Cloud Information Systems

Prof. Dr. Viktor Leis

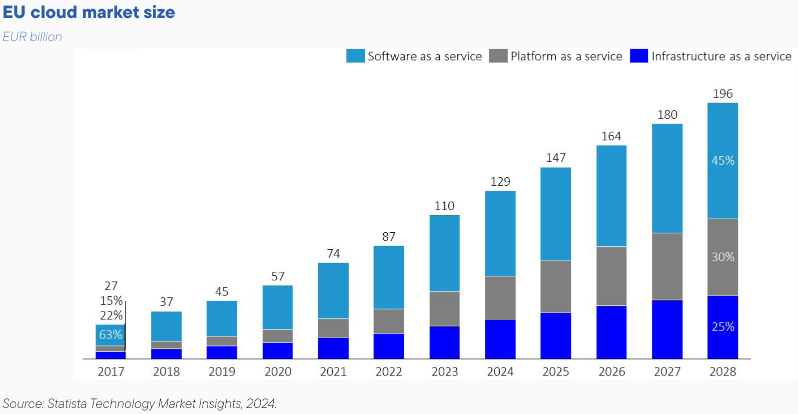
Chair for Decentralized Information Systems and Data Management

Foundations

Everything as a Service

- Infrastructure as a Service: renting hardware (Amazon EC2), storage (S3)
- Platform as a Service: development and deployment framework (Google App Engine)
- Software as a Service: Webapp (Gmail, Office 356)

EU Cloud Market



source: `https://commission.europa.eu/document/download/ec1409c1-d4b4-4882-8bdd-3519f86bbb92_en`

The Cloud From a Provider's Perspective

- Consider building a business around a flexible, pay-as-you-go computing service where a customer can change their allocation of resources at any time
- What do you need?
    - data center
    - network
    - ways of adjusting resources (computing, storage)
    - techniques that ensure that customers ("tentants") do not interfere with each other
    - billing and accounting mechanism

Foundations

Data Centers

Economic Factors Motivating The Move To Cloud

in pre-cloud IT

- cheap computers encouraged every department to acquire and operate its own server or set of servers
- as the number of servers grew, each department hired staff to operate them
- an organization faced overlap and duplication of expertise across many departments one IT exec quipped: “Cheap computers have turned into a major expense.”

Data Centers

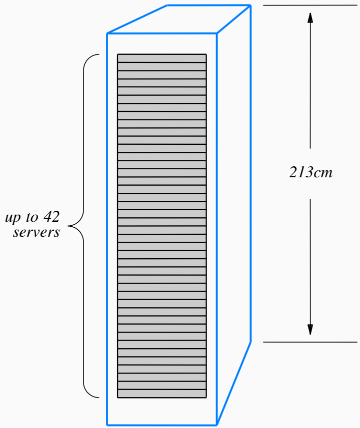
- centralized compute facilities
- maybe "compute center" would be a better term, but has other historical meaning

Google Data Center: `https://youtu.be/XZmGGAbHqa0?t=168`

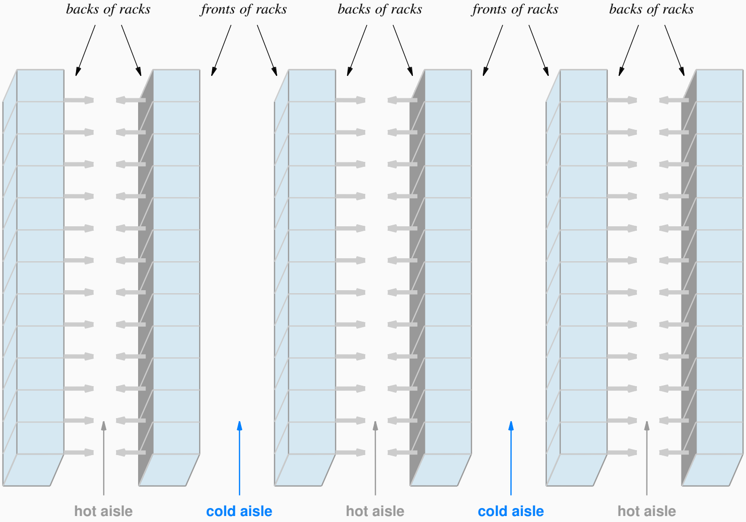
Hetzner Data Center: `https://www.youtube.com/watch?v=5eo8nz_niiM`

Racks

- allows squeezing many servers into a small space
- standardized sizes with servers using one or height units (1U, 2U)
- connections at the back of each server: power, network cables
- pod: set of racks servers, storage facilities, network switches+connections, and power connections



Cooling Many Racks



500 Watt/server * 40 servers/rack * 12 racks/row * 6 rows $\approx$ 1.4MW

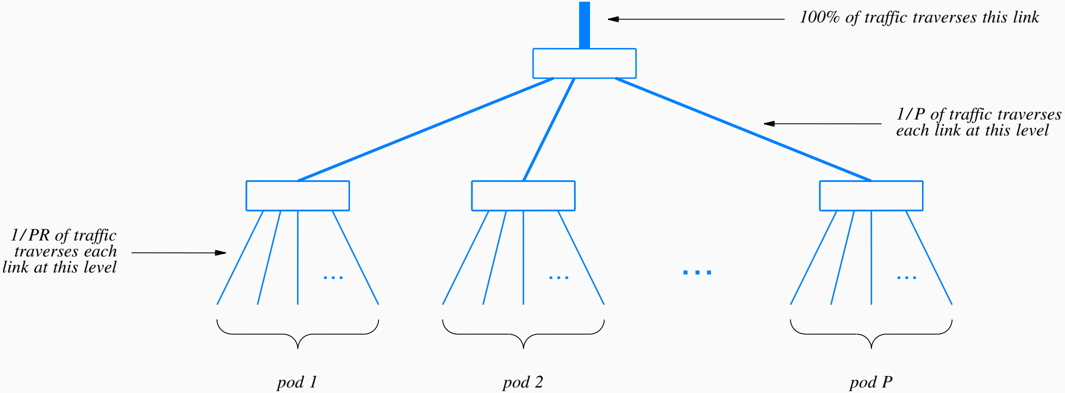
Foundations

Networking

Networking

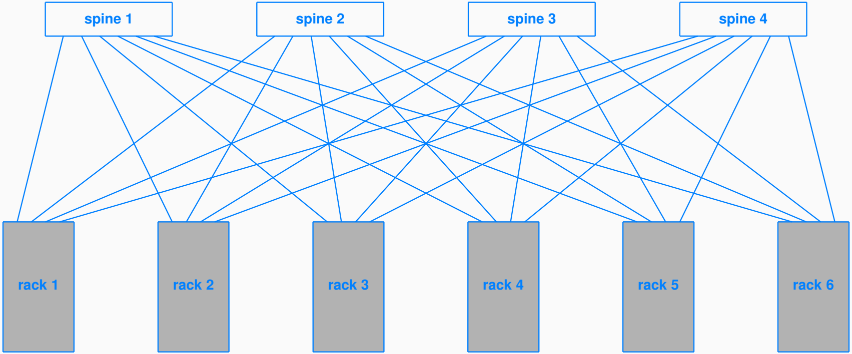
- large data centers use Ethernet networks
- 10 Gbit is standard, 40 Gbit is common, 100 Gbit available
- each rack has a *Top-of-Rack (ToR)* switch
- additional switches to connect ToR switches

Fat Tree Topology



- links between ToR switches and other switches should be very fast
- root switch is single point of failure

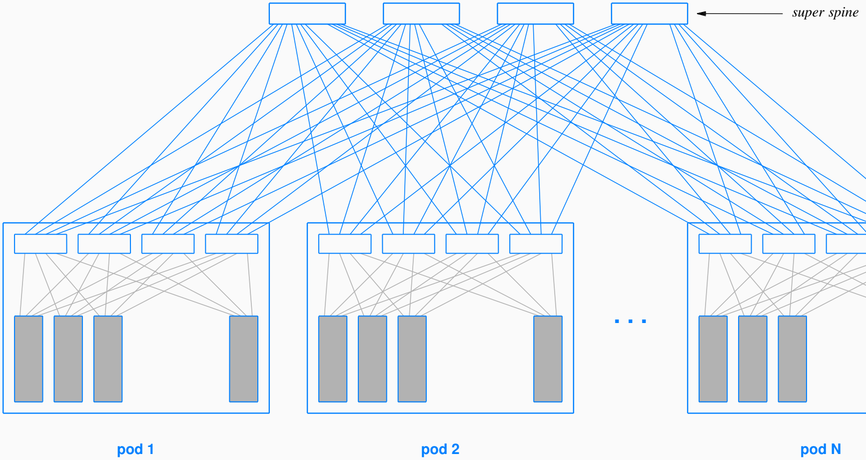
Leaf-Spine Topology



- can survive death of spine switch
- can be scaled to more racks to some extent

Leaf-Spine With A Superspine Level

- switches have a limited number of connections
- second level solves this, enables very large data centers

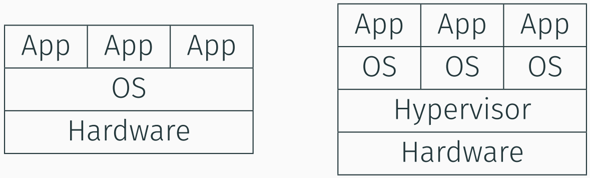


Foundations

Virtualization

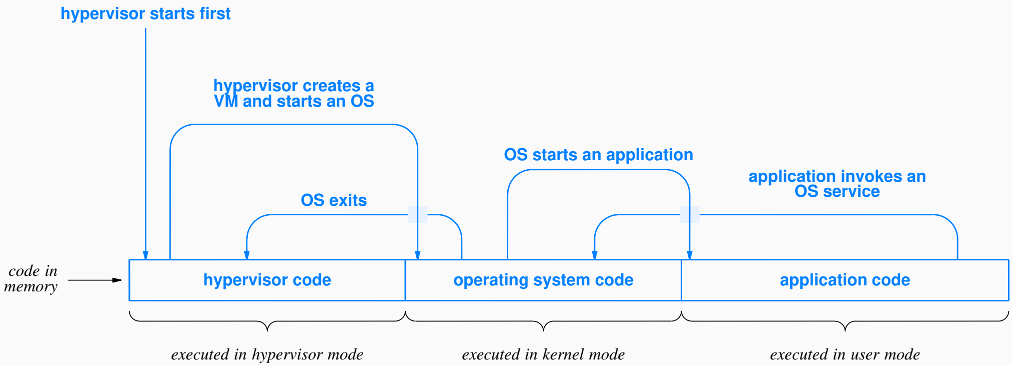
Virtual Machines

- hypervisors (xen, KVM) allow virtualizing hardware
- multiple VMs on one physical machine, may be different OSs
- hypervisors emulates a specific complete computer (e.g., x86, ENA NIC)



How Does Virtualization Work?

- modern CPUs have hardware support for virtualization (Intel VT-x, AMD-V)
- three levels of privilege: hypervisor, kernel, user mode



What About I/O Devices?

- hypervisor provides virtual hardware devices for each VM/guest OS
- when an operating system attempts to access a device, control passes to the hypervisor, which invokes virtual device software
- emulated device does not have to be "real" one (see virtio standard)
- with fast devices, emulated devices can be slow
- modern cloud vendors are transitioning to special hardware (e.g., SSDs) that have built-in virtualization, so that hypervisor is not involved any more on the "data path" (e.g., every read operation)

Implications

- virtualization isolates tenants on very low level
- security isolation seems to be good: no major breach has been reported
- performance isolation is not perfect: performance may vary depending on what neighbor tenants are doing

How To Split Hardware Resources?

- fair share (static)
- competitive (dynamic)
- token-bucket (overcommitting)

Foundations

Storage

Virtual Disks

- a VM can access a local disk, but
  - if instance is stopped disk content is generally gone
  - no physical access to disk
- solution: virtual remote disk
- can be provisioned on demand
- elastic pricing
- allows stopping a VM without losing the data
- also allows changing instance type (stop VM, start on other instance with same virtual disk)

How Are Virtual Disk Implemented?

- virtual disks are generally connected through the network
- option 1: Network Attached Storage (NAS) reuses existing network
- option 2: Storage Area Network (SAN) has separate network just for virtual storage
- SAN is more expensive but has more predictable performance

Foundations

Other Compute Abstractions

Containers

- virtual machines provide good isolation but are relatively heavyweight (e.g., boot time, memory)
- containers are lightweight, isolated environments
- based on extended process isolation facilities of operating systems
- Docker is implemented using Linux features such as
    - *namespaces* provide own process ID space, network stack, mount points, user management
    - *cgroups* allow limiting CPU, memory, I/O
    - overlay file systems provide separate storage

VMs vs. Containers

containers

- lower startup times
- better memory utilization

VMs have

- better security
- better performance isolation
- and allow changing the OS

if you run a container in a public cloud, it will probably be within a VM

Functions as a Service (“Serverless”)

- both containers and VMs are still fairly low level
- e.g., someone has to decide when to start and stop them
- public cloud vendors offer a service to upload code (a function):
    - exposed through http
    - launched and scaled automatically by cloud provider
- of course serverless is an oxymoron (in the end everything runs on a server)
- in practice FaaS products have major limitations (more later)

Unikernels

- one of the main tasks and major source of complexity of an operating systems is to isolate processes from each other
- e.g., processes of different users should not be able to read each other's memory
- in the cloud we often run only one service per VM
- VMs are isolated from each other through the hypervisor
- we don't necessary require process isolation
- unikernels are simple single-address space operating systems
- public clouds have only limited set of devices to support
- examples: unikraft, osv, nanos

Foundations

Data Center Total Cost of Ownership

Total Cost of Ownership (TCO)

data center TCO components

- Capital expenses (CAPEX): facilities, compute, storage, networking
- Operational expenses (OPEX): energy, maintenance, employees, ...

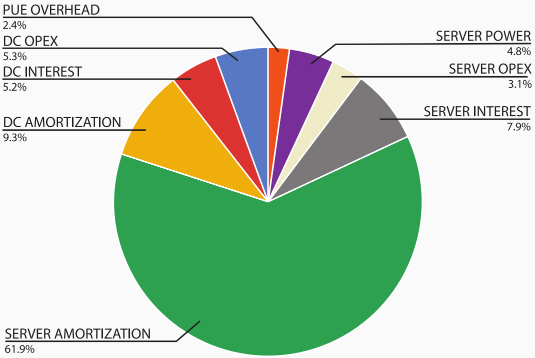
TCO = data center depreciation + data center OPEX + server depreciation + server OPEX

Data Center TCO

- power is crucial for data centers
- construction cost for big data center is around \$10/W
- all of the data center's primary components (power, cooling, and space) roughly scale linearly with watts
- a big data center might cost \$200M and consume 20MW
- a typical server may consume about 500W
- a big data center consists of 20MW : 500W = 40,000 servers
- for comparison: superMUC has $\approx$ 6,000 servers using $\approx$ 3MW
- see *The Datacenter as a Computer: Designing Warehouse-Scale Machines*, Chapter 6 for more

Typical Example

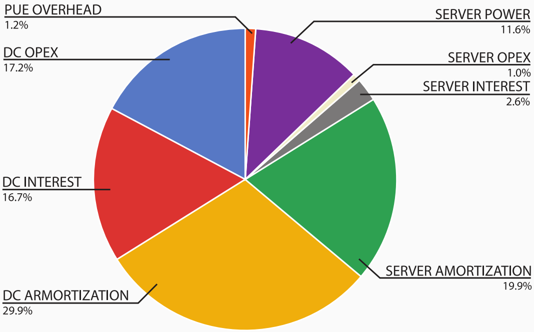
- $5,000 servers with 340W peak
- $0.067/kWh, drawing 75% on avg.
- 3 year server lifetime, 8% interest
- maintenance is 5% of Capex per year
- $10/W data center construction cost
- depreciated over 20 years, 8% interest
- data center Opex $0.04/W per month (without server power)
- power usage effectiveness (PUE) 1.5



Example 2

cheaper, faster, higher-powered server:

- consuming 600W at peak
- costing only $2,000
- electricity cost is $0.10/kWh



Example 3

Amortized Cost	Component	Sub-Components
$\sim$45%	Servers	CPU, memory, storage systems
$\sim$25%	Infrastructure	Power distribution and cooling
$\sim$15%	Power draw	Electrical utility costs
$\sim$15%	Network	Links, transit, equipment

Table 1: Guide to where costs go in the data center.

The Cost of a Cloud: Research Problems in Data Center Networks, Greenberg et al., SIGCOMM 2008

Summary

- server cost is still often dominant
- but other costs can be significant
- very cheap machines do not necessarily have the lowest TCO due to other costs
- public clouds therefore generally use single or dual-socket server *commodity* hardware
- not the cheapest option, but also not very high-end servers
- for a fair cost comparison between on-premise and cloud cost, it is necessary to consider more than just hardware cost