Graduate Operating Systems
(Virtual Machines)

Fall 2020

Paper “Architecture”

• Interoperability, impregnability, versatility
• Interfaces (ISA) & abstractions (files)
• Virtualization vs. abstraction
• Architecture vs. implementation
• ISA, ABI, API
• Process vs. system
Paper “Architecture”

• Process VM = execute individual process
• System VM = complete system environment
• Guest, host, run-time, VMM

Paper “Architecture”

• Process VM
  – Replication: multiprogramming
  – Emulation: different HW, interpretation, dynamic binary translation (+ cache)
  – Optimization: same-ISA optimizers
  – High-level language VM
Paper “Architecture”

- System VM
  - Multiple, isolated guest Oses
  - Isolation, platform replication
  - Classic system VMs
  - Hosted VMs
  - Whole-system VMs
  - Multiprocessor virtualization
  - Codesigned VMs

Paper “Xen”

- Goals of Xen; challenges of VMs
- Resource containers
- Accounting issue, QoS crosstalk issue
Paper “Xen”

- Memory management
  - TLB: SW/HW, tagged/flush
  - Push page table responsibility to guest OS
  - Xen avoids TBL flush
    - Give guest OS control over page table management
    - Protect Xen from triggering flushing

Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including Domain0 running control software in a XenLinux environment.
Paper “Xen”

• CPU management
  – Privilege levels
  – Validate privileged calls by Xen
  – System calls handled without Xen involvement
• I/O management
  – Xen does not emulate devices
  – Uses shared-memory buffer-descriptor rings

Paper “Xen”

• Hypercalls and events
• I/O rings
• BVT scheduling
• Virtual address translation
• Physical memory
• Virtual firewall-router
• Disk