Questions about "Xen and the Art of Virtualization"

Intro
- Three challenges listed; what are they? Does the paper address them?
- What is the main point that the intro makes to motivate Xen?
- How would you write an intro for this paper?
- Denali targeted thousands of hosted OS instances; Xen 100.
  What is the usual limit to such instances? What is a realistic goal?

Xen: Approach and Overview
- What are the reasons listed for the partial virtualization Xen supports?
- Four design principles are listed; what are they, and what do you think of them?
- Claim is that with a software-managed TLB, virtualization is easier;
  why is this so? (What is hard about a hardware-managed approach?)
- Is Xen architecture motivated by difficulty, or other reasons?
- Why is it important that Xen is mapped into each VM address space?
- How is the Xen approach to guest page tables different than shadow
  page tables used by more traditional VMMs on x86?
  (Think about typical PT operations, such as adding mappings, changing protection, etc.)
- Why are x86 rings important? What does this remind you of from Disco?
- The text says "A table describing the handling for each type of exception is registered with Xen";
  how is this different than a typical VMM?
- What then happens upon an exception?
- Why must the page-fault handler be written differently than usual?
- How are system calls handled? (Any concerns here?)
- Domain0: Is there a good question to ask about this? What is it??

Detailed Design
- How come every paper claims to be working on some other OS port?
- What is the main communication mechanism between OS and Xen?
  What is interesting about it?
- How is time virtualized?
- For memory virtualization, how does Xen "validate" a PT update?
- OSes can batch PT updates for performance; what kind of race does this introduce?
- How is physical memory managed by Xen? By an OS on Xen?
- Why are low-level exposed machine addresses potentially useful to the OS?
- Disks: why are reorder barriers needed?

Evaluation
- What is the experimental setup for this paper? Anything interesting about it?
- What does Figure 3 show us?
- What do we learn from Tables 3, 4, and 5?
  Do system calls execute quickly on Xen? Why?
  Why does fork take so long? Does the hypercall cost fully explain the difference?
  Why should micro benchmarks not be taken "too seriously"??
- What does the performance isolation experiment show us?
- What was the result of the "scalability" experiment?