Institute of Computer Science
Department of Mathematics and Computer Science

Freie Universität Berlin

Algorithms and Programming IV

Cloud Computing: Background Technologies (21-2)

Summer Term 2021 | 25.06.2021

Barry Linnert

Essential Concepts

Virtualization:

- Cloud computing virtualizes systems by pooling and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility.

Abstraction:

- Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren't specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous.

Sosinsky, B. (2010). *Cloud computing bible* (Vol. 762). John Wiley & Sons.

Freie Universität Berlin

Cloud Computing: Background Technologies

VIRTUALIZATION

Virtualization

- Resources can be detached from their physical nature by means of *virtualization.*
- Virtualized resources for the cloud include servers, data stores, networks, and software.
- The basic idea is to pool physical resources, make them transparent, and manage them as a whole.
- Pools provide transparent access to resources.

Benefits for Providers

Resource usage: Load balancing towards full capacity of physical servers

Management: Resource pool management can be automated

Consolidation: Reducing the number of physical servers, increasing efficiency and cost effectiveness

Energy consumption: From consolidation

Less space required: From consolidation

Emergency planning: Moving virtual machines from one resource pool to another

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Benefits for Customers

Dynamic behaviour: Resource pools can easily satisfy almost all requests dynamically—even when requests are peaking

Availability: Automated fault tolerance procedures make services highly available

Access: Isolation of virtual machines make it possible to safely delegate management functionality to the customer

Emancipation: Customers can purchase IT capabilities from a self-service portal, without humans in the loop

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Drawbacks

Cost of abstraction: The abstraction layer requires resources; however, modern virtualisation techniques make such a cost more and more negligible

System management: The physical infrastructure grows in complexity; automation of procedures and availability of management tools requires however much less staff in the overall

Concepts of Virtualization

OS virtualization: Applications run within containers layered upon the host OS.

Platform virtualization: OS & applications run in a virtual environment, based on a VM monitor or hypervisor.

Storage virtualization: Dynamically scalable storage space as a service.

Network virtualization: Virtual IP addresses, Virtual LAN (VLAN).

Application virtualization: Software sales model for centralized management and distribution of applications through the network.

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Hypervisor

Also called Virtual Machine Monitor (VMM) or Virtualization Manager.

It is a software program or part of the code in firmware that manages either multiple OS or multiple instances of the same OS on a single computer system.

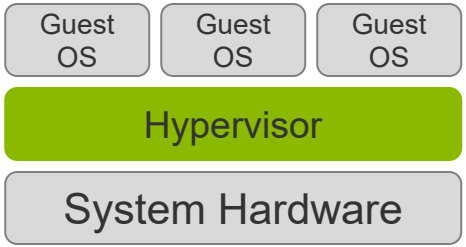
What is controlled by the hypervisor:

- CPU
- Memory
- Other resources required by the operating system

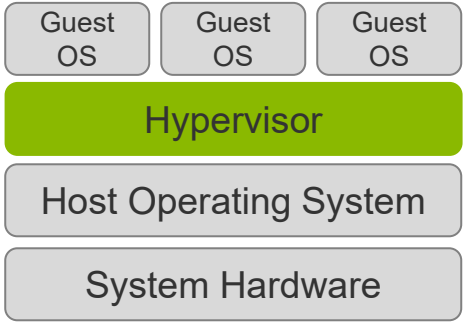
It validates all the guest-issued CPU instructions and manages any executed code that requires additional privileges.

Types of Hypervisor

Type 1 hypervisors are those that run directly on the system hardware and offer a higher level of virtualization efficiency and security.



Type 2 hypervisors are those that run on a host operating system that provides virtualization services, such as I/O device support and memory management.

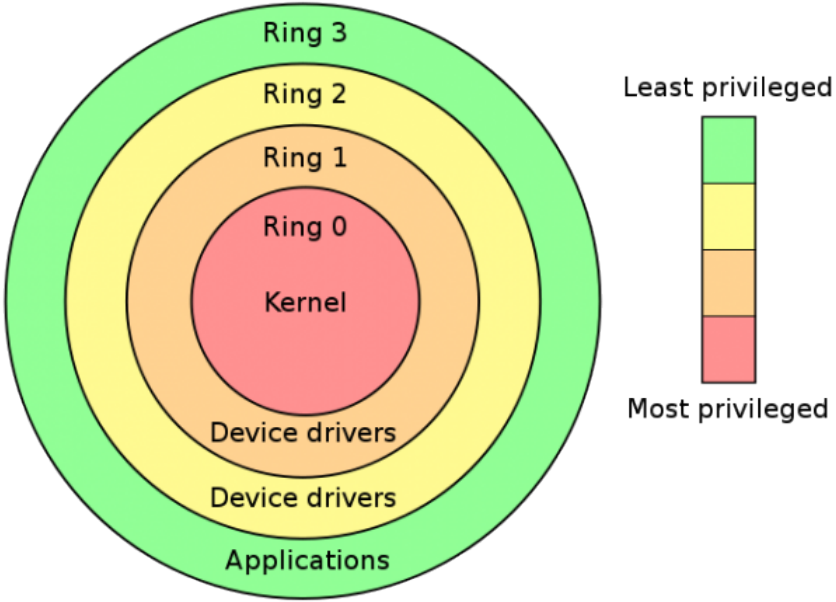


Needed Concept: Protection Rings

- ***Concept:*** Mechanism to set the layers of privilege used to protect the data and the functionality

- ***Ring 0 (kernel):*** Can execute any CPU instruction and reference any memory address

- ***Ring 3 (user):*** The access to hardware and memory reference needs to be arbitrated

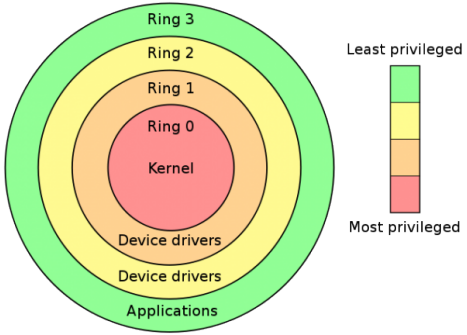


Emulation

VM emulates/simulates complete hardware. Unmodified guest OS for a different PC can run in this emulation. It is possible to emulate one architecture in a completely different one.

What runs where?

- Ring 0: run the host
- Ring 3: run the guest OS (as applications)



Pros: guest OS doesn't need modifications

Cons: very slow because of the not native execution

Example: Virtual PC for Mac

Full virtualization

Host OS emulates a hardware layer for each guest OS.

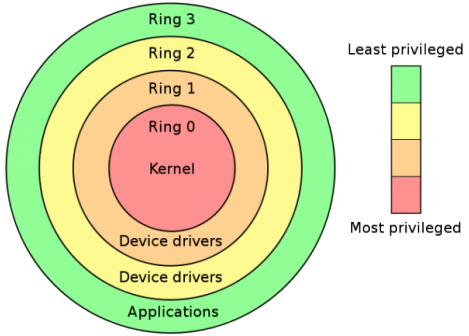
What runs where?

- Ring 0: run the guest OS privileged operations
- Hypervisor provides CPU emulation to handle CPU operations

Pros: Stability, guest OS doesn't need modifications

Cons: System resources, time performance

Examples: VMWare, Parallels

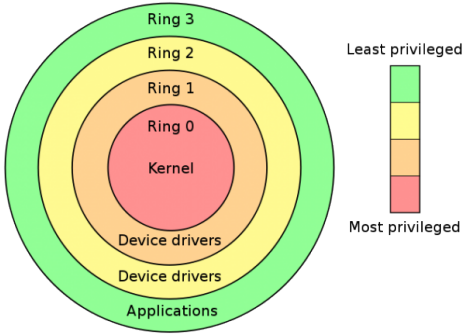


Paravirtualization

The hypervisor interfaces the hardware to all OS's (host and guest). Guests OS are modified to run on the hypervisor which performs all tasks on behalf of the guest system. Replacement of privileged operations (ring 0) with calls to the hypervisor.

What runs where?

- Ring 0: run an hypervisor
- Ring 1: run the guest OS
- Ring 3: run the applications



Pros: Stability and performance is very good and overhead is very low

Cons: Not easy to implement and to maintain because OS needs to be adapted

Example: VMWare

Hardware-Assisted Virtualization

Hardware provides support to run instructions independently for each OS.

What runs where?

- Ring 0: run the guest OS privileged operations
- Latest generation CPUs provide built-in features to run unmodified guest OS without overhead

Pros

- No need to patch the guest OS
- Most optimal performance
- Great stability

Cons

- Need to be supported by the hardware

Example: Linux KVM, Virtual Box

Cloud Computing: Background Technologies

ABSTRACTION

Service-Oriented Architectures

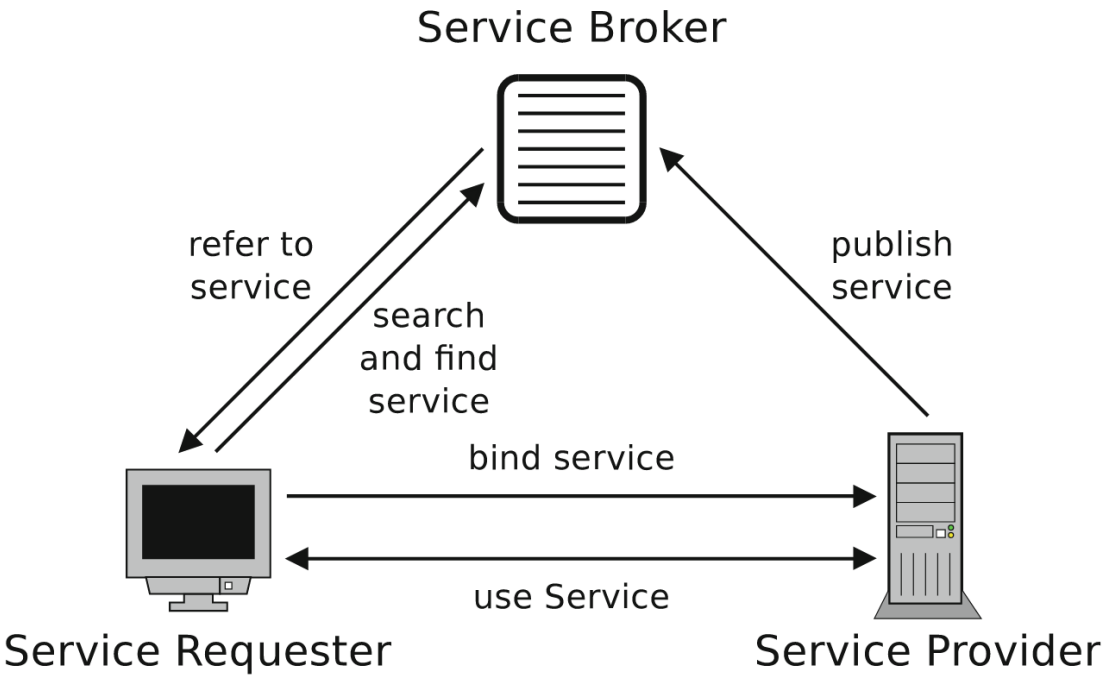
An SOA is a system architecture that represents varied, different, and possibly incompatible methods or applications as re-usable and openly accessible services and thus allows to use and re-use them in a platform- or language-independent manner.

Typical properties of an SOA are:

- It consists of distributed components, i.e. the services.
- Heterogeneous service consumers and service providers are interoperable across platforms.
- Services are loosely coupled and will be bound dynamically at runtime.

Marcel, C. B., Marcel, K., Mimis, J., & Tai, S. (2011). Cloud Computing: Web-based Dynamic IT Services.

SOA Participants and Actions



Marcel, C. B., Marcel, K., Mimis, J., & Tai, S. (2011). Cloud Computing: Web-based Dynamic IT Services.

Defining a Web Service

Generic definition: *Any application accessible to other applications over the Web.*

Definition of the UDDI consortium: *Web services are self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces.* (http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf)

Definition of the W3C: *A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.* (http://www.w3.org/TR/ws-arch/)

Characteristics of a Web Service

A web service interface generally consists of a collection of operations that can be used by a client over the Internet. The operations in a web service may be provided by a variety of different resources, for example, programs, objects, or databases.

The key characteristic of (most) web services is that they can process XML-formatted SOAP messages. An alternative is the REST approach.

Each web service uses its own service description to deal with the service-specific characteristics of the messages it receives.

Simple Object Access Protocol (SOAP)

SOAP is designed to enable both client-server and asynchronous interaction over the Internet. It defines a scheme for using XML to represent the contents of request and reply messages as well as a scheme for the communication of documents.

It is used for information exchange and RPC, usually (but not necessarily) over HTTP.

(Very) basic SOAP architecture