2D Page Walks - Notes

Background

- Describe the general process of x86 translation in AMD64 long mode; how big of a virtual address space can this support?

- Multilevel (4 levels) page tables with 48 bit Virtual address input. Each level takes 9 bits of VA as Index (9 X 4 = 36 bits). Remaining 12 bits are used for page offset.

- Given this, how does the basic nested page table approach work? (use Figure 1 as your guide, as well as the text)

- CR3 stores the page table base address for L4 table. This value is used along with 9 bits of virtual address to retrieve the address for L3 table.
- This process is repeated till a value from L1 is obtained, which is combined with 12 bit offset to get the physical address.

- With a guest PT of n levels, and a VMM PT of m levels, the authors say that nm + m + n page entry references are needed; describe why.

- For each gLn level, m levels of Nested Page Tables are walked to get SPA (System Physical Address) at each level and the SPA from level m is used to generate GPA (Guest Physical Address).
- This process is repeated for **n iterations** (**mn + n walks**) and for the final level of n, **m walks** of nested page table gives the final required SPA. This gives the number of page entry references **mn + m + n**. Refer Fig. 1.

- The original system just has CR3; what is this for?

- To store the page table base address.

- What are gCR3 and nCR3, and when are they used during the nested page walk?

- gCR3 is used to store the base address of guest OS page table to obtain GPA for level L4.
- nCR3 stores the base of nested page table. It is used for referencing at all n + 1 levels.

- What ends up in the TLB when this process is complete?

- Guest Virtual Address (gVA) to System Physical Address (SPA) mapping.

Page Walk Characterization

- What do we learn from Table 1? How would you summarize these results? What stands out?

- It summarizes the cost of TLB misses. For a perfect TLB, the performance improvement scope is significant for nested paging compared to native performance.

- What is learned from Figure 3? How should that affect caching designs?

- Unique references are significantly less at higher levels. Hence reuse opportunity is high.
- Caches for higher level page table entries could exploit this reuse opportunity to improve performance.

- What do we learn from Figure 4?

- For {nL1, gPA} page entries level, reuse opportunity is analyzed for the five workloads used for evaluation.
- IntCpu & FpCpu has high spatial locality, where 70% of cache lines have at least 2 valid page entries. This justifies caching {nL1, gPA} level. But it has lower reuse opportunity compared to other higher level of page tables.

Page Walk Acceleration
- What is cached in the 1D PWC approach?

- Only guest OS page table entries. (GVA → GPA entries, except for the last level due to its large number of unique entries)

- What is cached in the 2D approach?

- In addition to 1D approach, 2D approach caches nested page table entries.

- What is different about the third approach (2D + NT)? How does the Nested TLB (NTLB) work?

- A Nested TLB (NTLB) is added. It acts as the guest physical address to system physical address translation buffer.
- It reduces the average number of page entry references during 2D page walk.

How/when is it accessed as compared to the normal TLB?

- 2D page walk is used on Normal TLB miss. So Nested TLB is accessed to find GPA → SPA translation on a Normal TLB miss.

Evaluation
- What does Figure 6 show? What metric do they use?

- Speedups of each of the page walk caching techniques in comparison to the baseline architecture with no page walk caching.
- Performance speedups for each of the benchmarks.

- What do we learn from Figure 7?

- It examines the page walk translation cache access and page entry caching miss for 2D PWC and 2D PWC+NT. It shows that number of access is less in 2D PWC+NT due to the presence of NTLB and hence there is less misses too.

- Which has a bigger impact, bigger PWC or bigger NTLB?

- Bigger PWC. As shown in Fig. 9.

- What else did you learn from the measurements?

- Traces from retired instructions are used instead of whole benchmarks to reduce the simulation time.
- Big pages results in lesser references and reduced TLB pressure.

Not discussed in class

- How is the approach in the paper evaluated? What are your thoughts on this?
- How does increasing TLB size compare to the solution within the paper?
- Does the solution in the paper actually "solve" the problem?

Other Discussions

- Contents of gCR3 changes on Hypervisor rescheduling.
- Contents of nCR3 changes for different VMs on the machine.
- On PWC miss, page entry data may reside in L2, L3 cache or main memory. Not caching on L1 is a design decision. It could be due to the size and high performance requirement of L1 cache.