Address Translation in a Virtual Machine Environment

Lena Olson, presenting work from Bhargava et al, Hoang et al, Wang et al

March 6, 2012
Set Up

- Classic Address Translation
  - Virtual Address (VA) $\rightarrow$ Physical Address (PA)

- Adding Virtualization
  - Guest VA $\rightarrow$ Guest PA $\rightarrow$ Host PA
  - TLB: Guest VA $\rightarrow$ Host PA
  - Two page tables:
    - Guest VA $\rightarrow$ Guest PA
    - Guest PA $\rightarrow$ Host PA
The Problem

Shadow Page Table: Guest VA → Host PA

“Nested” Page Table Walk: $O(n^2)$ memory references for n-level page tables
  1. 2D Translation Cache: Skip some levels of nested page table
  2. Nested TLB: Guest PA → Host PA
  3. Hash Table: Alternate structure for nested dimension

Dynamically switching between shadow and nested page tables
Address Translation: VA $\rightarrow$ PA

- Check the TLB
- Walk the page table

Figure from AMD-V Nested Paging White Paper
Two-Step Translation

- **Terminology**
  - **GVA**: Guest VA
  - **GPA**: Guest PA
  - **HPA**: Host PA

- Need GPA → HPA

- Need to be fast and transparent to guest
The Problem

**Shadow Page Table:** Guest VA $\rightarrow$ Host PA

“Nested” Page Table Walk: $O(n^2)$ memory references for n-level page tables

1. 2D Translation Cache: Skip some levels of nested page table
2. Nested TLB: Guest PA $\rightarrow$ Host PA
3. Hash Table: Alternate structure for nested dimension

Dynamically switching between shadow and nested page tables
Shadow Page Table

- Early version: Disco on MIPS R10000
- TLB is software managed
- OS keeps GVA → GPA page tables
- VMM keeps GPA → HPA page tables (called pmap)
- TLB has GVA → HPA mapping
- TLB modifications by guest are intercepted and corrected

Source: Disco: Running commodity operating systems on scalable multiprocessors, Bugnion et al. SOSP’97
Add a third structure: shadow page table

- Translate GVA → HPA
- Walk it in hardware
- Keep it synchronized with host/guest page tables

No hardware support needed
Shadow Page Table: Disadvantages

- Must keep page tables consistent
- Every time guest modifies its page tables, trap to VMM
  - Can sometimes delay the updates to shadow page table
  - Must distinguish between guest/host causes of page faults
- Need to keep dirty bits up to date in guest page table
- Overhead of storing / recreating shadow page tables
Outline

- The Problem
- Shadow Page Table: Guest VA → Host PA
- "Nested" Page Table Walk: $O(n^2)$ memory references for n-level page tables
  1. 2D Translation Cache: Skip some levels of nested page table
  2. Nested TLB: Guest PA → Host PA
  3. Hash Table: Alternate structure for nested dimension
- Dynamically switching between shadow and nested page tables
Nested Page Table

- Walk both host and guest page tables in hardware
- Guest can modify its own page table
- No consistency to worry about
- Also known as Extended Page Table (EPT), Rapid Virtualization Indexing (RVI)
Nested Page Table: Page Table Walk

Source: Accelerating Two-Dimensional Page Walks for Virtualized Systems, Bhargava et al. ASPLOS’08
Nested Page Table: Disadvantages

- Every access to guest page table might result in a host walk
- Worst case for $n$ level host, $m$ level guest page table
  - $n \times m + n + m$ page entry references!
- Nested walks on average 3.9-4.6x slower than native

Source: Accelerating Two-Dimensional Page Walks for Virtualized Systems, Bhargava et al. ASPLOS’08
1. 2D Translation Cache

- Page Walk Cache (PWC) or Paging Structure Caches
- For a non-virtualized system:
  - Stores partial translations
  - Allows the page walk to skip one or more levels of the table
1. 2D Translation Cache

- Cache the nested dimension too
- Since there is a lot of locality, this should help
- Percentage of all unique page entries for each reference:

![Diagram showing percentage of unique page entries for each level of L cache]

Source: Accelerating Two-Dimensional Page Walks for Virtualized Systems, Bhargava et al. ASPLOS’08
2. Nested TLB

- Keep GPA → HPA translations for page table entries
- Avoid more steps of the page walk

Source: Accelerating Two-Dimensional Page Walks for Virtualized Systems, Bhargava et al. ASPLOS’08
2. Nested TLB

Source: Accelerating Two-Dimensional Page Walks for Virtualized Systems, Bhargava et al. ASPLOS’08
2. Nested TLB: 2D Translation Cache + NTLB

Access rate and hit percentage per page entry reference

Source: Accelerating Two-Dimensional Page Walks for Virtualized Systems, Bhargava et al. ASPLOS’08
2. Nested TLB: Speedups

Speedup for each configuration. 2MB NP entry uses large pages.

Source: Accelerating Two-Dimensional Page Walks for Virtualized Systems, Bhargava et al. ASPLOS’08
3. Hash Table

- Why use a multi-level page table for both?
- Use a hash table for the nested page table
- Expected nested page walk length: $O(2n)$ steps

Source: A Case for Alternative Nested Paging Models for Virtualized Systems, Hoang et al. 2010
3. Hash Table: Performance Comparison

Source: A Case for Alternative Nested Paging Models for Virtualized Systems, Hoang et al. 2010
Outline

- The Problem
- Shadow Page Table: Guest VA $\rightarrow$ Host PA
- “Nested” Page Table Walk: $O(n^2)$ memory references for n-level page tables
  1. 2D Translation Cache: Skip some levels of nested page table
  2. Nested TLB: Guest PA $\rightarrow$ Host PA
  3. Hash Table: Alternate structure for nested dimension

- Dynamically switching between shadow and nested page tables
Dynamically Switching

- When are nested page tables better than shadow page tables?
  - Page table modifications are cheaper with nested page tables
  - TLB misses are cheaper with shadow page tables
- Dynamically switch between modes

Source: Selective Hardware/Software Memory Virtualization, Wang et al., VEE’11
Dynamically Switching

- Sample page fault and TLB miss rates
- Compare to thresholds
  - Page fault rate high and TLB miss rate low: nested paging
  - TLB miss rate high and page fault rate low: shadow paging
  - Both low: stay in current mode
  - Both high: use ratio of rates to decide
- Switching is expensive: have to rebuild shadow page table

Source: Selective Hardware/Software Memory Virtualization, Wang et al., VEE’11
Dynamically Switching: Execution Time

Dynamically switching as good as minimum of shadow/nested paging

Source: Selective Hardware/Software Memory Virtualization, Wang et al., VEE’11
The Problem

> Shadow Page Table: Guest VA $\rightarrow$ Host PA

> “Nested” Page Table Walk: $O(n^2)$ memory references for n-level page tables
  1. 2D Translation Cache: Skip some levels of nested page table
  2. Nested TLB: Guest PA $\rightarrow$ Host PA
  3. Hash Table: Alternate structure for nested dimension

Dynamically switching between shadow and nested page tables
Conclusion

- GVA $\rightarrow$ HPA translation is nontrivial
  - Even with hardware support!
- Nested page tables are not always better
- Translation mechanism can have a large effect on performance
References

AMD.
AMD-V™ nested paging revision 1.0.

Bhargava, R., Serebrin, B., Spadini, F., and Manne, S.
Accelerating two-dimensional page walks for virtualized systems.

Bugnion, E., Devine, S., and Rosenblum, M.
Disco: running commodity operating systems on scalable multiprocessors.

Hoang, G., Bae, C., Lange, J., Zhang, L., Dinda, P., and Joseph, R.
A case for alternative nested paging models for virtualized systems.

Wang, X., Zang, J., Wang, Z., Luo, Y., and Li, X.
Selective hardware/software memory virtualization.
Questions?