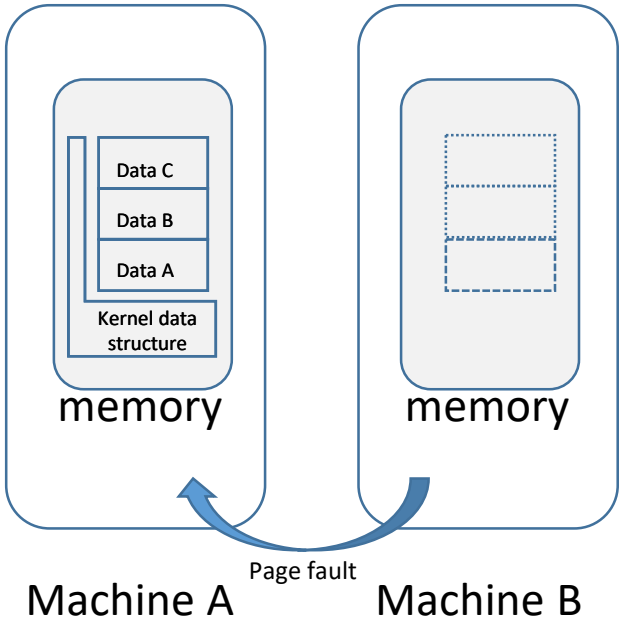




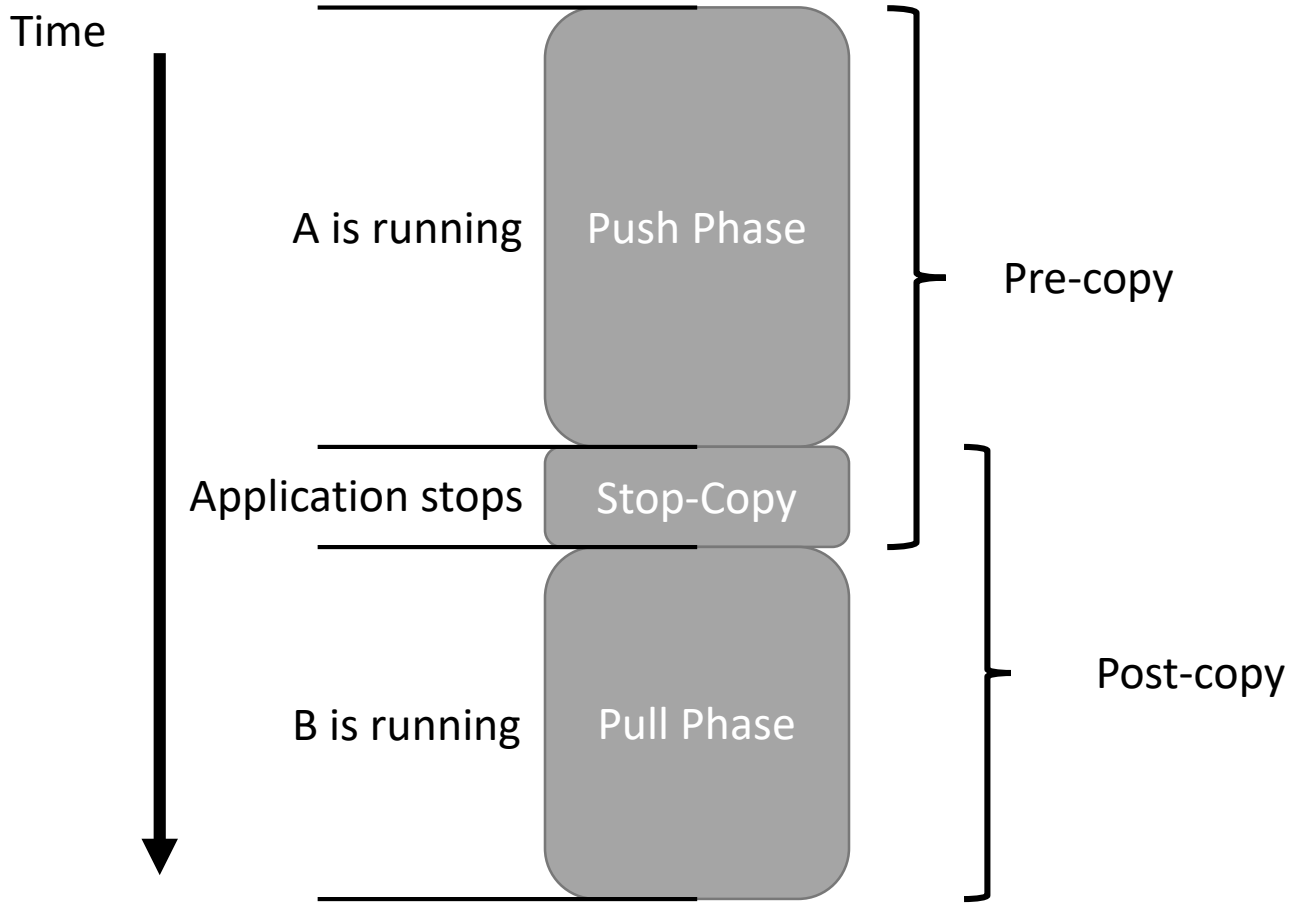
Memory Migration

Three methods when migrating memory from machine A to machine B:

- Pre-copy phase
- Stop-and-copy phase
- **Post-copy phase**
  - Transfer necessary data to B and let B run
  - B will incur page fault, and hypervisor pull pages from A to fix it



Summary of Memory Migration



Outline

- Introduction
- Design
- **Implementation**

- Optimization
- Evaluation
- Summary

Writable Working Set

- Some pages trend to be updated frequently, while others don't
  - A simple lemma of locality
- The latter is fine candidates for pre-copy, the former is tracked by ***Writeable Working Set*** (WWS)

- Total migration time of pre-copy:

$$\frac{RAM\ Size}{Network\ Speed} + alpha$$

alpha depends on memory update speed

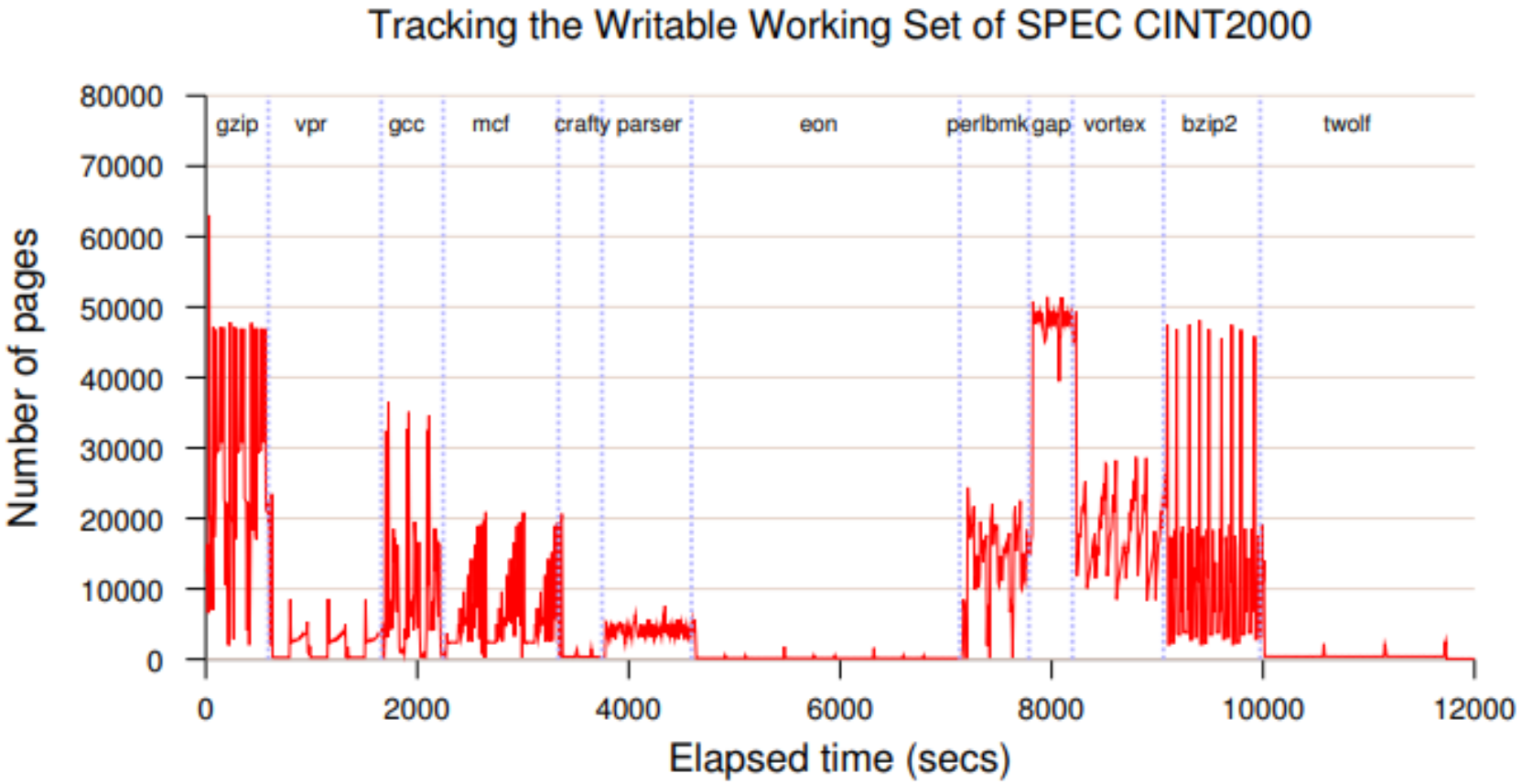


Figure 2: WWS curve for a complete run of SPEC CINT2000 (512MB VM)

CPU and I/O

- CPU
  - In the view of a hypervisor, a vCPU is a `struct` consisting of lots of registers
  - Stop VM -> Read status from vCPU -> Package and send to destination
- Disk
  - NAS (Network Attached Storage), RAID-1, iSCSI…
  - Migrate before live migration…
- Network
  - ARP (Address Resolution Protocol) maps IP to MAC
  - Send faked ARP packets (A's IP -> B's MAC)
  - Can ARP spoofing really work?

Outline

- Introduction
- Design
- Implementation
- **Optimization**

- Evaluation
- Summary

Optimization

- Dynamic rate-limiting algorithm
  - A single network limit is inappropriate
  - Low limit
    - Good service performance, extended downtime caused by hot pages
  - High limit
    - Reduced downtime, additional network contention
  - Adapt bandwidth limit to the dirtying rate

Dirtying rate = nums of dirty pages / duration

Next round Limit = pre dirtying rate + a constant increment (50Mbit/s)

Optimization

- Dynamic rate-limiting algorithm

- Identify rapid page dirtying
  - It’s vain to transfer a page dirtied by next iteration
  - If a page is dirtied many times before, it’s probably dirtied again in this iteration
  - Don’t send that!

Optimization

- Dynamic rate-limiting algorithm
- Identify rapid page dirtying
- Paravirtualized optimization
  - Stunning rogue processes
    - Stop processes which updates memory frequently
    - It’s surely the application…
  - Freeing page cache pages
    - Guest OS knows about which pages are free
    - These pages need no migration

Outline

- Introduction
- Design
- Implementation
- Optimization
- **Evaluation**

- Summary

Key Points

- Downtime
  - Best downtime is zero
- Total migration time
  - Has a negative effect on service quality
- Service Quality
  - For example, throughput of a web server

A Simple Web Server

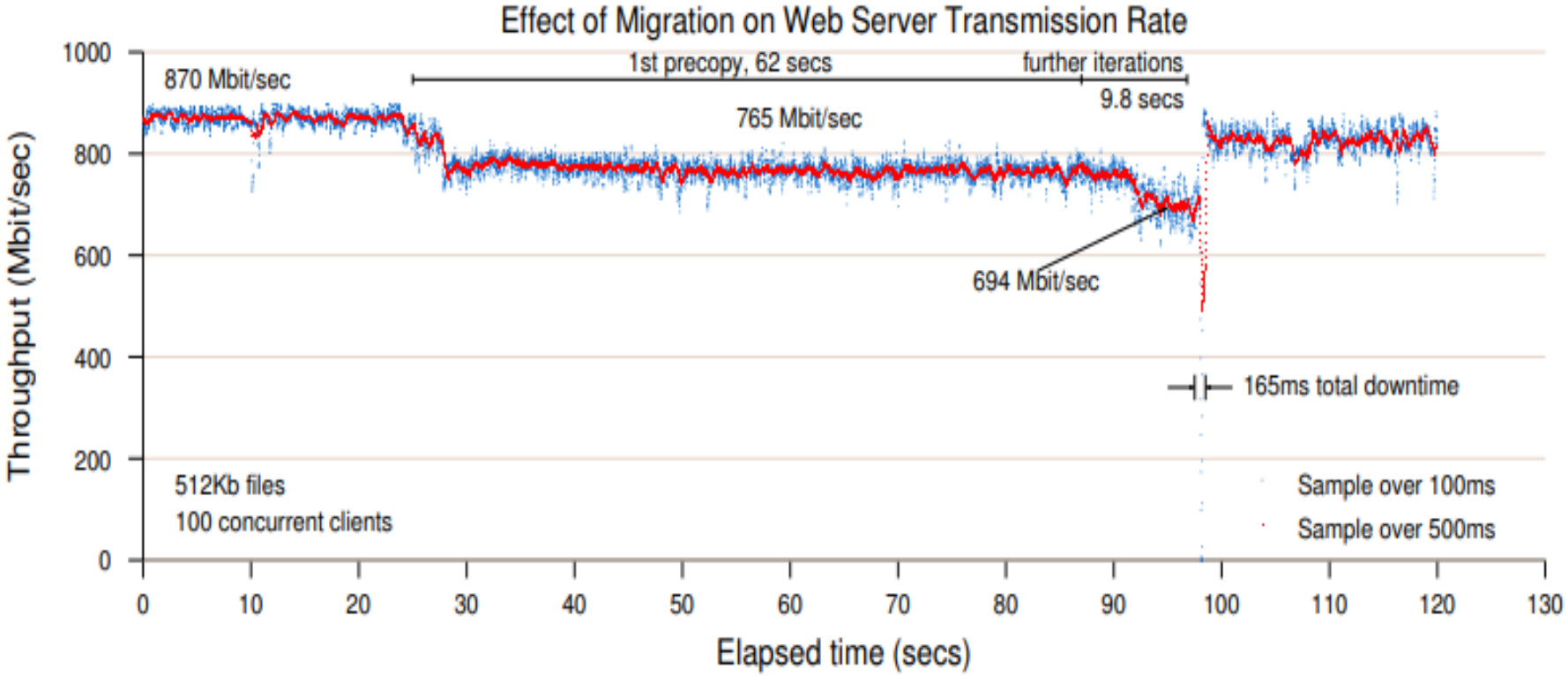


Figure 8: Results of migrating a running web server VM.

Quake 3 Server Migration

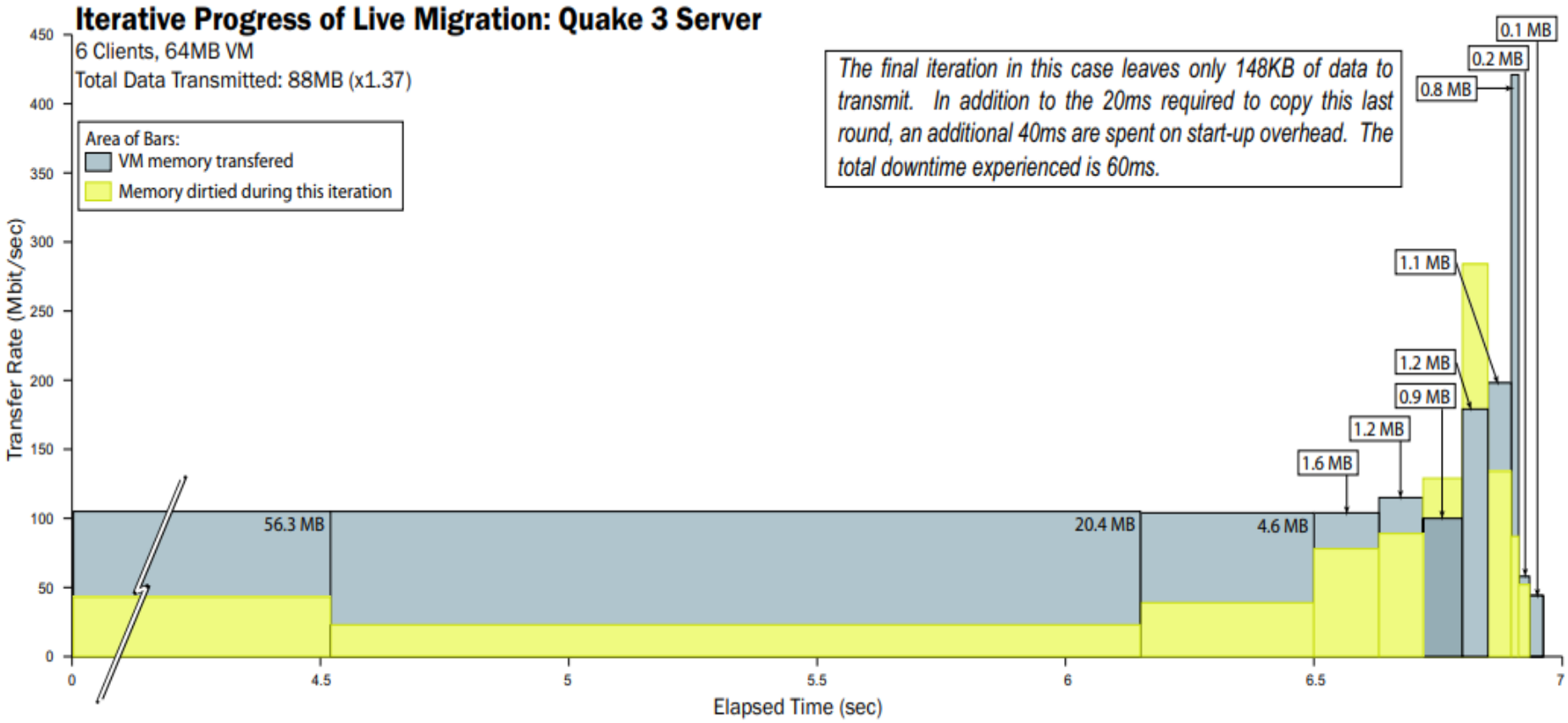


Figure 11: Results of migrating a running Quake 3 server VM.

Quake 3 Server Migration

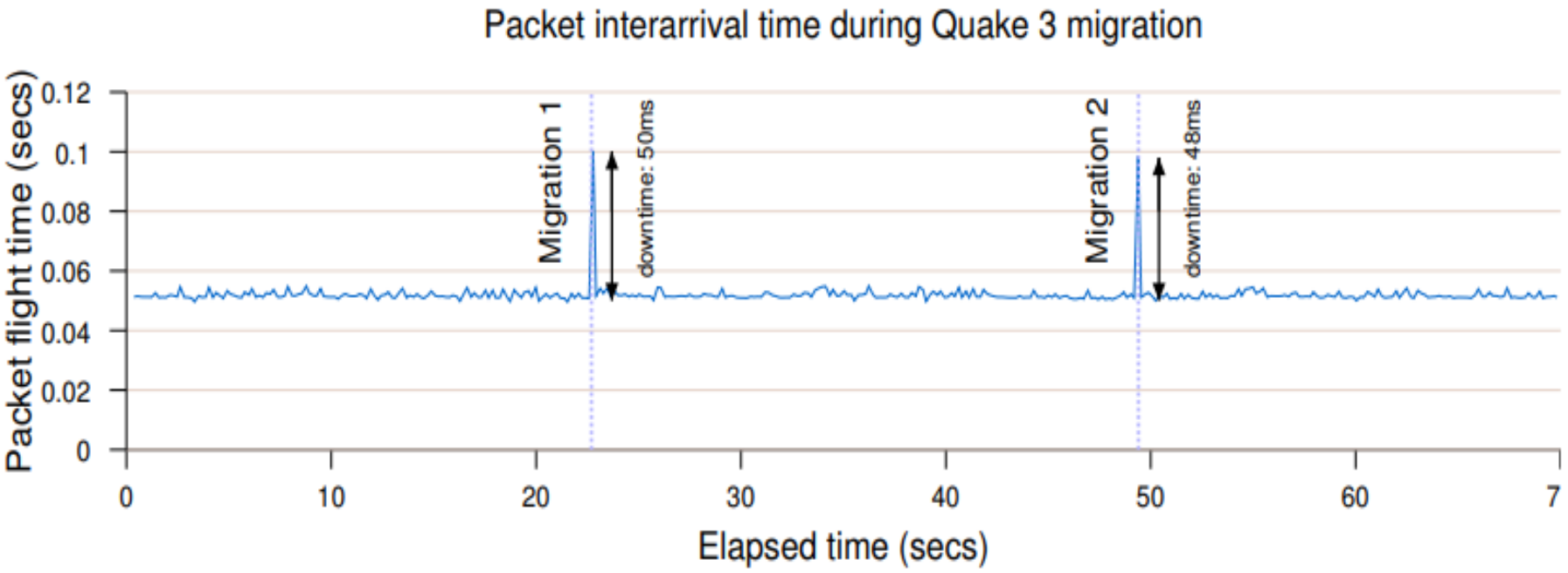


Figure 10: Effect on packet response time of migrating a running Quake 3 server VM.

Diabolical Workload

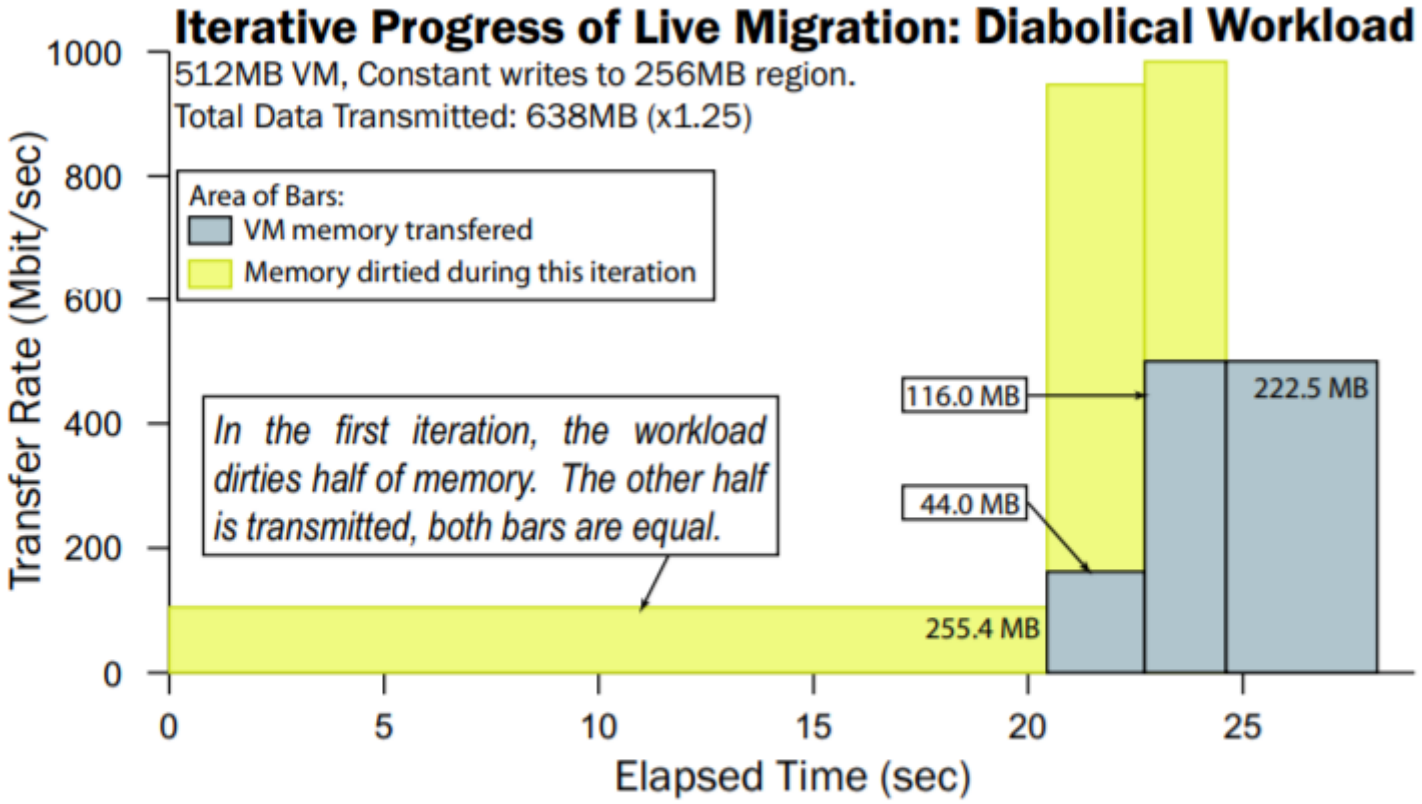


Figure 12: Results of migrating a VM running a diabolical workload.

Outline

- Introduction
- Design
- Implementation
- Optimization
- Evaluation
- **Summary**

Summary

- Bottleneck of live migration: memory
  - Pre-copy phase
  - Stop and copy
  - Post-copy phase
- Writable Working Set
- Optimization
  - Dynamic rate-limiting algorithm
- Evaluation
  - Downtime
  - Total migration time
  - Service quality

Q & A