



Hypervisors

COSC349—Cloud Computing Architecture

David Evers

Learning objectives

- Define the term **hypervisor**
- Explain the basic architecture of **Xen** and how it supports VMs and hardware access
- Understand why **security** of the hypervisor is critical
- Illustrate benefits hypervisors can facilitate, such as **VM migration** and **high availability**

Hypervisors: low level machine managers

- **Hypervisors** are typically the lowest level of software
 - May be called a **virtual machine monitor** (VMM)—it runs VMs
 - Generally do not offer complete OS functionality...
 - ... just enough functionality to isolate VMs
- Historically, divided into two types (x86 examples shown):
 - **Type 1**—runs directly on computer's hardware
 - VMware ESX/ESXi; Microsoft Hyper-V; Xen
 - **Type 2**—runs as a process within existing operating system
 - VMware Workstation; Parallels for Mac; Oracle VirtualBox
 - ... although some, like Linux KVM, have aspects of both types

Typical hypervisor capabilities

- Hypervisor has to manage **CPU**, **RAM** and **device I/O**
 - Device I/O typically covers disk, network, graphics, USB, ...
- For device I/O, must **prevent guests ‘breaking out’**
 - Base security approach on per-device capabilities:
 - e.g., NIC hardware might be VM-aware and can isolate functions
 - Need to be very careful about capabilities such as DMA—
 - direct memory access lets devices access memory bypassing CPU

Rise of x86 servers and consolidation needs

- Early x86 computers ran a few desktop applications
 - Word+Excel+... on Windows 95 probably maxed-out resources
 - Server work used expensive **server-class computer hardware**
- PC computing power increased—could run servers
 - Isolating servers led to piles of under-utilised machines
- **x86 virtualisation** was appealing to consolidate servers
 - However x86 didn't support virtualisation easily
 - x86 virtualisation surged mid-2000s thanks to the **Xen hypervisor**

History: Xen project was a means to an end

- ‘XenoServers’ research at University of Cambridge
 - Project wanted to **allow computing resources to migrate**
 - So if running a Quake server, server could migrate near players
 - (Reducing network latency significantly improves responsiveness.)
- But the project needed a means to **migrate servers**
 - VMware Workstation provided necessary features ...
 - ... but was closed-source commercial software and expensive
- Research team realised that together they had expertise to **build a new hypervisor** themselves

“Xen and the art of Virtualisation”

- Research paper that shared **Xen project** initially
 - Published at SOSP 2003—a top academic OS conference
- Xen (then) required **paravirtualised OSs** as VMs
 - (We will discuss paravirtualisation in a lecture soon...)
 - Demonstrated on both Linux (XenoLinux) and Windows XP
- Figure 3 shows Xen at worst 92% of native Linux speed
 - ... also how much faster Xen was than VMware Workstation 3.2
 - Showed that a single, commodity server could **run 100 VMs!**

Xen's success involved many factors

- Great team that had just the right types of knowledge:
 - Expertise in high performance OS **memory management**
 - Expertise in lock-free data structures—**concurrency support**
 - Connection to **Microsoft Research** (Windows XP source code)
- Also great timing in terms of components required:
 - VMware Workstation alleviated the most pressing needs
 - Commodity PCs had gained sufficient RAM capacity for VMs
 - Linux network bridge support facilitated VM networking

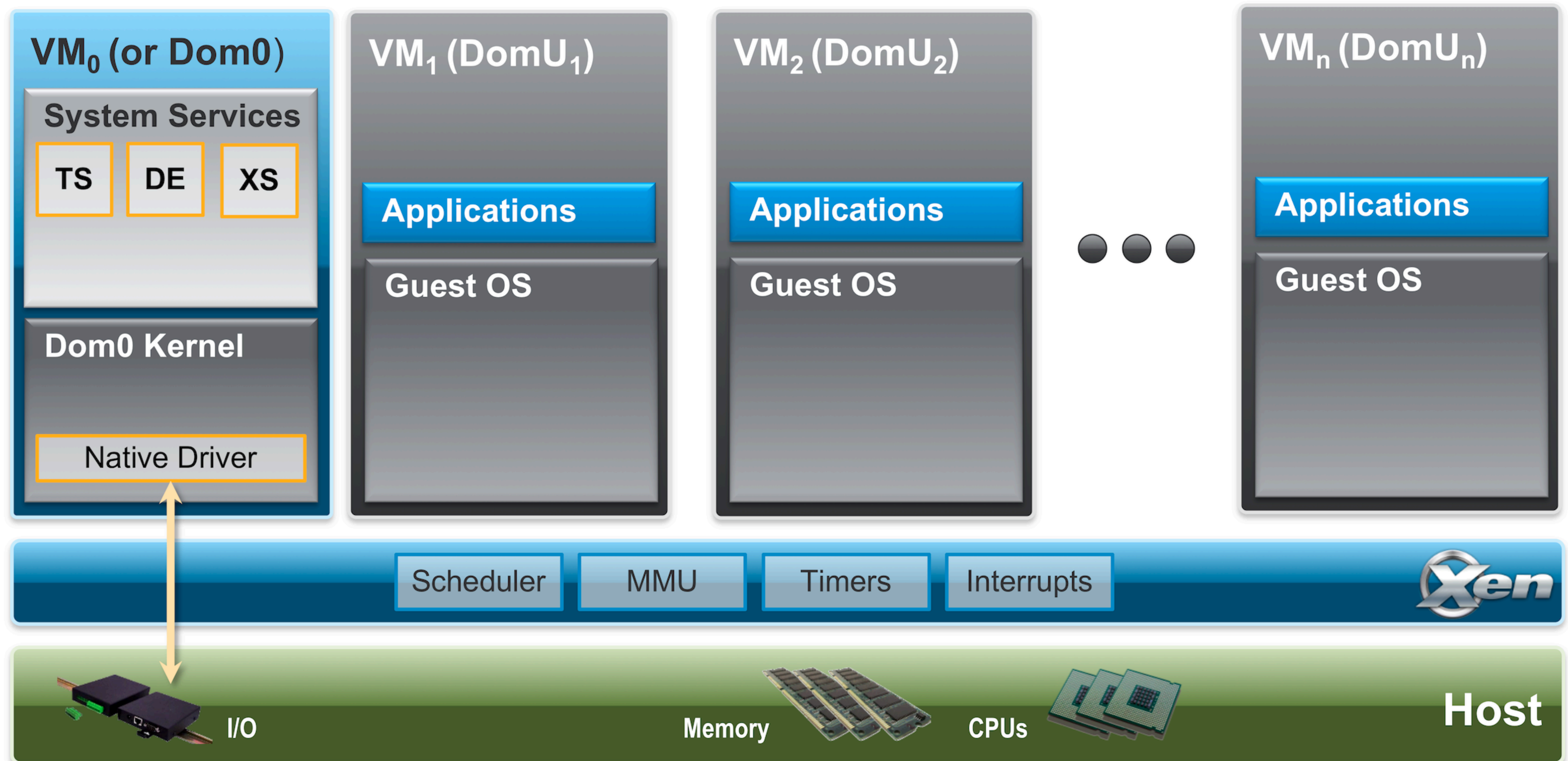
Xen and Amazon EC2

- Unknown to the Cambridge people, **Amazon noticed**
 - Amazon were running a successful global bookstore
 - Needed **distribution of computing**; had great engineers
- Xen gave Amazon **great system at ideal price**
 - Alternative would have been VMware / IBM costs
- Xen founders were not left out of pocket though:
 - Had formed XenSource company to provide Xen support
 - XenSource bought by Citrix & boosts desktop virtualisation
 - e.g., the Otago student desktop...

Xen hypervisor approach to a complete OS

- Xen team needed to conserve developer resources
 - Reuse an existing operating system to manage hardware
 - Described as XenLinux in the SOSP paper
 - **Uses existing Linux device drivers** to control hardware
- Pragmatic approach: Xen divides host into **domains**
 - However **Dom0** is special: the Linux that runs actual hardware
 - Then the **DomUs** can make **hypercalls** to access hardware
 - ... but Xen hypervisor delegates hardware interaction to Dom0

Illustration of the Xen architecture



Security considerations for hypervisors

- Users expect isolation, as if on **separate computers**
 - Yet clearly aggregation necessarily means **resource sharing**
 - Also, risk whenever there is control interaction with hypervisor
 - Hypervisor management commands have caused security failures
- Hypervisor must thus have a **small attack surface area**
 - OS kernels have large attack areas, and are hard to secure
- **Hyperjacking** is the term used to describe taking over (hijacking) the hypervisor from within a guest process

Real hypervisor security flaw—VENOM

- **VENOM** (Virtualized Environment Neglected Operations Manipulation)—CVE-2015-3456
- Problem with the floppy disk controller in QEMU
 - ... but VirtualBox, Xen and KVM also used QEMU's code
- Basically, guest accesses 'floppy drive' via 'I/O port'
 - QEMU driver keeps track of floppy drive commands in a buffer
 - ... but specially crafted requests could **overflow buffer**
 - Malicious VM can then take control of QEMU system

Live migration of VMs—a Xen functionality

- **Live migration**—move running VM to another host
 - A good demonstration that your hypervisor is efficient!
 - ‘Live’ usually means no detectable downtime
 - *i.e.*, cannot pause VM, copy VM state, resume VM on new host
 - Repeatedly stream memory updates until can do switch-over
- Requirements of physical hosts supporting migration:
 - NICs are receiving the **same MAC address**
 - Simplest for **storage to be network-based** (e.g., iSCSI / NFS)

High availability (HA)

- **High availability** (HA) means VMs are **robust to failure**:
 - VMs may restart on the same host (e.g., given typical OS crash)
 - Or host may fail: ensure VMs can continue on a different host
- Live migration & high availability have common needs
 - HA requires VM that might take over being up-to-date
 - Similar to a persistent live migration from leader to follower
- Care needed to ensure **safe and consistent failover**

Live migration of VMs & cloud computing

- Cloud providers make great use of **consolidation**
 - Difficult to guess what providers' infrastructure actually is
- Cloud providers **typically avoid migrating** VMs though
 - Very useful to have the ability, e.g., for repairing hardware
 - ... however the spike in network use is significant
 - Network spike greater still, if not already using network storage
- Providers don't want to over-engineer cloud network