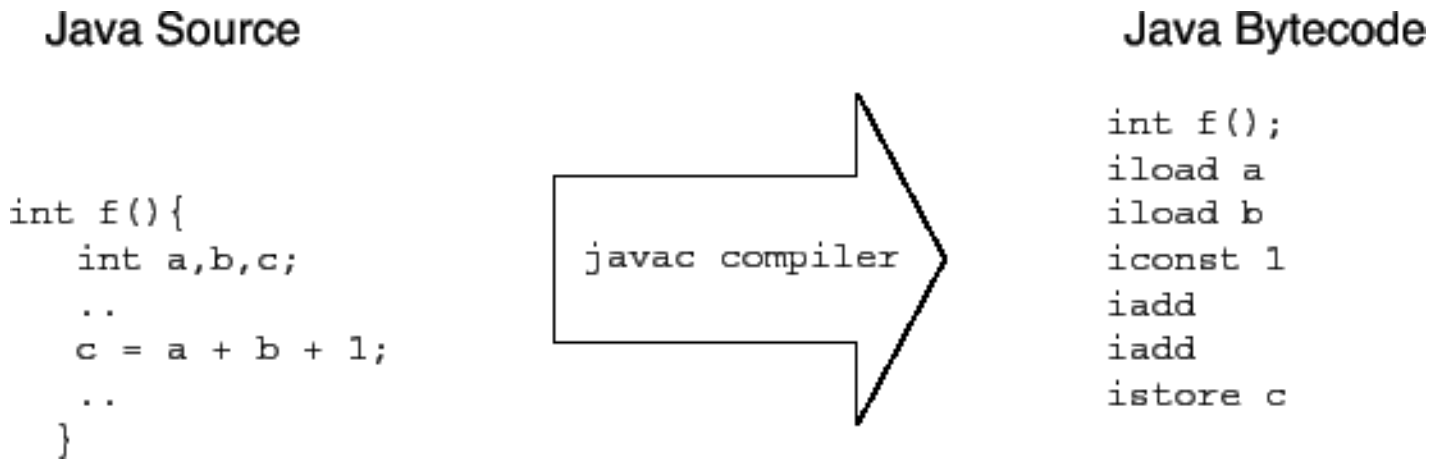
Virtual Machine Monitor

Recap

- Emulator
  - Emulate hardware (e.g., Nintendo) in software
- Java virtual machine (JVM)
  - Platform neutral machine (typically s/w) that runs java bytecode
- Virtual machine monitor (VMM)
  - A kind of OS for virtual machines

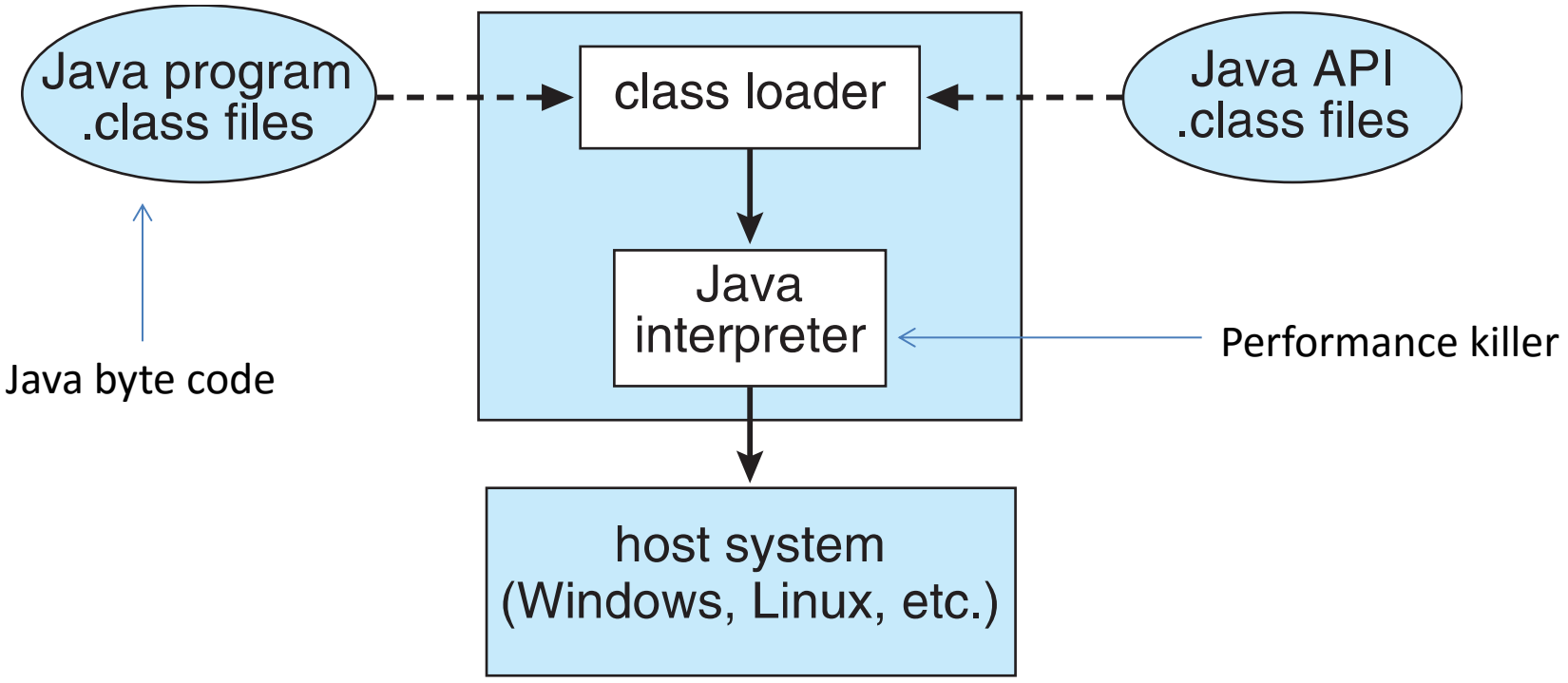
Java Bytecode



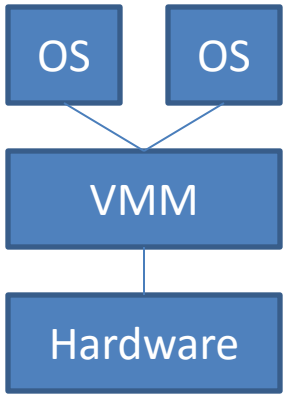
- Java bytecode = instructions for JVM
- One byte opcode + variable length args
  - 198 instructions are in use

Figure source: http://www.cs.toronto.edu/~matz/dissertation/matzDissertation-latex2html/node5.html

Recap: Java Virtual Machine

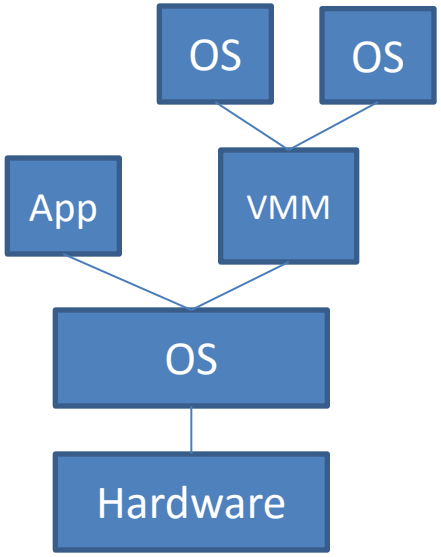


Recap: Types of VMM



- Native (or Type 1) VMM
  - VMM runs directly on top of bare hardware
  - Vmware ESX, Microsoft Hyper-V
  - VMM is a kind of a OS on its own right

- Hosted (or Type 2) VMM
  - VMM runs within an OS
  - VirtualBox, VMWare Workstation
  - VMM relies on functionalities of the host OS



Today

- Virtualization
- Container
- Docker

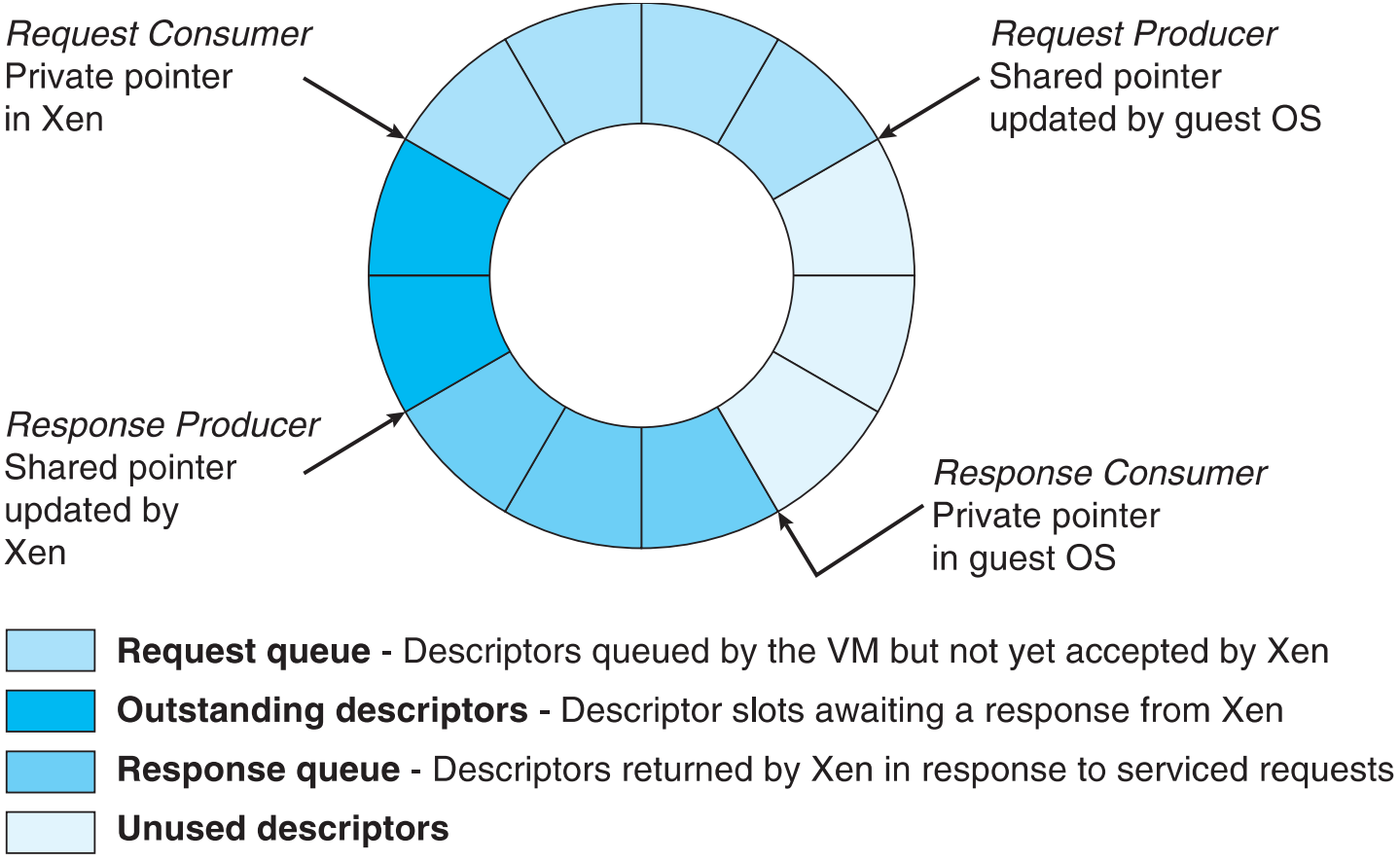
Virtualizing Interrupts & I/O

- VMM receives h/w interrupts
  - Determines which VM to receive
  - Emulate interrupt controller for the VM
- VMM emulate a specific h/w devices
  - Guest OS → VMM → devices
    - E.g., AMD Lance PCNet ethernet device
- Lots of I/O → performance killers

Para-virtualization

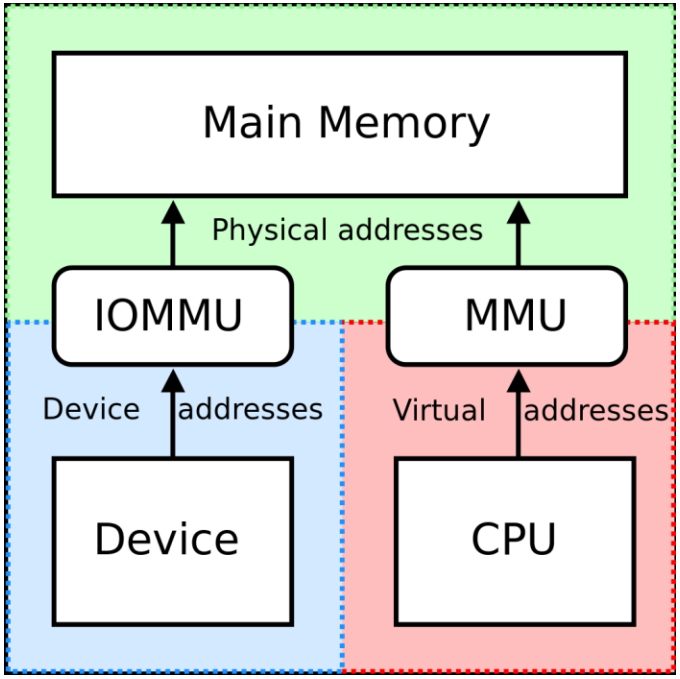
- Idea: provides simple/fast APIs to guests
  - Instead of emulating actual hardware (e.g., PCNet32 ethernet card)
  - Pros
    - can be a lot faster (more efficient I/O)
  - Cons
    - need to modify the guest OS

I/O in Xen via Shared Buffer



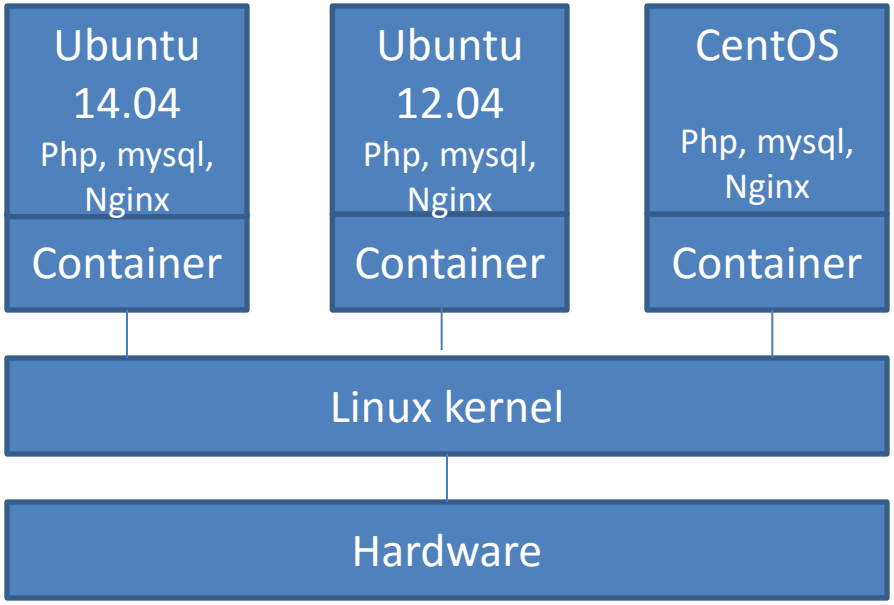
IOMMU

- Problem: How to do DMA in a VM?
  - DMA controller needs host physical address, not guest physical address
- IOMMU
  - MMU for IO devices
  - maps guest physical → host physical for the I/O devices



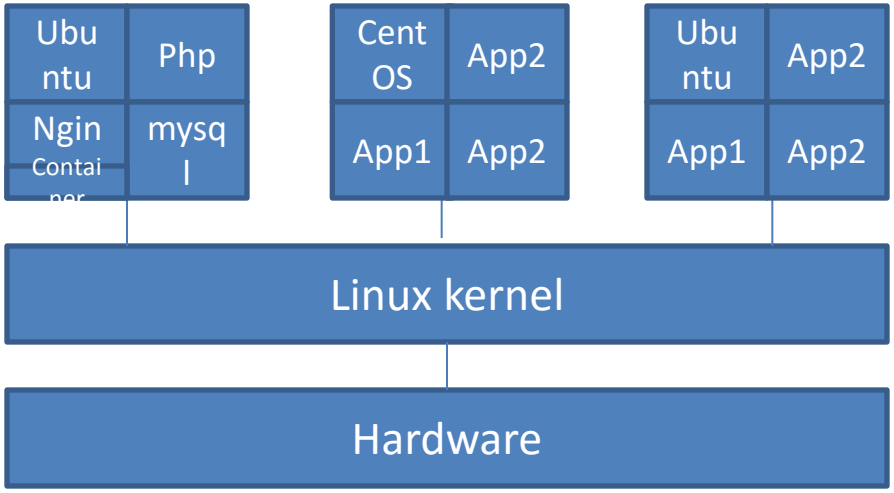
https://en.wikipedia.org/wiki/Input%E2%80%93output_memory_management_unit#/media/File:MMU_and_IOMMU.svg

LXC: OS (Linux) Container



- Same kernel, separate user-space
- Virtualize OS, not machine
- Low overhead, flexible

Docker: Application Container



- A container contain one application (process)
- Built on top of OS containers
- Even more flexible

Summary

- Virtual Machine (hardware virtualization)
  - Trap & emulate
  - Binary translation
  - Para-virtualization
  - Hardware support for virtualization
- Containers
  - OS container: same kernel, different user-space
  - App container: same kernel, per-process space