Questions for 2D Page Walks

Background

- Describe the general process of x86 translation in AMD64 long mode; how big of a virtual address space can this support?
- Given this, how does the basic nested page table approach work? (use Figure 1 as your guide, as well as the text)
- 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.
- The original system just has CR3; what is this for?
- What are gCR3 and nCR3, and when are they used during the nested page walk?
- What ends up in the TLB when this process is complete?

Page Walk Characterization

- What do we learn from Table 1? How would you summarize these results? What stands out?
- What is learned from Figure 3? How should that affect caching designs?
- What do we learn from Figure 4?
- What does all of this imply for caching?

Page Walk Acceleration

- What is cached in the 1D PWC approach?
- What is cached in the 2D approach?
- What is different about the third approach (2D + NT)? How does the Nested TLB (NTLB) work? How/when is it accessed as compared to the normal TLB?
- In general, what is assumed about PWC access time, as opposed to other parts of the memory hierarchy?

Evaluation

- How is the approach in the paper evaluated? What are your thoughts on this?
- What does Figure 6 show? What metric do they use?
- What do we learn from Figure 7?
- Which has a bigger impact, bigger PWC or bigger NTLB?
- What else did you learn from the measurements?

General

- How does increasing TLB size compare to the solution within the paper?
- Does the solution in the paper actually "solve" the problem?