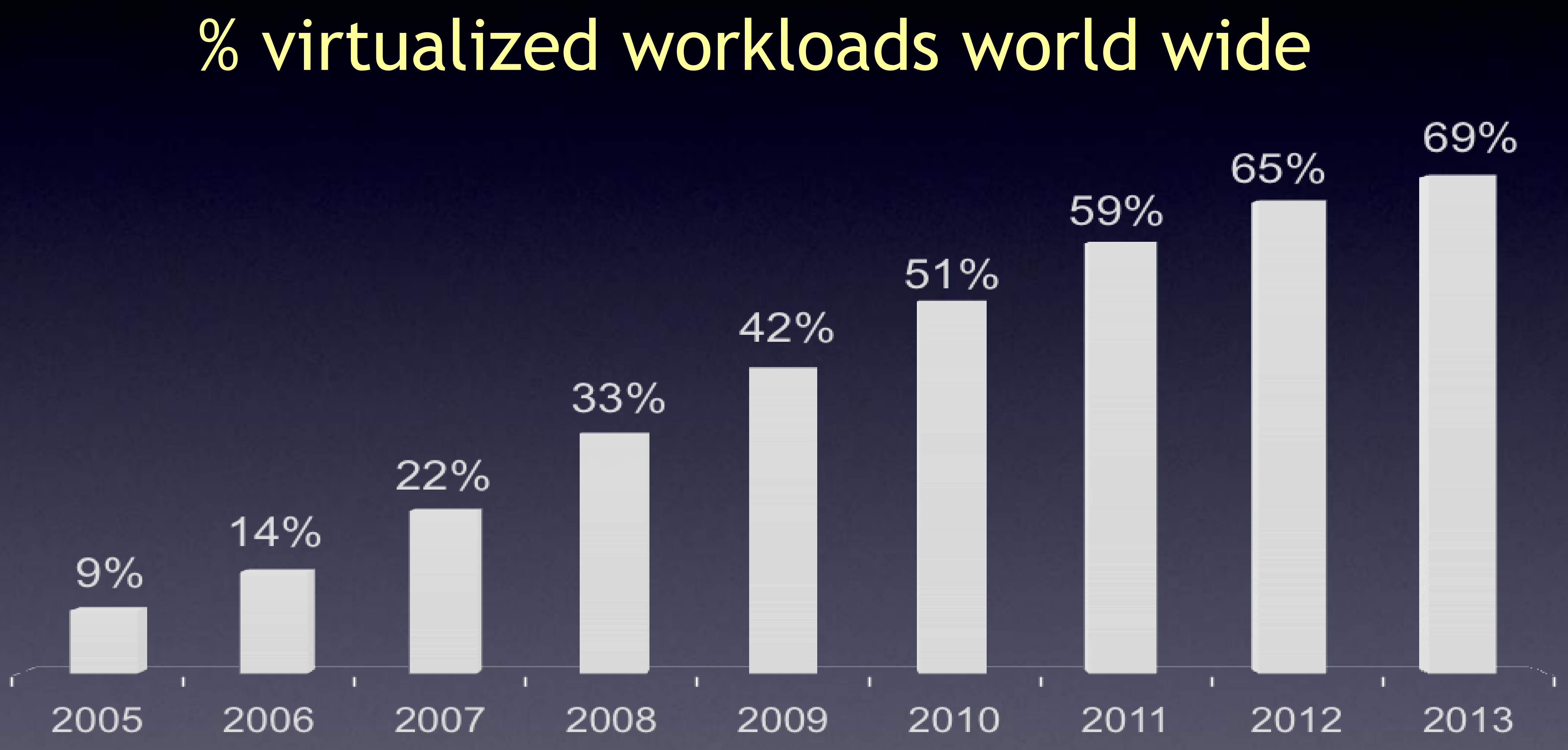
Software Techniques for Avoiding Hardware Virtualization Exits

Ole Agesen Jim Mattson Radu Rugina Jeffrey Sheldon

VMware

Server Virtualization

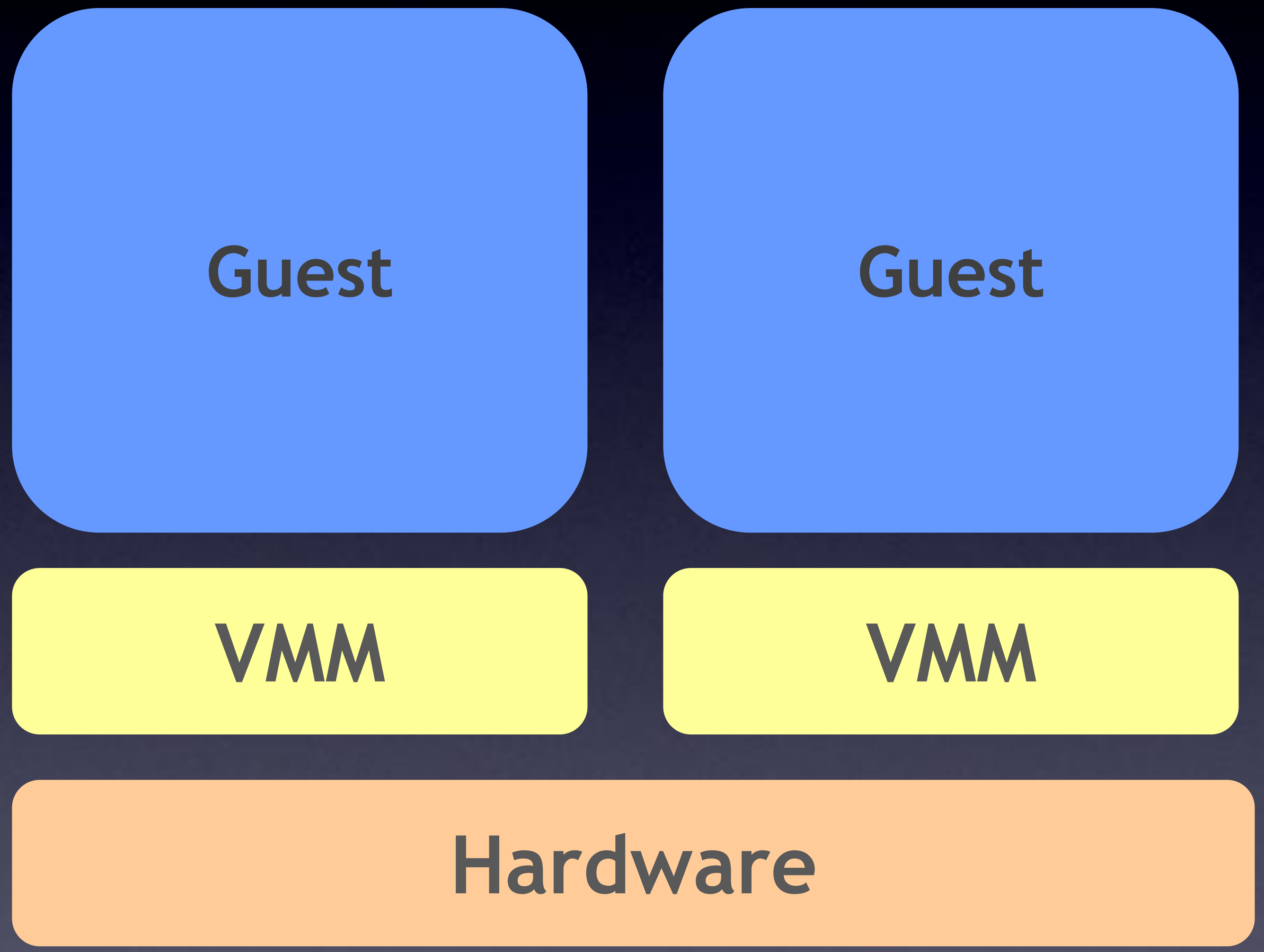


Source: IDC Worldwide Virtualization Tracker

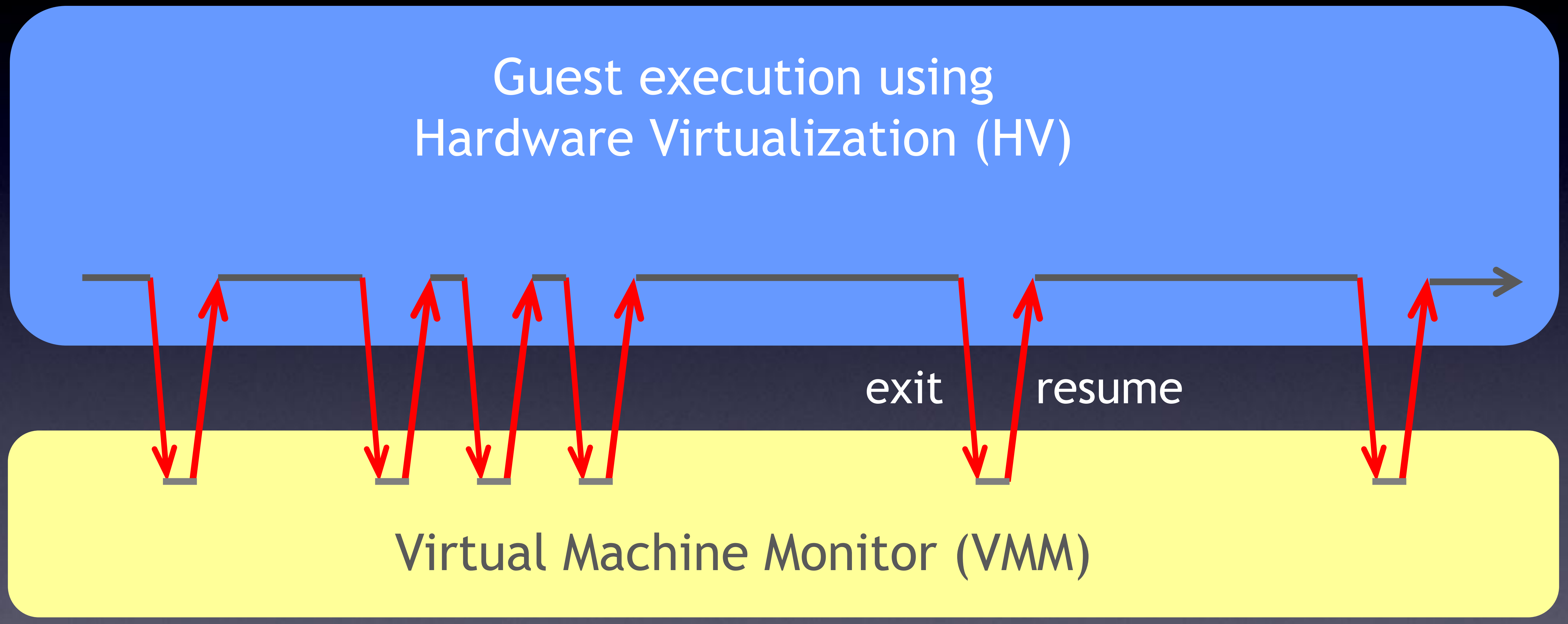
x86 Virtualization

- Virtual Machine Monitor (VMM) abstracts physical hardware
- Two approaches:
  - Binary Translation (BT)
  - Hardware-Assisted (HV)

    (Intel VT-x and AMD-V)



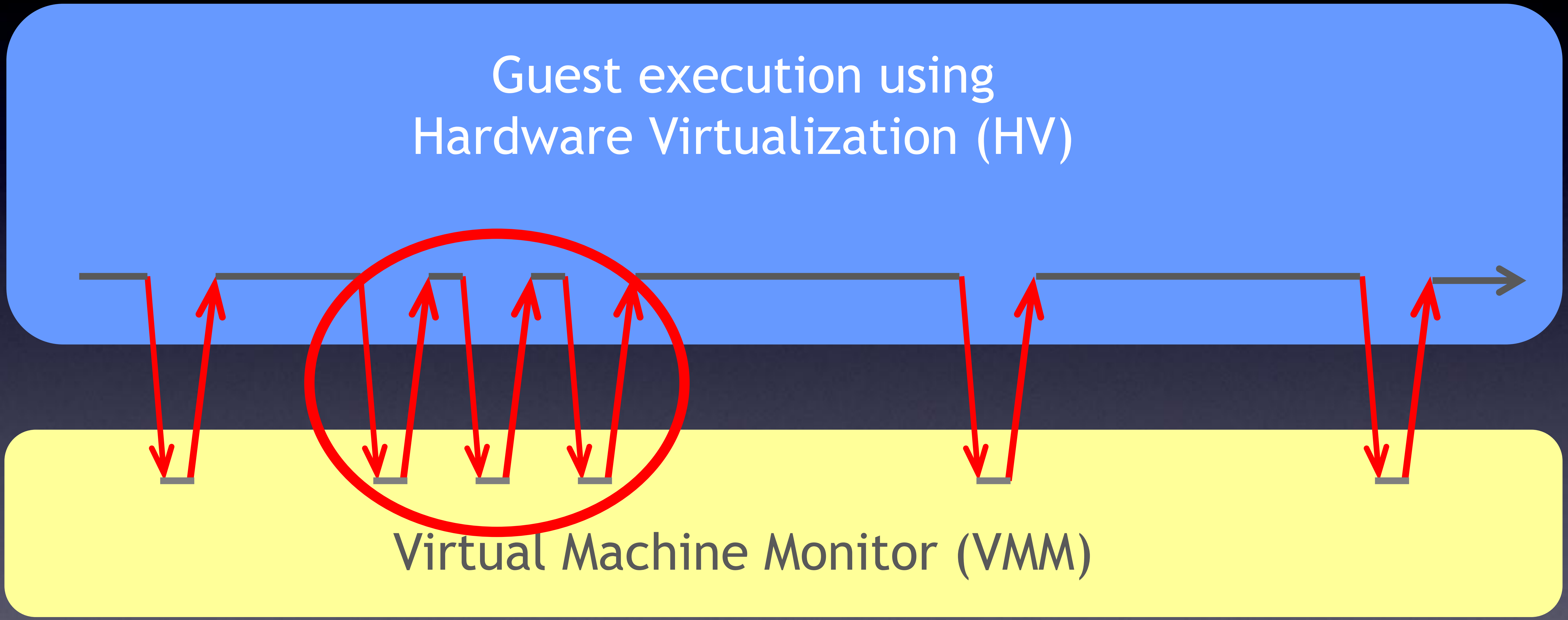
Hardware-Assisted x86 Virtualization



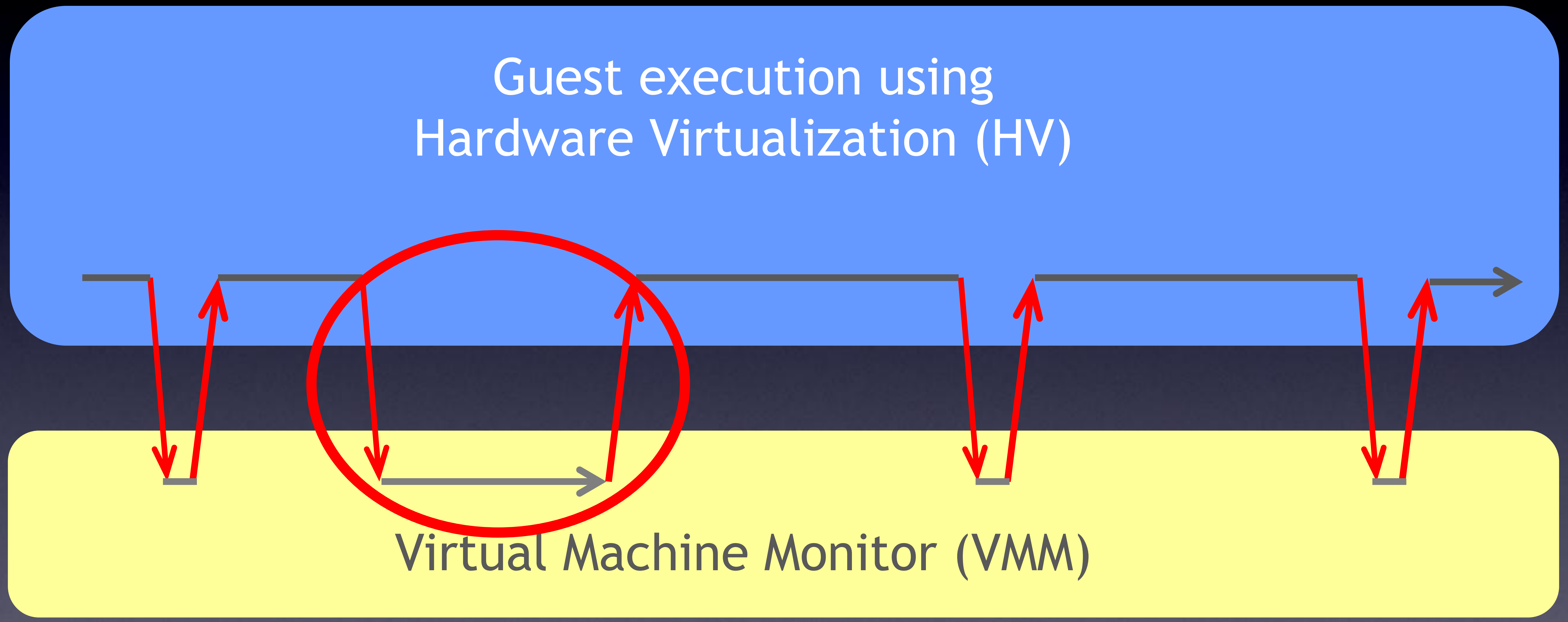
Virtualization Exits Are Expensive

Microarchitecture	Launch Date	Hardware Exit + Resume (cycles)
Prescott	2005	3963
Merom	2006	1579
Penryn	2008	1266
Nehalem	2009	1009
Westmere	2010	761
Sandy Bridge	2011	784

This Talk: Cluster HV Exits



This Talk: Cluster HV Exits



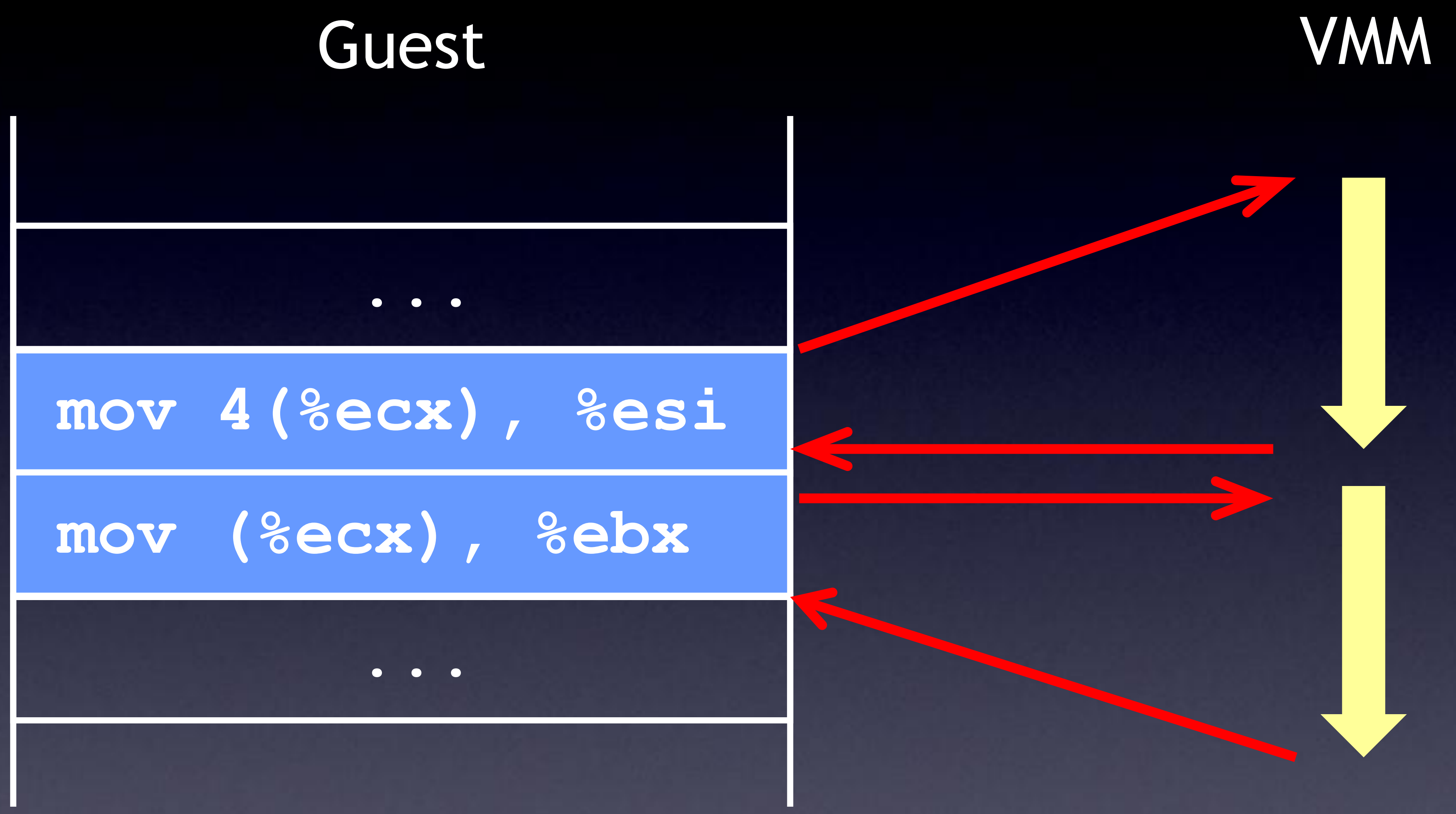
Outline

- Exit Pairs
- Exit Clusters
- Nested Virtual Machines
- Results
- Conclusions

Exit Pairs

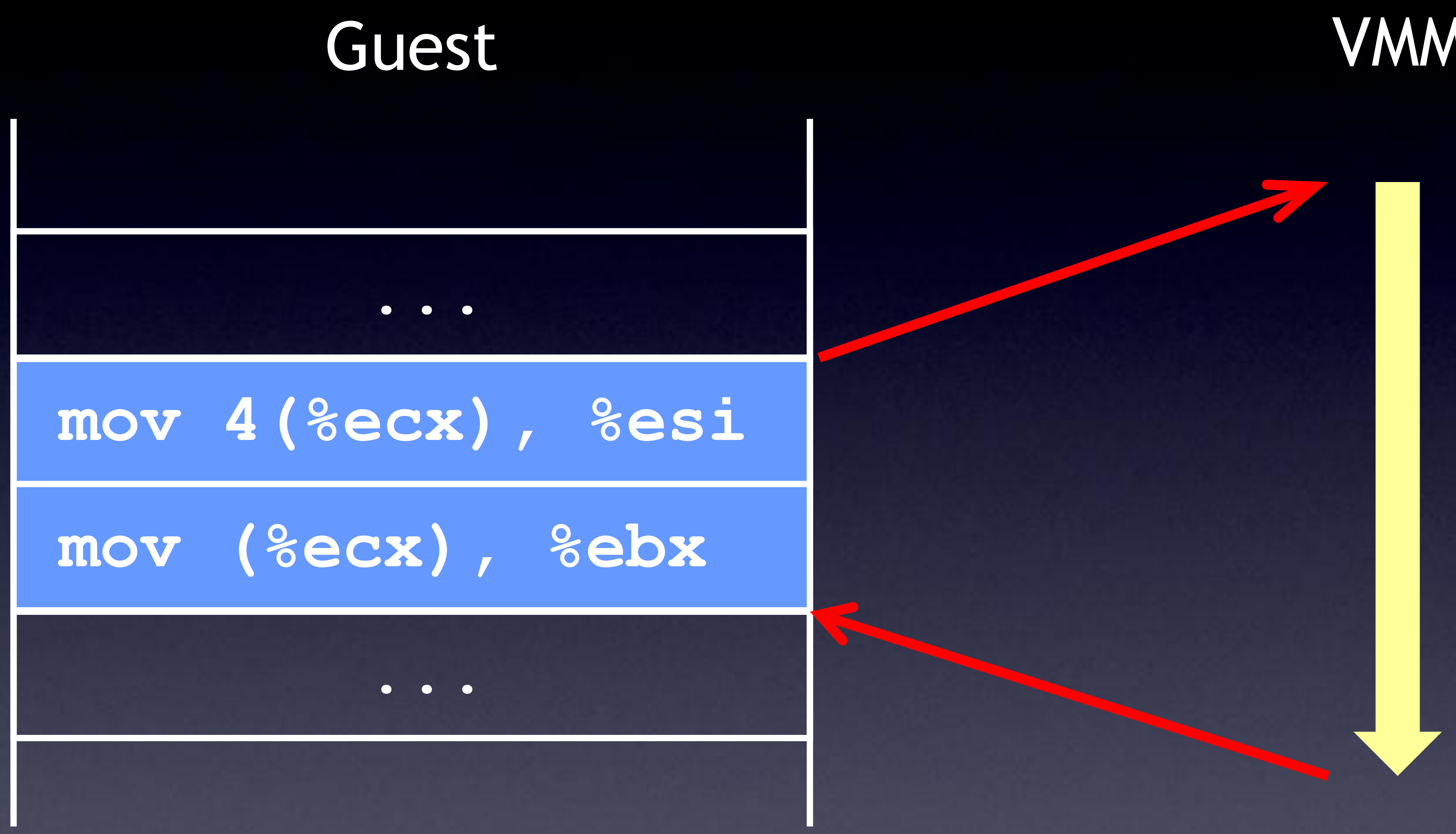
- 32-bit Guest OS using Physical Address Extension (PAE)
- Shadow paging
- Page table entry updates use two 32-bit writes
- Each write causes an exit

Exit Pairs



VMM updates shadow page table entries

Exit Pairs



Upon exit:

- VMM inspects next instruction
- Detects access to adjacent byte
- Executes both instructions

Exit Pairs

Upon exit:

- VMM inspects next instruction
- Detects access to adjacent byte
- Executes both instructions
- Optimizes execution

Exit Clusters

- Upon exit:
  - Scan a few (up to 16) instructions downstream
  - Identify exiting instructions
  - Form a cluster of instructions, executed all at once
- Challenges:
  - Cluster formation
  - Efficient execution

Cluster Formation

Cluster Formation

Guest

```
in %al, %dx
out 0x80, %al
mov %al, %cl
mov %dl, 0xc0
out %al, %dx
out 0x80, %al
xchg %ah, %al
xor %cl, %cl
```

VMM

HV exit occurs
Decode instructions

Cluster Formation

Guest

```
in %al, %dx
out 0x80, %al
mov %al, %cl
mov %dl, 0xc0
out %al, %dx
out 0x80, %al
xchg %ah, %al
xor %cl, %cl
```

VMM

HV exit occurs
Decode instructions
Identify exiting instructions

Cluster Formation

Guest

```
in %al, %dx
out 0x80, %al
mov %al, %cl
mov %dl, 0xc0
out %al, %dx
out 0x80, %al
```

VMM

HV exit occurs
Decode instructions
Identify exiting instructions
Form cluster

Cluster Formation

Guest

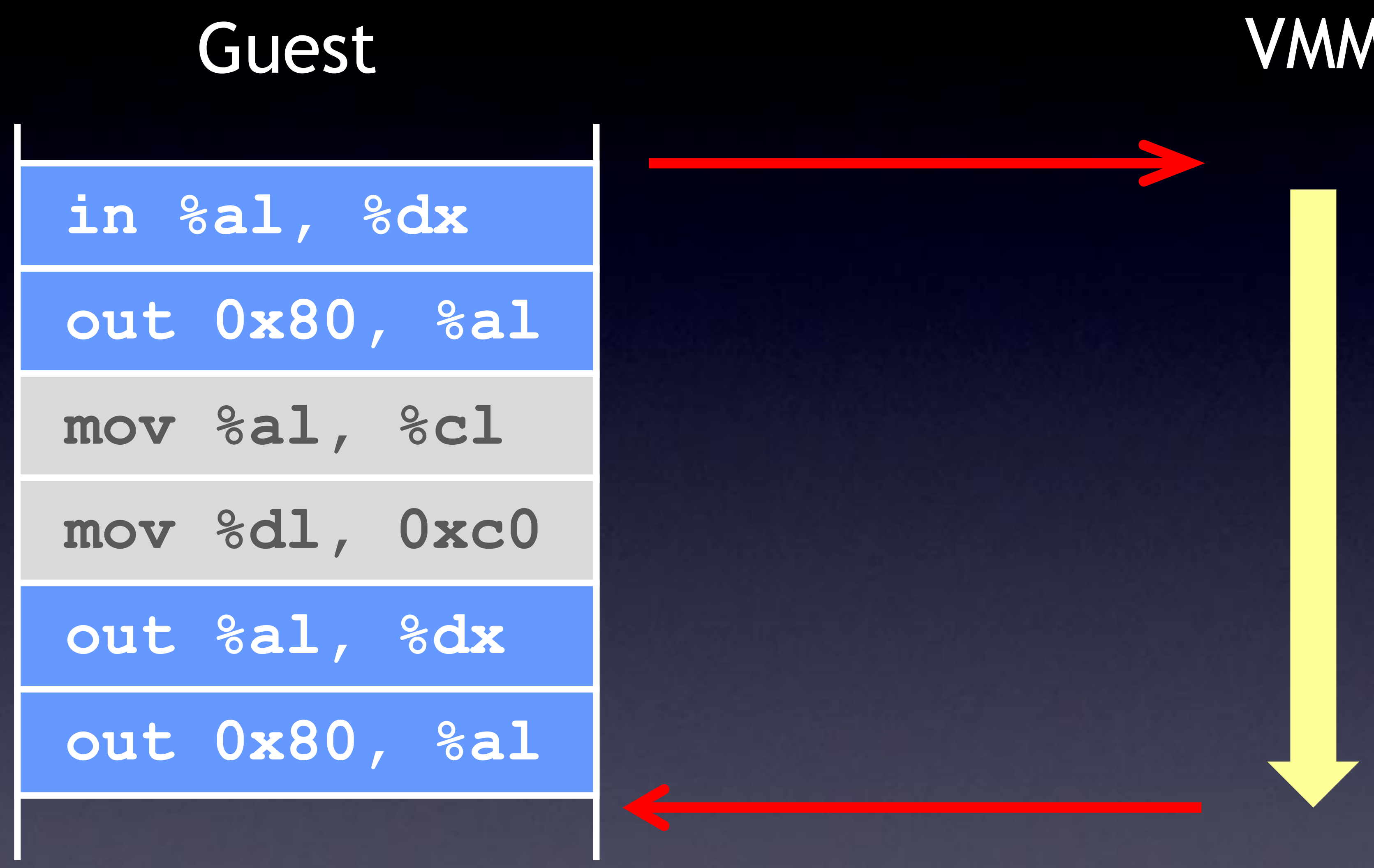
```
in %al, %dx
out 0x80, %al
mov %al, %cl
mov %dl, 0xc0
out %al, %dx
out 0x80, %al
```

gap fillers

VMM

HV exit occurs
Decode instructions
Identify exiting instructions
Form cluster

Cluster Formation

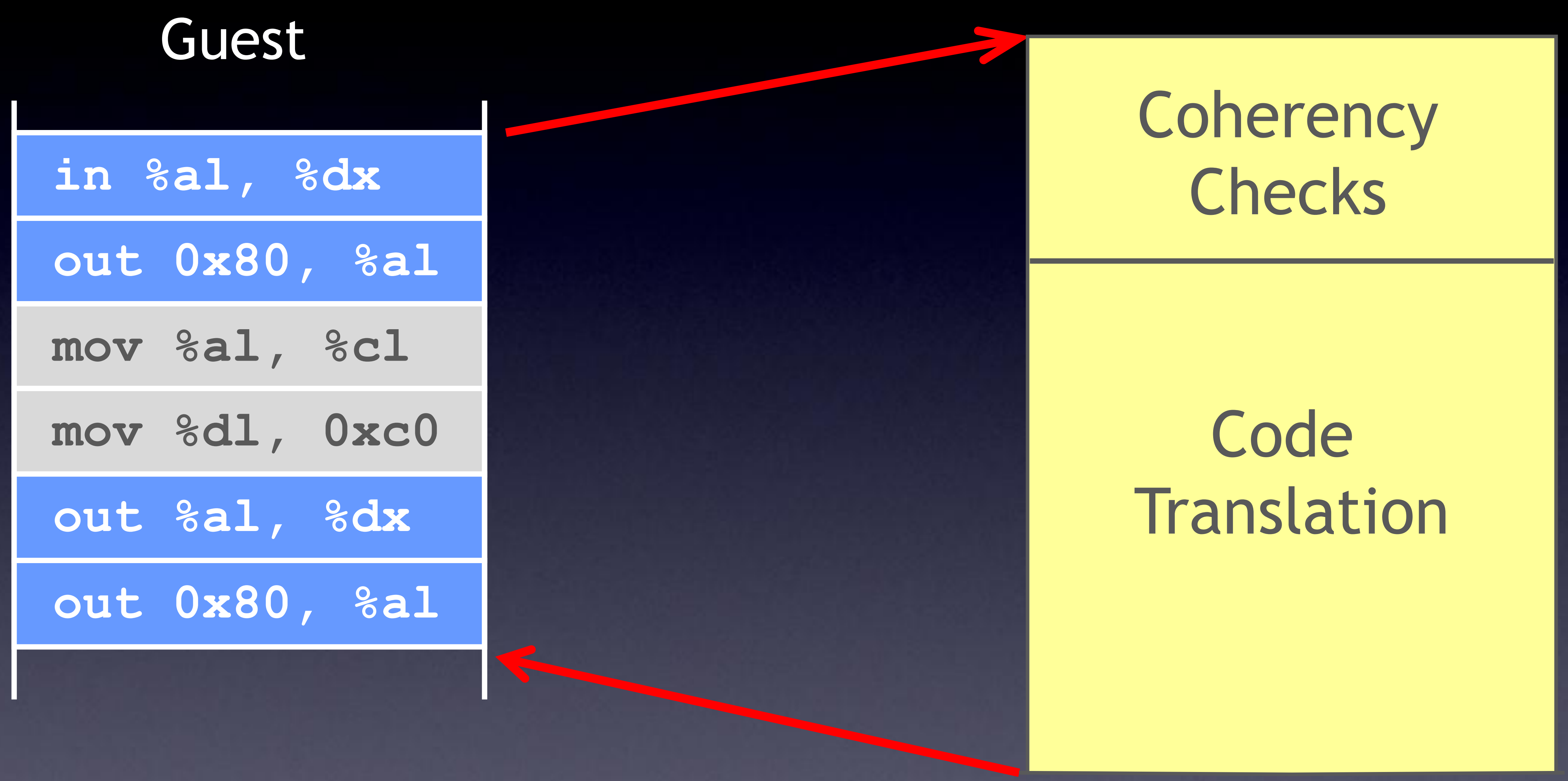


HV exit occurs
Decode instructions
Identify exiting instructions
Form cluster
Execute and resume HV

Efficient Exit Handling

- Cluster translation:
  - Generate a translation for the cluster
  - Insert it in a Translation Cache (TC)
  - Specialize translations (e.g., on addressing mode)
- Translation reuse:
  - Compile once
  - Reuse for all subsequent exits

Cluster Translation



Dynamic Cluster Formation

Guest

```
out %al, %dx
mov %cx, -0x12(%bp)
sub %si, %cx
mov %dl, 0x3c5
mov %al, 0x1
out %al, %dx
shr %bl
sbb %ah, %ah
shl %al
mov es:(%di), %ah
```

Dynamic Cluster Formation

Guest

```
out %al, %dx
mov %cx, -0x12(%bp)
sub %si, %cx
mov %dl, 0x3c5
mov %al, 0x1
out %al, %dx
shr %bl
sbb %ah, %ah
shl %al
mov es:(%di), %ah
```

Do memory accesses cause exits?

Dynamic Cluster Formation

Guest

```
out %al, %dx
mov %cx, -0x12(%bp)
sub %si, %cx
mov %dl, 0x3c5
mov %al, 0x1
out %al, %dx
shr %bl
sbb %ah, %ah
shl %al
mov es:(%di), %ah
```

Do memory accesses cause exits?

Sometimes.

Instruction Classification

- Strongly exiting:
  - Always cause exits
  - Examples: `in`, `out`, `cpuid`
- Weakly exiting :
  - Dynamic exiting behavior
  - Example: memory accesses
  - Runtime VMM support for detecting such cases

Dynamic Cluster Formation

Guest

3	`out %al, %dx`
0	`mov %cx, -0x12(%bp)`
0	`sub %si, %cx`
0	`mov %dl, 0x3c5`
0	`mov %al, 0x1`
2	`out %al, %dx`
0	`shr %bl`
0	`sbb %ah, %ah`
0	`shl %al`
2	`mov es:(%di), %ah`

Translation postponed
Count exits
Translate on 3rd exit

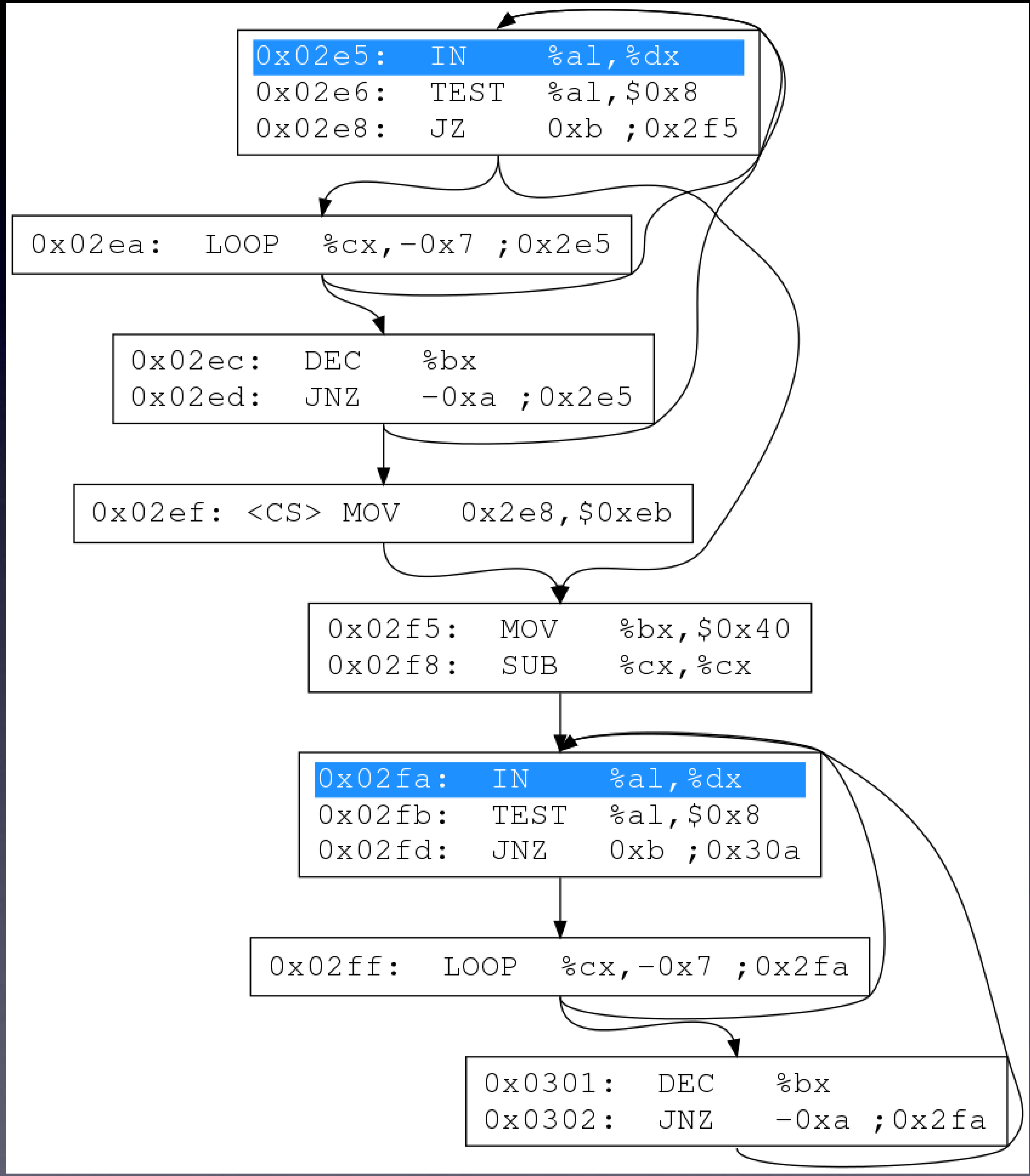
Dynamic Cluster Formation

Guest

3	`out %al, %dx`
0	`mov %cx, -0x12(%bp)`
0	`sub %si, %cx`
0	`mov %dl, 0x3c5`
0	`mov %al, 0x1`
2	`out %al, %dx`
0	`shr %bl`
0	`sbb %ah, %ah`
0	`shl %al`
2	`mov es:(%di), %ah`

Translation postponed
Count exits
Translate on 3rd exit

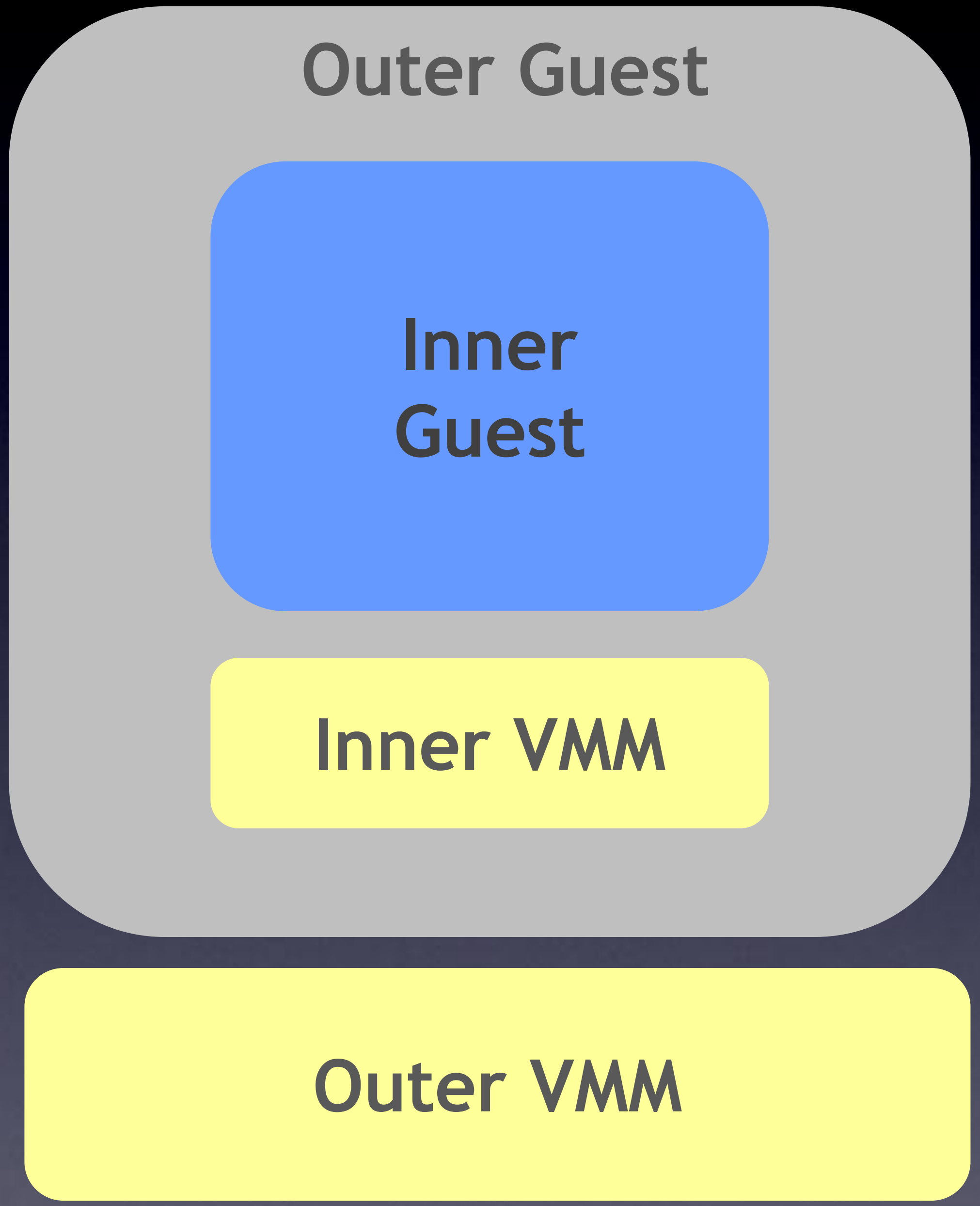
Cluster With Complex Control-Flow



Nested Virtual Machines

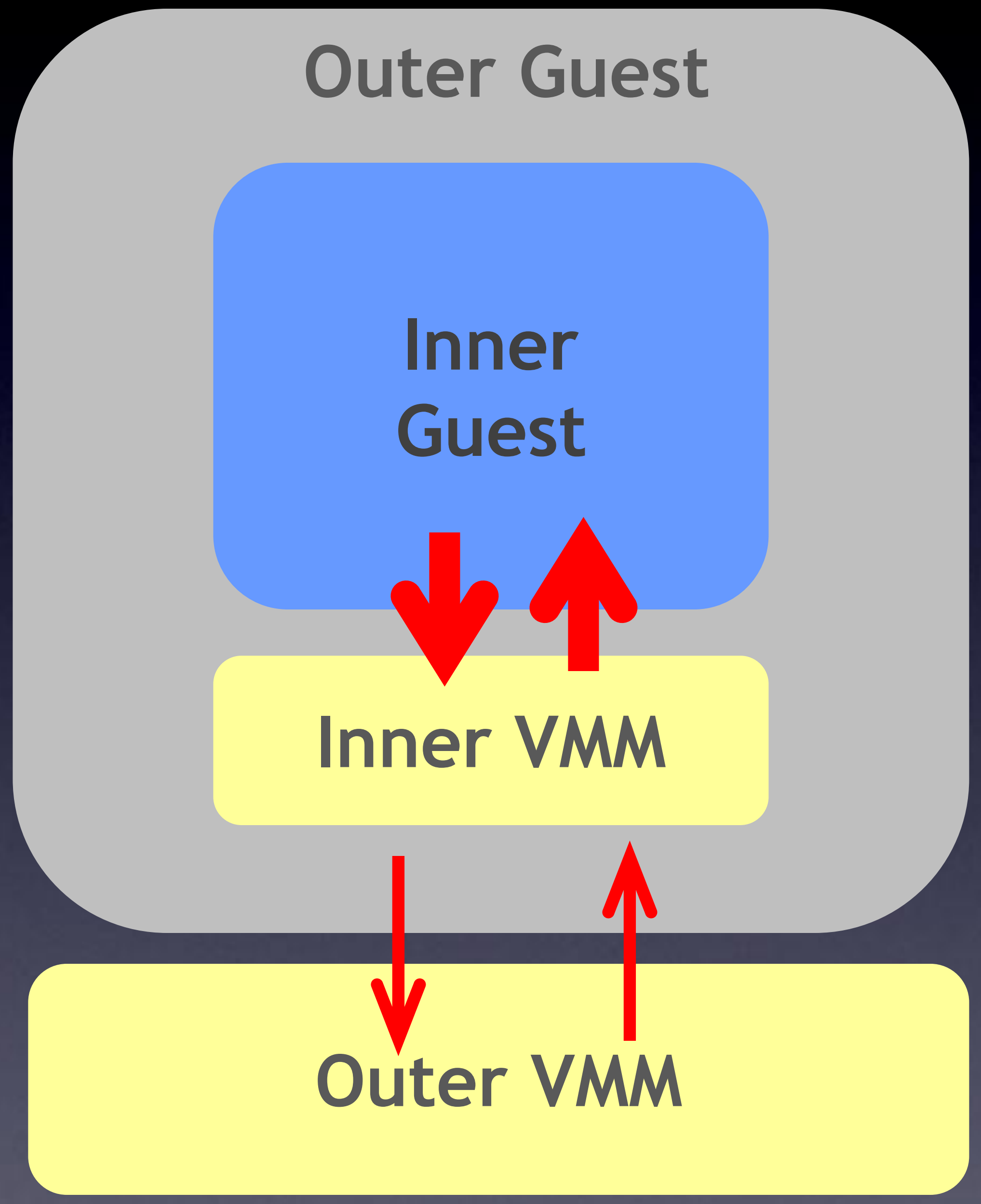
Nested VMs

- Run hypervisor as a guest
  [Ben-Yehuda et al., OSDI'10]
- Simulate large-scale virtualized environments with fewer hosts
- Training, testing, debugging
- "Windows XP mode" in Win 7



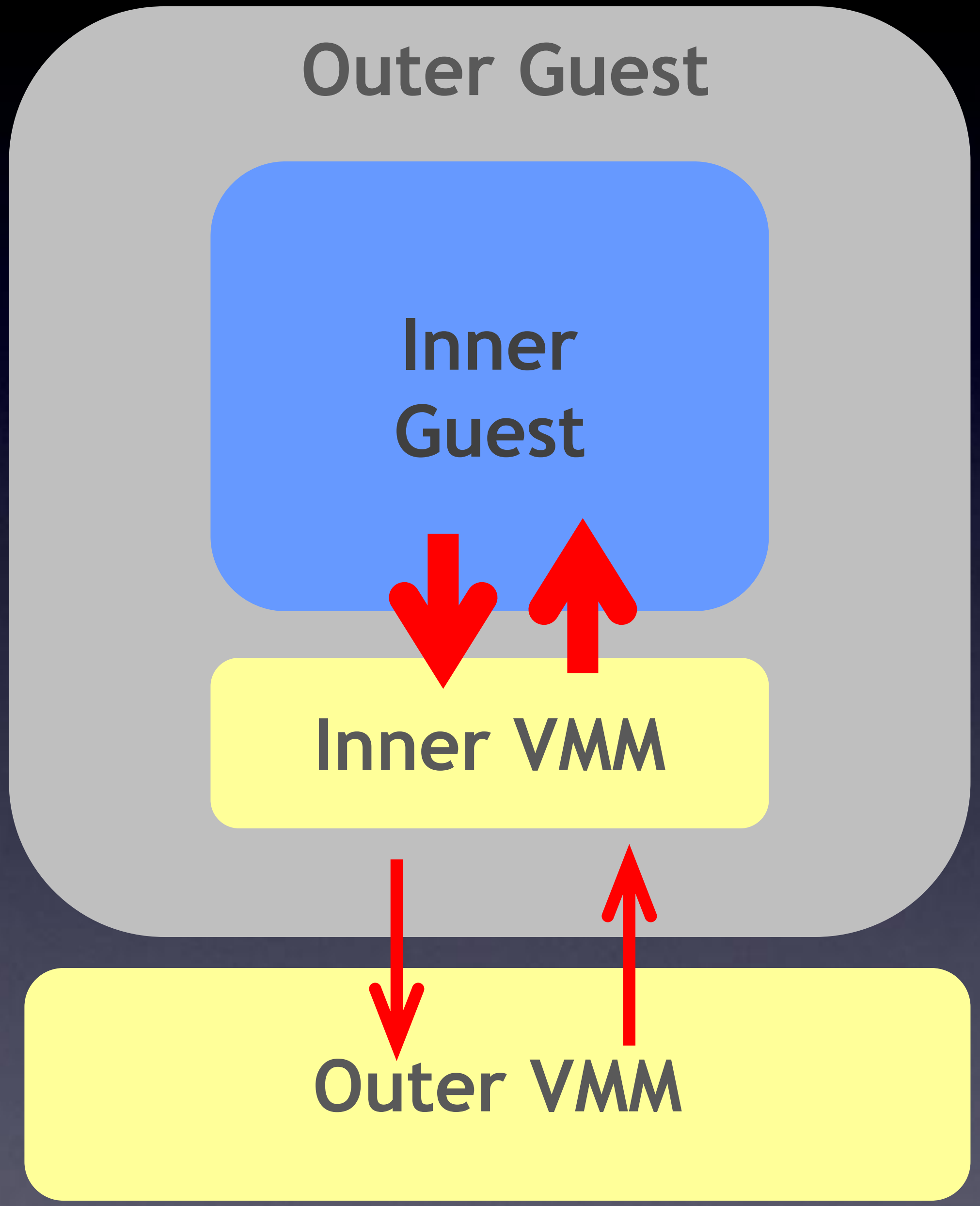
HV for Nested VMs

- Outer VMM transitions
  - Handled in hardware
- Inner VMM transitions
  - Virtual hardware exits are emulated in software



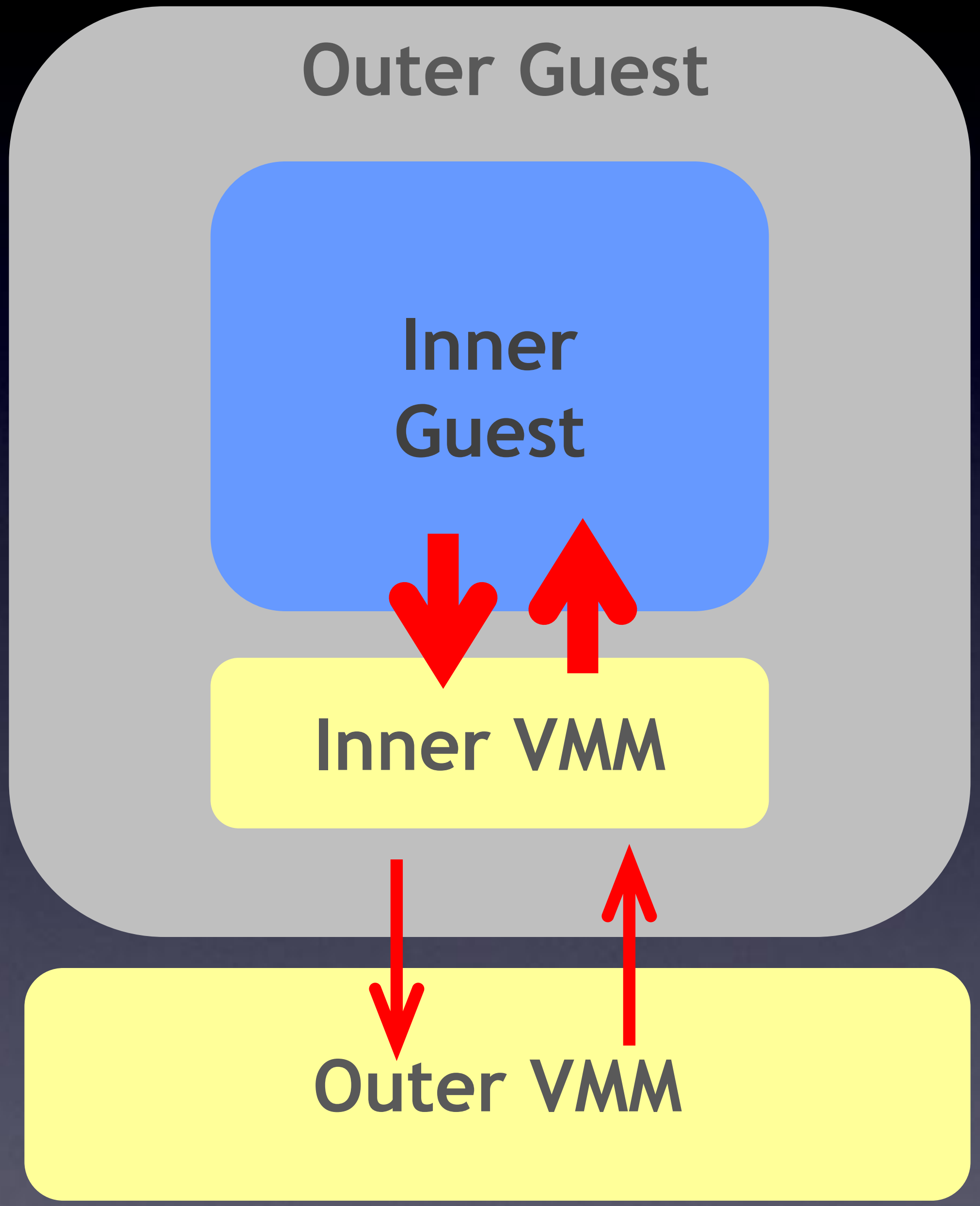
HV for Nested VMs

- Outer VMM transitions
  - Handled in hardware
- Inner VMM transitions
  - Virtual hardware exits are emulated in software
  - Virtual software exit path has lots of exiting instructions (e.g., `vmread`, `vmwrite`)



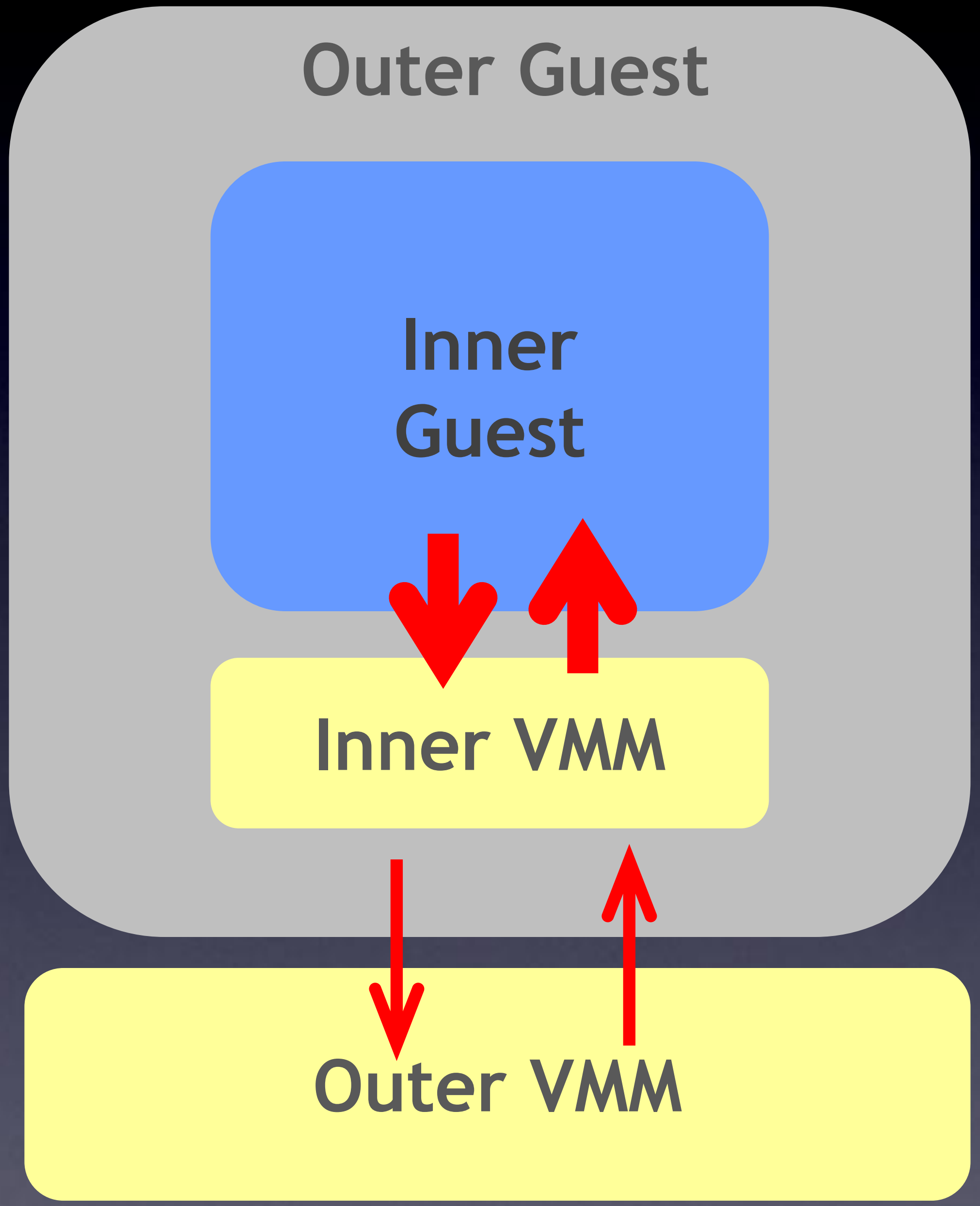
HV for Nested VMs

- Outer VMM transitions
  - Handled in hardware
- Inner VMM transitions
  - Virtual hardware exits are emulated in software
  - Virtual software exit path has lots of exiting instructions (e.g., `vmread`, `vmwrite`)
  - 10x slowdown!



Exit Avoidance

- Cluster inner exits
  - Big impact due to high cost of inner exits
- Cluster outer exits:
  - Big impact due to the high frequency of outer exits
  - Opportunity: cluster-friendly VMM instruction scheduling



Experimental Results

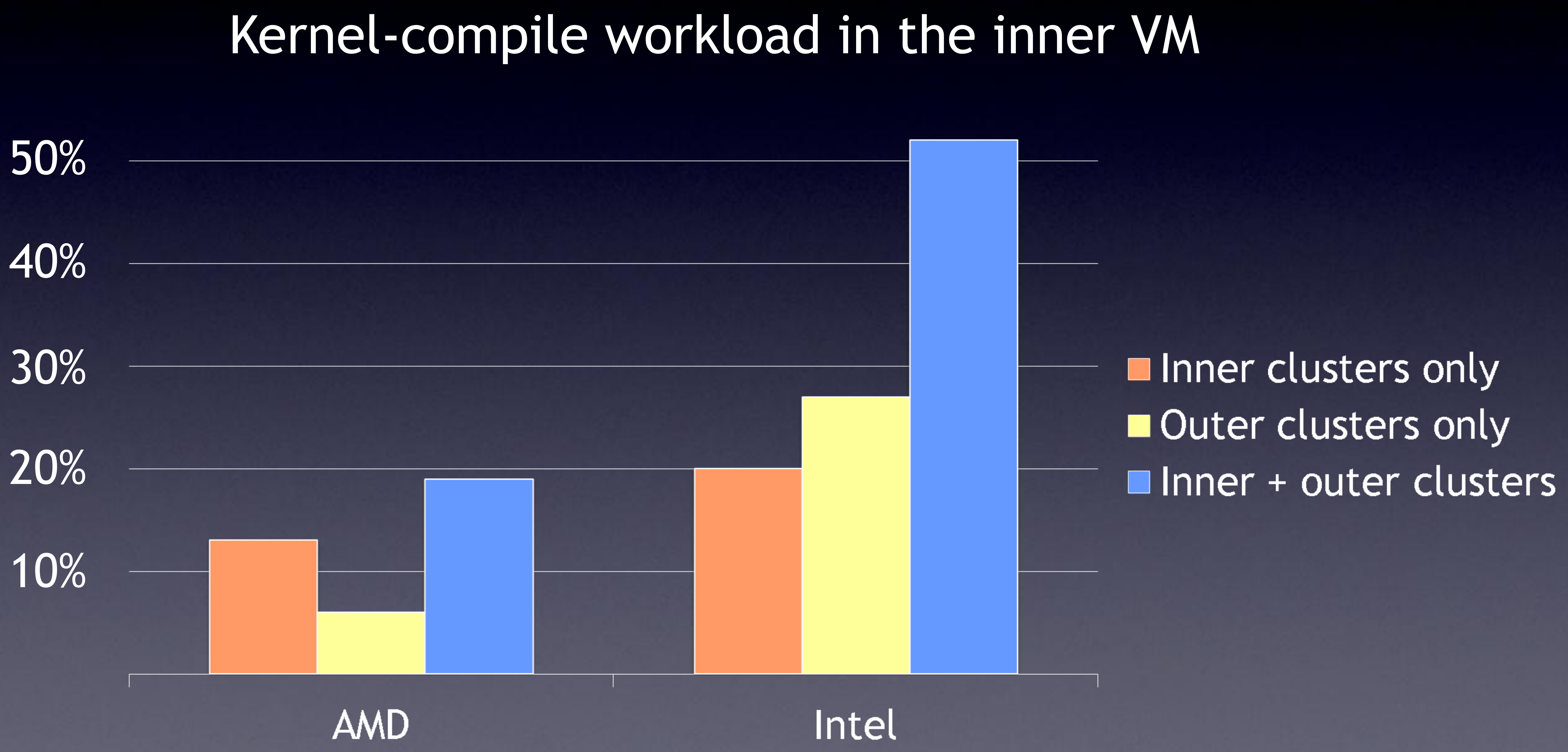
Implementation

- Fully implemented in VMware products
  - Workstation, Fusion, ESX
- Evolved over many years from exit pairs to complex clusters
- Validated by use in the field

Non-Nested VMs

- PassMark (2D graphics benchmark)
  - Clustering improves score by 50% - 80%
- VMmark (virtualization benchmark)
  - Consistent exit rate reduction
  - No measurable runtime improvement
- Netperf against a VM with virtual e1000 NIC
  - 24% reduction of exits per packet roundtrip

Nested VM Speedup



Future Directions

- Support more complex clusters
  - Non-contiguous clusters
  - Exits in the middle of loops
- Optimize memory accesses in clusters
  - Cache and reuse work for accesses on the same page
- Applications to virtualized low-latency workloads

Conclusions

- HV exits are expensive, can sap performance
- Software exit clustering complements hardware optimizations for HV exits
- Clustering is key to enabling reasonable performance for nested VMs