Norwegian University of Science and Technology

Operating Systems

Lecture overview and Q&A Session 10 – 28.3.2022

Michael Engel

Lectures 17 and 18

Virtual machines and microkernels

- OS architectures and history, monolithic kernels
- Microkernels, exokernels
- Virtualization

Cloud operating systems

- The cloud and virtualization
- Cloud service models
- Virtualization details: containers
- CPU, memory and I/O virtualization

OS architectures and history, monolithic kernels

Library operating systems

- Operating systems developed from a library of useful common functions for applications

Monolithic operating systems

- Monolithic systems were developed to enable multiple users to use a single computer: **multiprogramming**
- Features provided:
  - **Control** of hardware and software
  - **Isolation** between processes
  - Notion of **users**
  - **Accounting** of compute time, memory, storage, …
  - **Privilege system** to protect the OS agains applications

Microkernels and exokernels

Objectives

- Minimize functionality running in privileged CPU mode
- Isolate components in non privileged mode
- Syscalls and communication use message passing (IPC)
- Reduced functionality in the microkernel: less code!
  - fewer bugs, possibility of formal verification (seL4)
  - reduction of the Trusted Computing Base (TCB) size
- First-generation microkernels (e.g. Mach): large and slow
- Second-generation microkernels (e.g. L4): Optimization of IPC
  - *"A concept is tolerated inside of a microkernel only if moving it outside of the kernel would prevent the implementation of functionality required in the system"* (Liedtke)
- Exokernels simplify OS even further: *only resource partitioning!*

Virtualization

Objective

- Isolate & multiplex resources below the OS layer
  - Allow sharing of hardware between **guest OSs**
- Virtual machines (VMs) on system level virtualize…
  - processor(s), main memory, mass storage, peripherals
- Virtual machine monitor (VMM) or hypervisor
  - software component that provides the VM abstraction
- Old technology: introduced in 1960s by IBM 360 mainframe
- Type 1 vs type 2 hypervisors
  - run on bare hardware (e.g. Xen) vs. on top of an OS (e.g. KVM on Linux)
- Paravirtualization: VMM-aware optimization of guest kernels

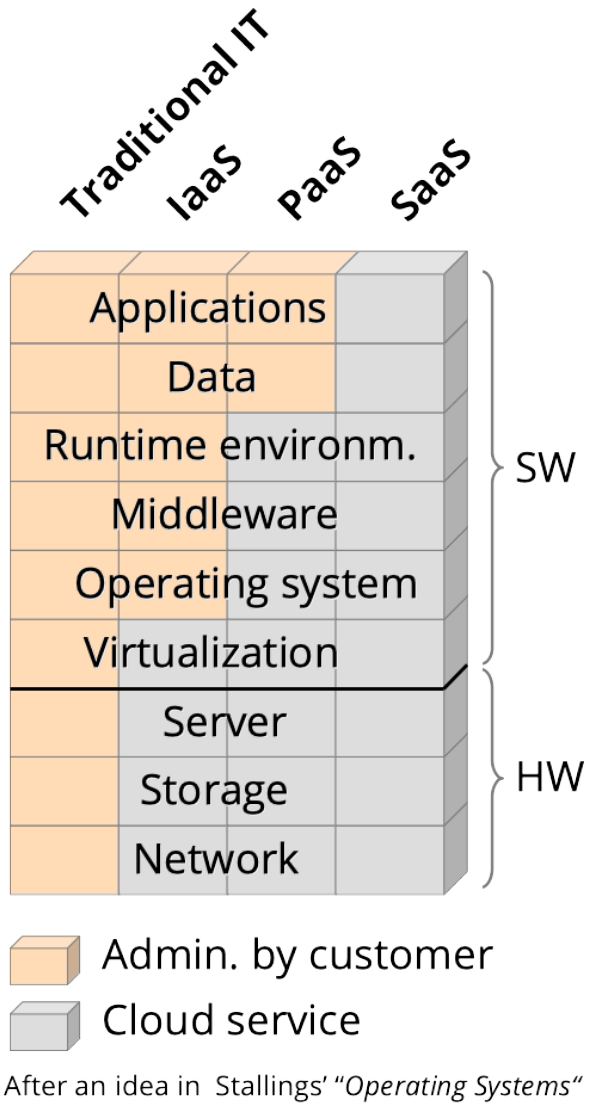
The cloud and virtualization

Objective

- Move compute and storage resources onto remote servers
- Enable customers to rent compute and storage capacities on demand
- **Cloud properties**
  - "self service" on demand, high throughput net access, availability of a resource pool, fast adaptivity, measurable (and billable…) service
- Hardware-supported virtualization basis for cloud systems
  - Enable fast installation/start of cloud OS
  - No installation of physical hardware required
  - Additional features: migration, checkpointing, resource allocation and monitoring

Cloud service models

- **SaaS** – Software-as-a-Service
  - Cloud service provider offers a complete application
  - e.g. Office365, Gmail, Zoom
- **PaaS** – Platform-as-a-Service
  - Execution environment for applications including the OS and runtime environment (depending on the programming language)
  - e.g. Engine Yard, Google App Engine
- **IaaS** – Infrastructure-as-a-Service
  - (Virtual) hardware platform
  - e.g. Amazon EC2, Microsoft Azure



After an idea in Stallings' "*Operating Systems*"

Virtualization details: containers

Idea

- Virtualization of a single OS kernel
  - Containers share a kernel
  - Libraries and system processes can be different
- The virtualization component takes care of...
  - **Separate views**, e.g. each container sees only its "own" processes
  - **Resource partitioning**, e.g. CPU time
  - **Efficient sharing**, e.g. avoid duplication of files
- Examples:
  - Docker – uses Linux cgroups features to create containers
  - Solaris Zones
  - FreeBSD Jails

CPU, memory and I/O virtualization

CPU virtualization

- Emulation+multiplexing: flexible but slow (qemu, bochs, MAME)
- Virtualization criteria by Popek and Goldberg
  - "Sensitive" instructions need to be intercepted
- Optimization: hardware support for virtualization

Memory virtualization

- Virtual memory problem: double virtual address translation
- Solutions: shadow page tables and nested page tables

I/O virtualization

- I/O device emulation + multiplexing can be complex & slow
- Alternative: device passthrough / PCIe I/O virtualization
  - Security for main memory access ensured by I/O MMU

Overview Theoretical Exercise 8

Kernels, virtualization, hypervisor, cloud, embedded

Why?

- All of these topics are only presented in an overview form for this course
- Each topic could easily fill a course of its own
  - …no time for this, unfortunately