18-447 Lecture 19: Survey of Modern VMs

James C. Hoe
Department of ECE
Carnegie Mellon University

Housekeeping

◆ Your goal today
  - see the many realizations of “VM”, focusing on deviation from textbook-conceptual norms
◆ Notices
  - Midterm 2 on Monday 4/10 (cover L11~19)
  - Lab 4, status check next week
  - HW 4, due on Wednesday
◆ Readings
  - should have read Jacob and Mudge for today
  - start reading P&H Ch 6
EA, VA and PA (IBM Power view)

EA, divided into X fixed-size segments

EAi, divided into X fixed-size segments

VA divided into Y segments (Y>>X); also divided as Z pages (Z>>Y)

PA divided into W pages (Z>>W)

Swap disk divided into V pages (Z>>V, V>>W)

segmented EA: private, contiguous + sharing
demand paged VA: size of swap, speed of DRAM

EA, VA and PA (almost everyone else)

EA0, with unique ASID=0

EAi, with unique ASID=i

EA and VA almost synonymous

VA divided into N "address space" indexed by ASID; also divided as Z pages (Z>>Y)

PA divided into W pages (Z>>W)

Swap disk divided into V pages (Z>>V, V>>W)

how do processes share pages?
**SPARC V9 PTE/TLB Entry**

- 64-bit Virtual Address
  - an implementation can choose not to map the high-order bits (must be sign-extended from the highest mapped bit)
  - e.g., UltraSPARC 1 maps only the lower 44 bits
- physical address space size set by implementation
- 64 entry fully associative I-TLB and D-TLB

```
+ context,3 + VA<63:13> =

v size nfo IE Soft diag + PA<40:13> + Soft L CP CV + v
```

**TLB Miss Handling**

- SPARC V8 (32-bit) defines a 3-level hierarchical page table for HW MMU page-table walk

```
context
context table descriptors +VA<31:24>
L1 Table: 256 descriptors (1024-byte)
L2 Table: 64 descriptors (256-byte)
L3 Table: 64 PTEs (256-byte)
```

- SPARC V9 (64-bit) defines Translation Storage Buffer
  - a software managed, direct-mapped “cache” of PTEs
  - HW assisted address generation on a TLB miss, e.g., for 8-k pages
    ```
    TSBbase63:21,
    Logic(TSBbase20:13,VA32:22,size,split?), VA21:13,0000
    ```
  - TLB miss handler (SW) searches TSB. If TSB misses, a slower TSB-miss handler takes over
IBM PowerPC (32-bit)

- segments 256MB regions
- 16-entry segment table
- seg#₄, seg offset₁₆, page offset₁₂
- seg ID₂₄, seg offset₁₆, page offset₁₂
- 128 2-way ITLB and DTLB
- PPN₂₀, page offset₁₂

64-bit PowerPC = 64-bit EA -> 80-bit VA -> 64-bit PA
How many segments in EA?

---

IBM PowerPC Hashed Page Table

- HW table walk
  - VPN hashes into a PTE group (PTEG) of 8
  - 8 PTEs searched sequentially for tag match
  - if not found in first PTE group search a second PTE group
  - if not found in 2nd PTE group, trap to software handler
- Hashed table structure also used for EA to VA mapping in 64-bit implementations
MIPS R10K

- 64-bit virtual address
  - top 2 bits set kernel/supervisor/user mode
  - additional bits set cache and translation behavior
  - bit 61-40 not translate at all (holes in the VA??)

- 8-bit ASID (address space ID) distinguishes between processes

- 40-bit physical address

- Translation -
  “64”-bit VA and 8-bit ASID → 40-bit PA

MIPS TLB

- 64-entry fully associative unified TLB
  - paired: each entry maps 2 consecutive VPNs to 2 different PPNs
  - software managed
    - 7-instruction page table walk in the best case
    - TLB Write Random: chooses a random entry for TLB replacement
    - OS can exclude some number of TLB entry (low range) to be excluded from the random selection, to hold translations that cannot miss or should not miss

- TLB entry
  - N: noncacheable
  - D: dirty (actually a write-enable bit)
  - V: valid
  - G: global entry, i.e., ignore ASID matching
MIPS Bottom-Up Hierarchical Table

- TLB miss vectors to a SW handler
  - page table organization is not hardcoded in ISA
  - ISA only defines HW features to speedup a certain reference page table scheme

- Bottom-Up Table
  - start with a basic 2-level hierarchical table (32-bit case)
  - map all of the L2 tables (empty or not) linearly in the mapped kseg
  - VPN is the index into this linear table in VA

A linear translation table that scales with VA size
Is this okay?

Bottom-Up Table Walk

VA on TLB Miss, trap

Which address space?

VA of PTE
/generated automatically by HW after TLB miss/

Can this load miss in the TLB?
What happens if it misses?

Notice translation also eats up TLB entries!
User TLB Miss Handling

- mfc0 k0,tlbcxt # move the contents of TLB context register into k0
- mfc0 k1,epc # move PC of faulting load instruction into k1
- lw k0,0(k0) # load thru address that was in TLB context register
- mtc0 k0,entry_lo # move the loaded value into the EntryLo register
- tlbwr # write entry into the TLB at a random slot number
- j k1 # jump to PC of faulting load instruction to retry
- rfe # RESTORE FROM EXCEPTION

HP PA-RISC: PID and AID

- 2-level translation: 64bit EA → 96bit VA (global) → 64bit PA
- Variable sized segmented EA to VA
- A different take on protection
  - everyone else: limit what can be named by a process
    - in PowerPC, OS controls what VA can be reached by a process by controlling what’s in the segment registers
  - HP-RISC: rights-based access control
    - user controls segment registers, i.e., user can generate any VA it wants
    - each virtual page has an access ID (not related to ownership by a processes) assigned by the OS
    - each process has 8 active protection IDs in special HW registers controlled by the OS
    - a process can only access a page if it has the key (PID) that fits the lock (AID)
Intel 80386

- Intended for two-level address translation with segments for naming and protection (similar to PPC)
  - user private 48-bit effective address
    - 16-bit segment number (implicit) + 32-bit segment offset
      each addr register has a corresponding segment register
  - a global 32-bit virtual address
    - 20-bit page number + 12-bit page offset
  - an implementation defined paged physical address space

- 32-bit VA too small to be shared by multiple processes
- OS did not use segmentation for protection
  - code, data, stack segments always mapped to 0~(2^{32}-1)
  - time multiplex VA between processes for naming and protection
  - set MMU to use a different table on context switch
  - must flush TLB on context switch because TLB entries are not defined to have ASIDs

Later IA32e added PCID as fix

What is strange about this?

Format of the Quiz

- Coverage
  - lectures (L11~L19), HWs, projects, assigned readings (textbooks and papers)
- Types of questions
  - freebies: can you remember the materials
  - probing: did you understand the materials
  - applied: can you apply the materials in original thoughts
- 100 minutes, 100 points
  - if a question is worth 5 points, don’t spend 20 minutes on it
  - skip questions you can’t do and come back to them later
  - closed-book, one 2-sided 8½x11-in² hand-written cribsheet
  - no calculators, no cell phone call, no electronics

*** Use pencil or black/blue ink only
*** Be on time, 12:30 sharp!!!