Cloud Computing & Big Data

PARALLEL & SCALABLE MACHINE LEARNING & DEEP LEARNING

Prof. Dr. – Ing. Morris Riedel
Associated Professor
School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland
Research Group Leader, Juelich Supercomputing Centre, Forschungszentrum Juelich, Germany

@Morris Riedel @MorrisRiedel @MorrisRiedel

LECTURE 4

Virtualization & Data Center Design

October 1, 2020
Online Lecture

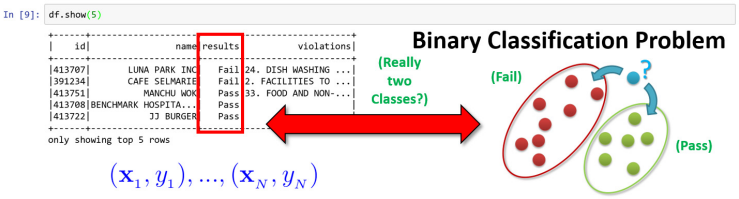
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE

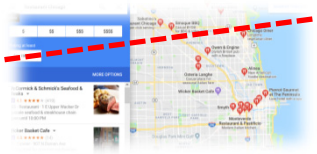
DEEP Projects

HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Review of Practical Lecture 3.1 – Building Scalable ML Pipelines with Apache Spark

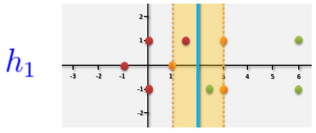


$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

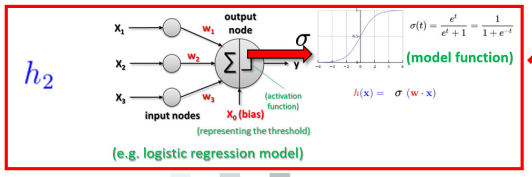


'training set' 'test set'

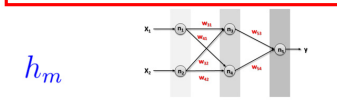
Training Examples
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
(historical records, groundtruth data, examples)



(e.g. support vector machine model)



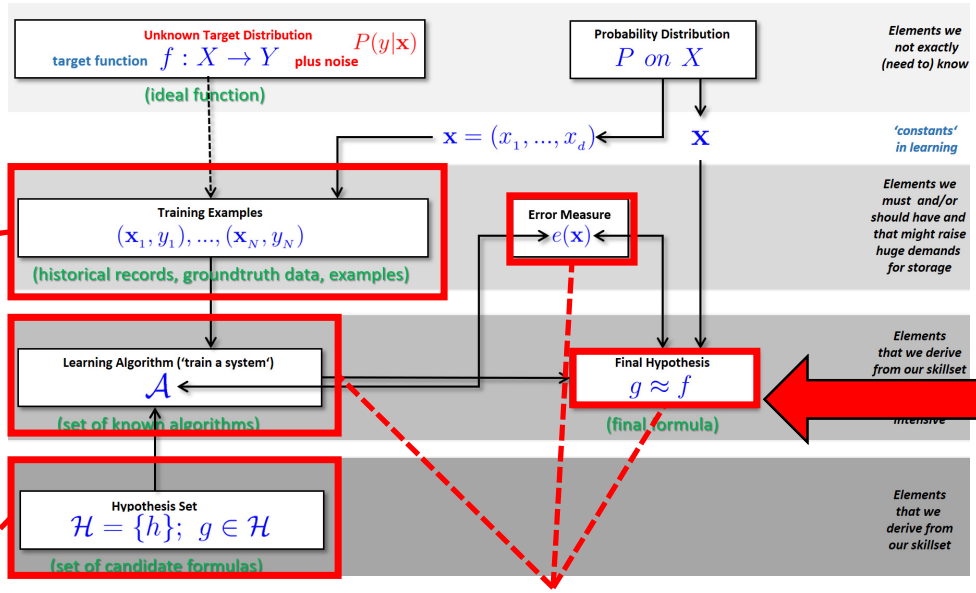
(e.g. logistic regression model)



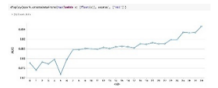
(e.g. artificial neural network model)

Learning Algorithm ('train a system')
$\mathcal{A}$

Error Measure
$e(\mathbf{x})$



(iteratively solving an optimization problem using training data and testing data)



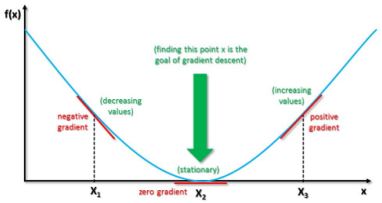
(logistic function)

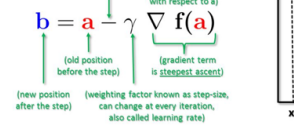
```
1 / (1 + exp(-w^T x))
```

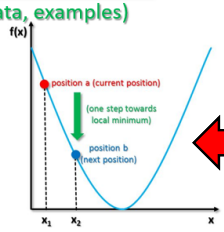
```
log(1 + exp(-yw^T x))
```

(loss function, logistic loss)

Training Examples
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
(historical records, groundtruth data, examples)

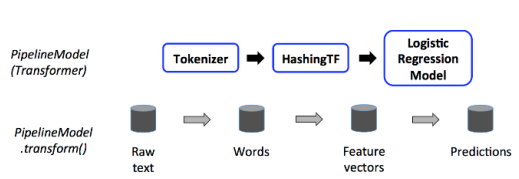


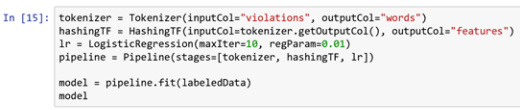




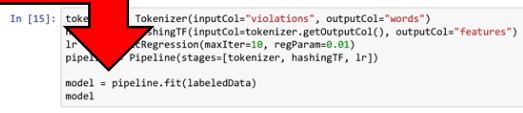
[2] www.big-data.tips, gradient descent

[1] Apache Spark

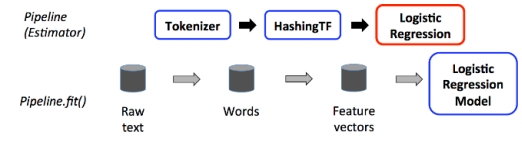




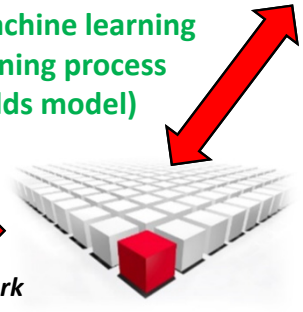
```
In [15]: tokenizer = Tokenizer(inputCol="violations", outputCol="words")
hashingTF = HashingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
lr = LogisticRegression(maxIter=10, regParam=0.01)
pipeline = Pipeline(stages=[tokenizer, hashingTF, lr])

model = pipeline.fit(labeledData)
model

PipelineModel_4ee589dd5c6759dd2297
```



```
In [15]: toke  Tokenizer(inputCol="violations", outputCol="words")
         hingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
lr       Regression(maxIter=10, regParam=0.01)
pipel    Pipeline(stages=[tokenizer, hashingTF, lr])

model = pipeline.fit(labeledData)
model

PipelineModel_4ee589dd5c6759dd2297
```



(machine learning training process builds model)



Outline of the Course

1. Cloud Computing & Big Data Introduction
2. Machine Learning Models in Clouds
3. Apache Spark for Cloud Applications
4. Virtualization & Data Center Design
5. Map-Reduce Computing Paradigm
6. Deep Learning driven by Big Data
7. Deep Learning Applications in Clouds
8. Infrastructure-As-A-Service (IAAS)
9. Platform-As-A-Service (PAAS)
10. Software-As-A-Service (SAAS)
11. Big Data Analytics & Cloud Data Mining
12. Docker & Container Management
13. OpenStack Cloud Operating System
14. Online Social Networking & Graph Databases
15. Big Data Streaming Tools & Applications
16. Epilogue

+ additional practical lectures & Webinars for our hands-on assignments in context

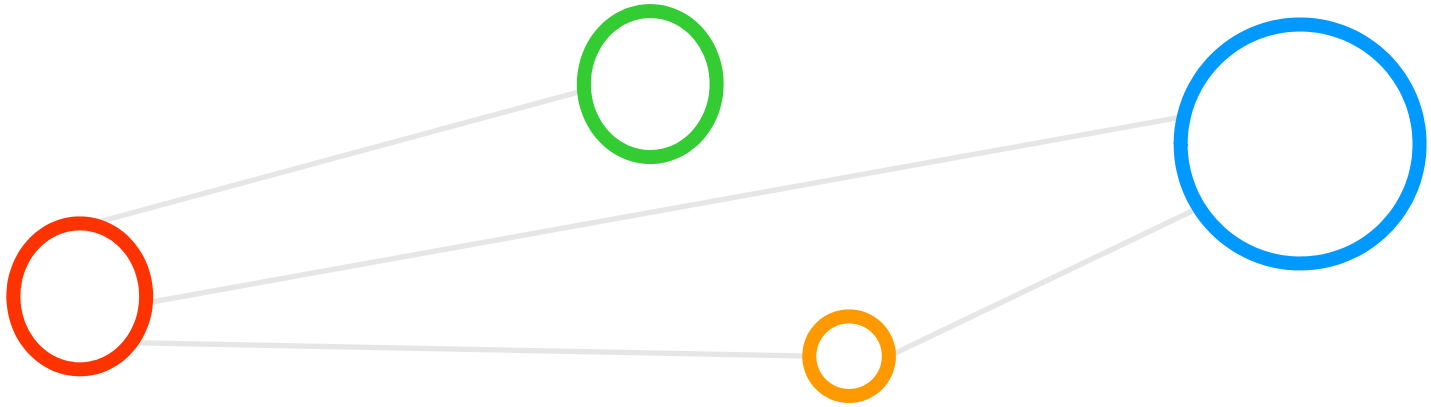
- Practical Topics
- Theoretical / Conceptual Topics

Outline

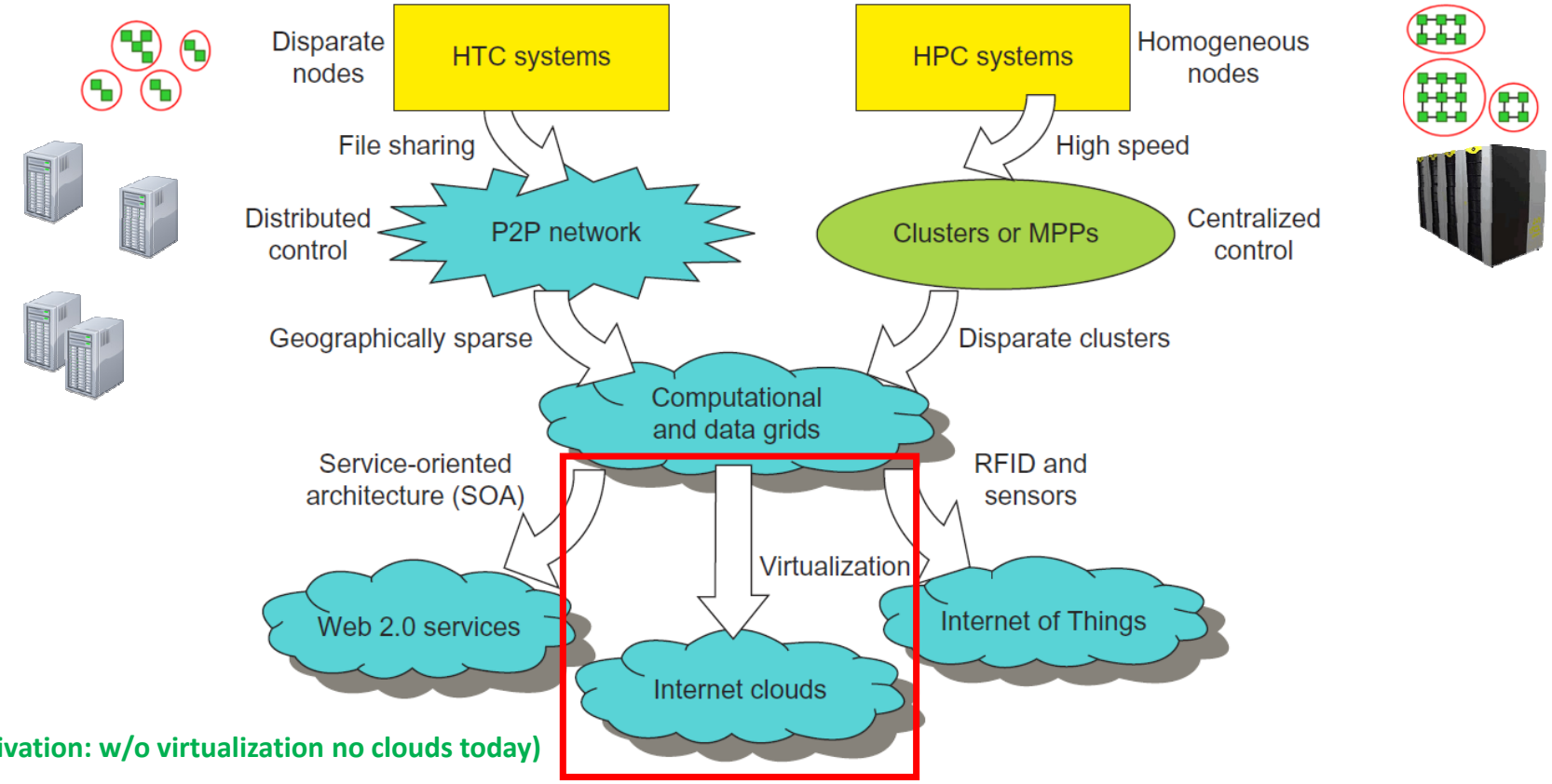
- Virtualization enabling Cloud Computing Technologies
  - Different Terminologies like Virtual Machines & Virtual Layers
  - Techniques of Migration & Virtual Appliances
  - Understanding Virtual Machine Images & Instances
  - Virtualization enabling Resource Sharing & Useful Operations
  - Virtualization Software Architectures & Virtualized Servers
- Cloud Data Center Design
  - Economics of Scale Approach & Total Cost of Ownership
  - Data Center Contruction Features & Role of System Failures
  - Building Data Center Infrastructure & Networking Aspects
  - Data Center Metrics & Scalability Requirements Analysis
  - Public/Private/Hybrid Cloud Deployment Models

- **Promises from previous lecture(s):**
- ***Lecture 1:*** **Lecture 4 provides more details about the underlying virtualization technology and its relevance for large data centers & clouds today**
- ***Lecture 1:*** **Lecture 4 provides more details about the computing paradigms HPC and HTC and their relevance to cloud computing systems today**
- ***Lecture 1:*** **Lecture 4 provides more details about storage technologies relevant for large data center designs that operate multi-user clouds today**

Virtualization enabling Cloud Computing Technologies



Cloud Computing – Evolution over time & Virtualization Dependencies



(motivation: w/o virtualization no clouds today)

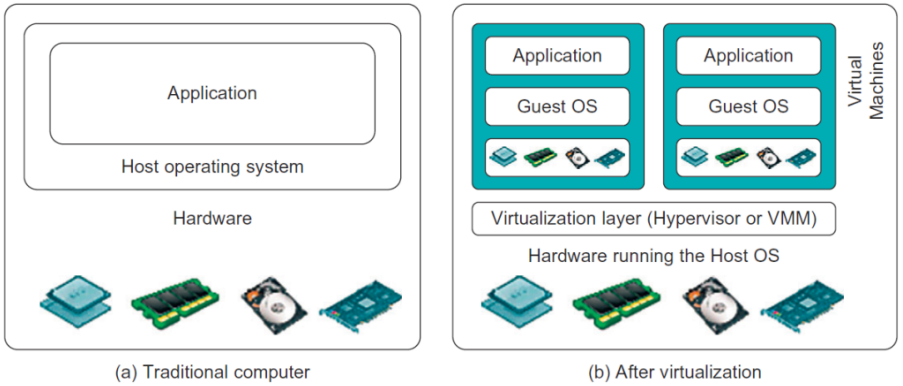
modified from [3] Distributed & Cloud Computing Book

- **Advances in virtualization make it possible to see Internet Clouds as new computing paradigm with an attractive way of serving multiple users today**

Virtualization Concepts

- Key Ideas
  - Virtualization concepts can be dated back to the 1960s
  - Purpose is to enhance resource sharing by many users
  - Improves computer performance in terms of resource utilization
  - Hardware resources (CPU, memory, I/O devices, etc.) or software resources (operating system, software libraries, etc.) are virtualized
  - Separate the hardware from the software to yield better system efficiency
- Terminologies involved
  - Virtual Machines
  - Virtual Machine Managers (VMMs) and hypervisors
  - Virtual layer virtualize shared physical hardware of a host machine into exclusive virtual resources of VMs

[4] M. Rosenblum, Reincarnation of VMs



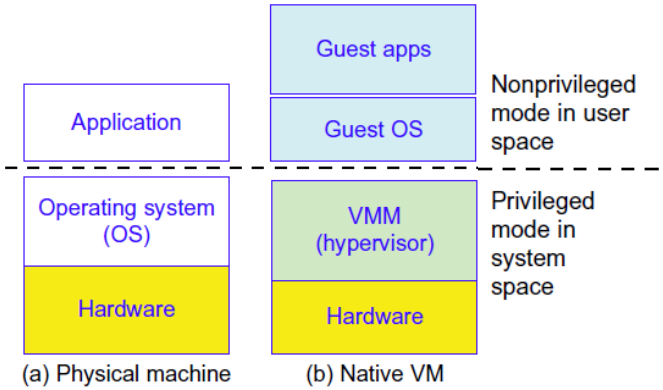
(a) Traditional computer (b) After virtualization

[3] Distributed & Cloud Computing Book

Virtual Machines

- Conventional computers
  - Built with physical hardware that represents the existing physical machine
  - Has one single operating system (OS) **(uses several privileges when running on hardware)**
  - Tightly couples application software to a specific hardware platform
  - E.g. x-86 architecture desktop running its installed Windows OS
- Native VM
  - Also known by experts as ‘bare-metal VMs‘
  - Can be provisioned for any hardware
  - VM is built with virtual resources managed by Guest OS
  - E.g. Linux System and XEN hypervisor **(user-level, no privileges)**

modified from [3] Distributed & Cloud Computing Book

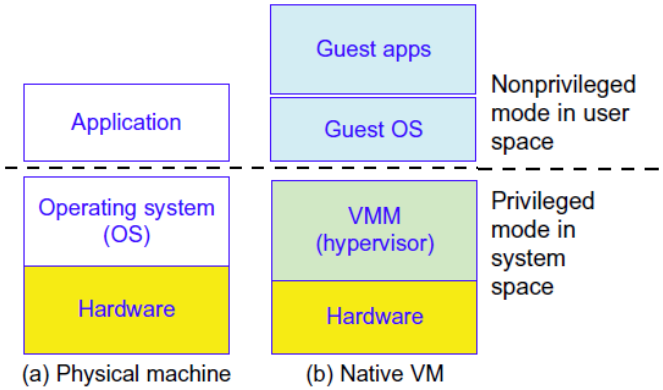


- **Virtual Machines (VMs) offer solutions to underutilized resources, application flexibility, software manageability, and security concerns on existing physical machines**
- **A Native VM uses a virtual machine manager (VMM) between the VMs and the hardware**
- **A VMM can be installed as ‘hypervisor‘ that handles bare hardware (CPU, memory, I/O, etc.)**

Virtualization Layer

- Hardware-level virtualization
  - Inserts virtual layer between real hardware & traditional operating system
  - N same/different operating systems ‘sit on same hardware at same time‘
  - VMMs manage the hardware resources of a computing system
  - E.g. one CPU hardware component is virtualized as n virtual CPU copies
- How it works
  - Each time programs access the hardware…
  - … the VMM captures the process
  - VMM thus acts as a traditional operating system

modified from [3] Distributed & Cloud Computing Book

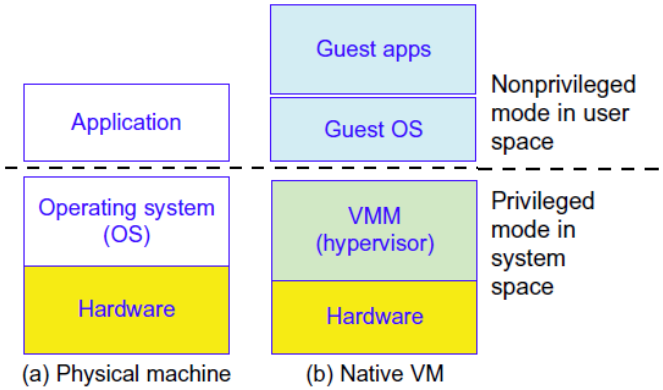


- **Virtualization software such as Virtual Machine Managers (VMMs) creates the abstraction of virtual machines by interposing a virtualization layer at various levels of a computer system**

Virtual Machine Manager

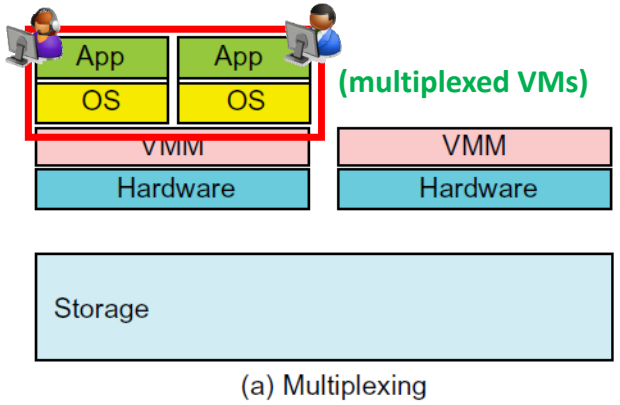
- Full virtualization
  - VMM exports a VM abstraction identical to the physical machine
  - A standard operating system can run as it would on the physical hardware
  - E.g. Windows or Linux
- VMs enhance utilization of servers
  - Avoid using more and more servers, but VMs are isolated from each other
  - Deployments of systems as a VM
  - Transparency in shared hardware
  - E.g. VMWare claims server utilization is often increased from 5 – 15 % to 60 – 80 %

modified from [3] Distributed & Cloud Computing Book

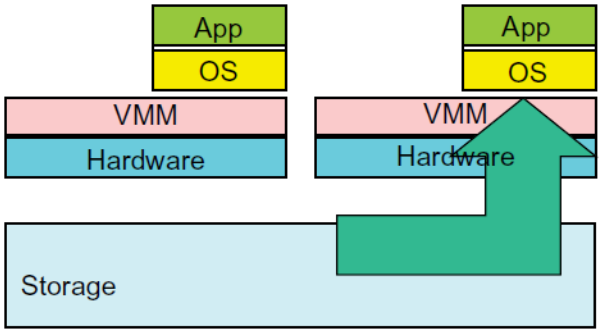


- **A Virtual Machine Manager (VMM) provides the VM abstraction to the guest operating system**
- **The VM approach offers hardware independence of the operating system and applications**

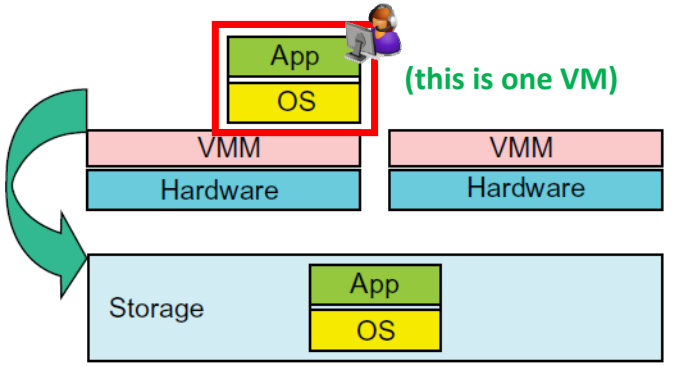
Virtual Machine Manager Operations



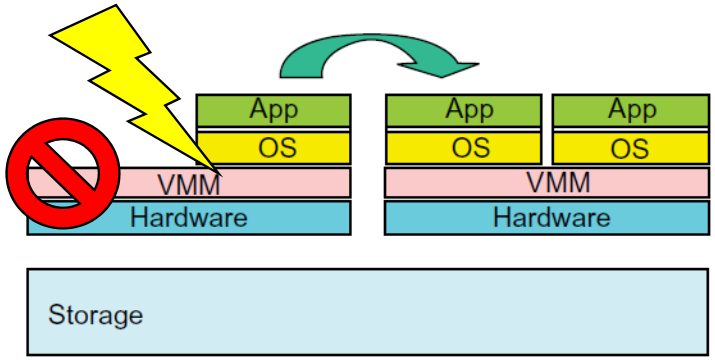
(a) Multiplexing



(c) Provision (resume)



(b) Suspension (storage)



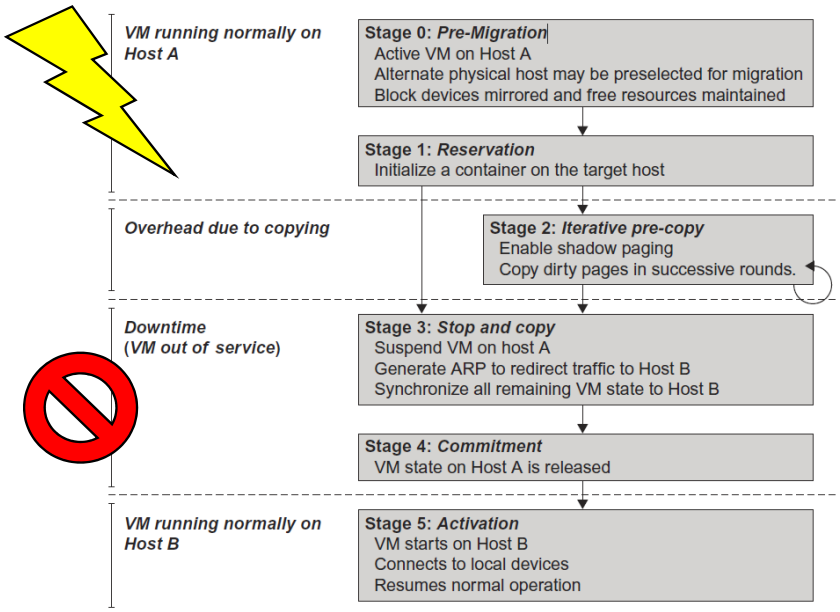
(d) Life migration

modified from [3] Distributed & Cloud Computing Book

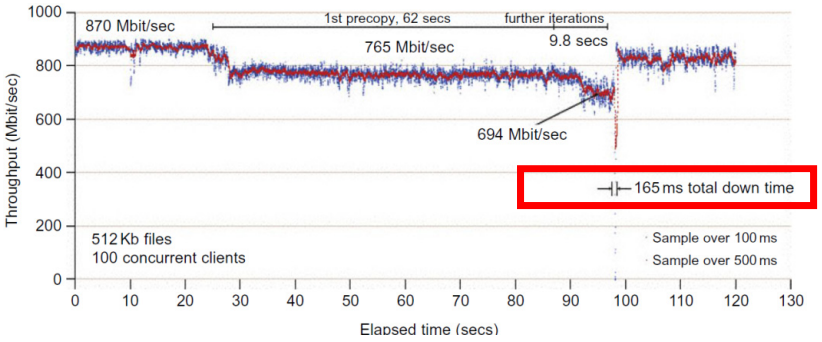
- **Common VMM operations such as multiplexing, suspension, provisioning, and life migration enable a VM to be provisioned to any hardware platform and makes application porting easier**

Live Migration Operation – Example

- Live migration process of a VM from one host to another
  - Reason: 'Host A' fails more and more and needs to get maintenance
  - Migration copies VM state file from the storage area to the host machine



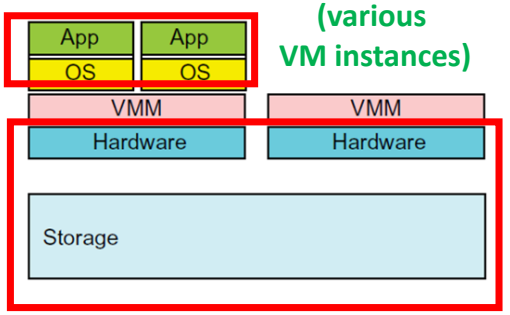
- **Goal of live migration: the effect of data transmission rate of a VM migrated from one failing host to another must be kept very minimal with small overhead; example: one failing Web server to another Web server to not (significantly) interrupt Web site operations**



modified from [3] Distributed & Cloud Computing Book

Virtual Appliance

- Hardware / software seperation
  - Physical resources (e.g. compute, storage, networking) used by different applications
  - Physical resources are mapped to applications running in various VMs
  - Dynamic mapping of system resources to specific applications with different needs
  - E.g. one VM needs more CPU power
  - E.g. second VM needs more memory & storage
- Key benefits
  - Decreased costs and increased efficiency and responsiveness
  - Virtualization enables server consolidation

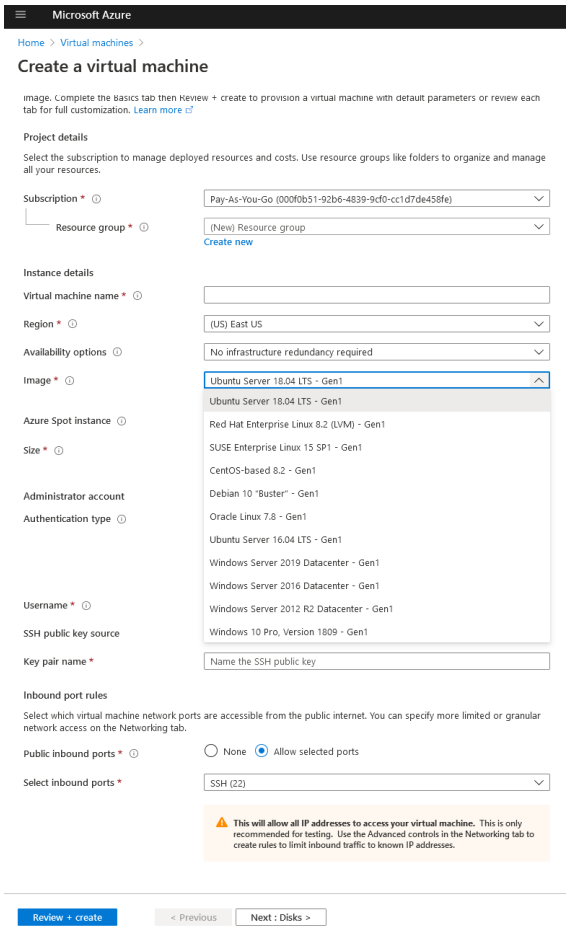


(various VM instances)

(physical resources available)

modified from [3] Distributed & Cloud Computing Book

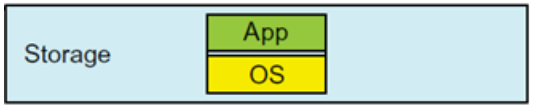
[5] MS Azure Virtual Machines



- **Application that run on its required operating system are bundled together as a 'virtual appliance'**
- **Virtual appliance can be ported to any hardware platform and often includes also software libraries**

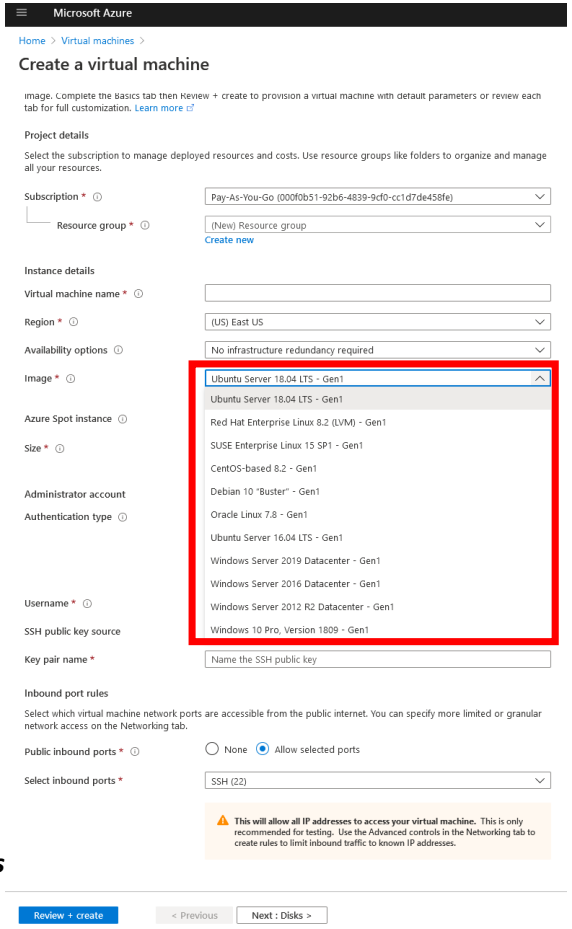
Virtual Machine Instance & Images

- Images
  - An image of a virtual machine is basically ‘a copy of the VM into file’
  - E.g. contain an OS, data files, and applications – it captures the VM state
- Images in catalogs
  - Images make sure that they are created with the proper patches
  - Images ensures any software is installed & configured with good settings
- Examples
  - Image can be a plain OS with pre-installed software or server alongside
  - Cloud computing providers like Amazon EC2, IBM, etc. use a lot pre-installed images



(VM images can be often downloaded or selected when using clouds)

modified from [3] Distributed & Cloud Computing Book



[5] MS Azure Virtual Machines

- **A virtual machine (VM) that is currently running and used by a user is known as a VM instance**
- **Virtual machine images are templates for creating new VM instances that run with a VMM**
- **User often choose images from existing catalogs or save own VM images of running instances**

Virtualization Software Architectures & Examples

- Key requirements
  1. Provide an environment essentially identical to original machine
  2. Programs run in this environment only with minor decreases in speed
  3. Virtualization software is in complete control of the system resources

Provider and References	Host CPU	Host OS	Guest OS	Architecture
VMware Workstation	x86, x86-64	Windows, Linux	Windows, Linux, Solaris, FreeBSD, Netware, OS/2, SCO, BeOS, Darwin	Full Virtualization
VMware ESX Server	x86, x86-64	No host OS	The same as VMware Workstation	Para-Virtualization
Xen	x86, x86-64, IA-64	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, Windows XP and 2003 Server	Hypervisor
KVM	x86, x86-64, IA-64, S390, PowerPC	Linux	Linux, Windows, FreeBSD, Solaris	Para-Virtualization

[3] Distributed & Cloud Computing Book

- **Depending on the position of the virtualization layer, there are several classes of VM architectures**
- **VM architectures are full virtualization, host-based, para-virtualization, and hypervisor architecture**

XEN Software – Hypervisor Architecture Example

- XEN is an open source hypervisor program
  - Developed by Cambridge University.
  - Provides hypercalls for the guest operating systems (OS) and applications
  - Converts physical devices into virtual resources dedicated for VMs to use
  - Does not include any device drivers natively (size is therefore rather small)
  - Provides mechanisms for a guest OS to have access to physical devices

[6] XEN Web Page

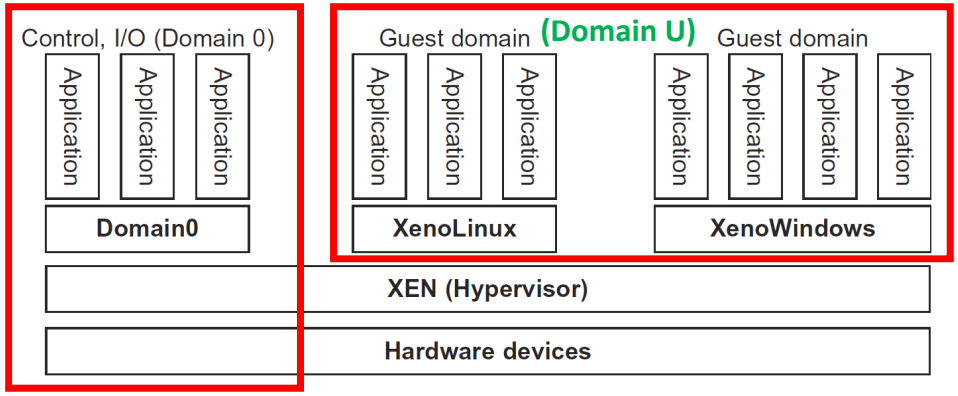
(not all guest OSs are created equal)

(special domain 0 controls the others)

(domain 0 designed to access hardware directly and manage devices)

(domain 0 allocates and maps hardware resources to guest domains)

(domain 0 is first loaded when XEN boots w/o any file system)

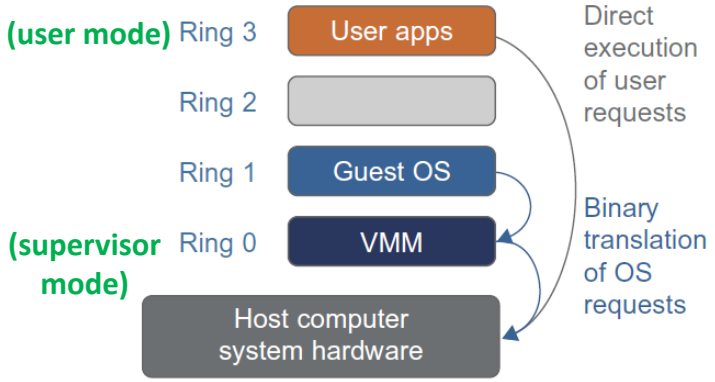


modified from [3] Distributed & Cloud Computing Book

VMWare Workstation – Full Virtualization Architecture Example

- Full Virtualization Architecture (Hypervisor/VMM)
  - Guest OSes and applications consist of non-critical & critical instructions
  - Non-critical instructions run on the hardware directly
  - Non-critical instructions do not control hardware or threaten the security
  - Critical instructions are discovered and replaced with ‘traps‘
  - ‘Traps‘ mean traps into the VMM to be emulated by software
  - The method used in this emulation is called ’binary translation’
- Example: VMWare Workstations
  - VMware Workstation Pro and VMware Workstation Player
  - E.g. run multiple operating systems as virtual machines on a single PC

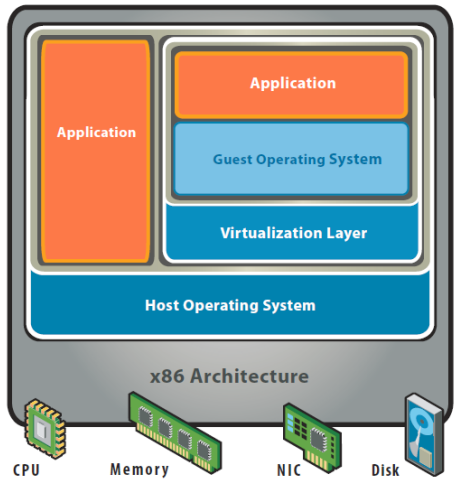
[7] VMWare Workstation



[3] Distributed & Cloud Computing Book

Host-Based Virtualization Architecture Example

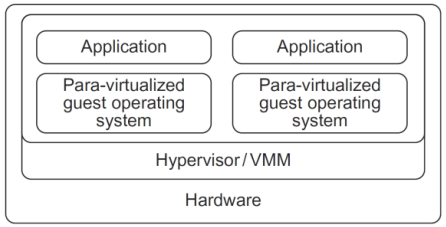
- Approach
  - Both a host OS and a guest OS are used
  - Virtualization software layer is built between the host OS and guest OS
  - Host OS is still responsible for managing the hardware
  - Idea is to install a virtualization layer on top of the host OS
  - The guest OSes are installed and run on top of the virtualization layer
- Motivated by Benefits
  - Dedicated applications
    may run on the VMs
  - Some other applications can also run with the host OS directly
- Lessons learned
  - High flexibility, but performance is too low to be useful in practice!



[8] VMWare Virtualization White Paper

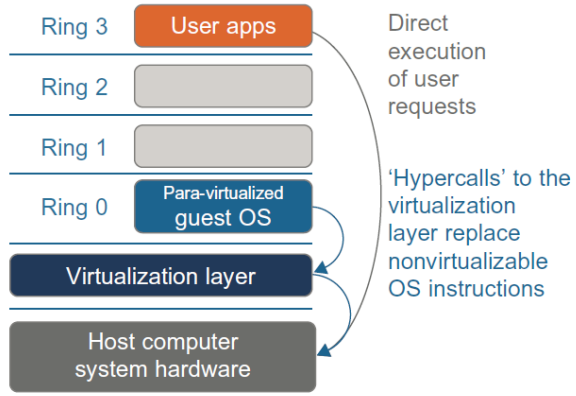
KVM Para-Virtualization Example

- Approach
  - Modify guest operating system (OS) with APIs
  - Para-virtualized virtual machine (VM) provide APIs
  - Reduce the virtualization overhead & keep VM fast
  - Improve performance by modifying only the guest OS kernel
  - Replace 'nonvirtualizable instructions' with hypercalls for the hypervisor or the VMM to carry out the virtualization process
  - Replacement done with special compiler
  - Cost of maintaining para-virtualized OS is too high (e.g. Kernel changes)
- Example
  - KVM software (Kernel-based VM) as part of Linux operating systems



[9] KVM Kernel Virtual Machine

(e.g. hardware environment not simulated)

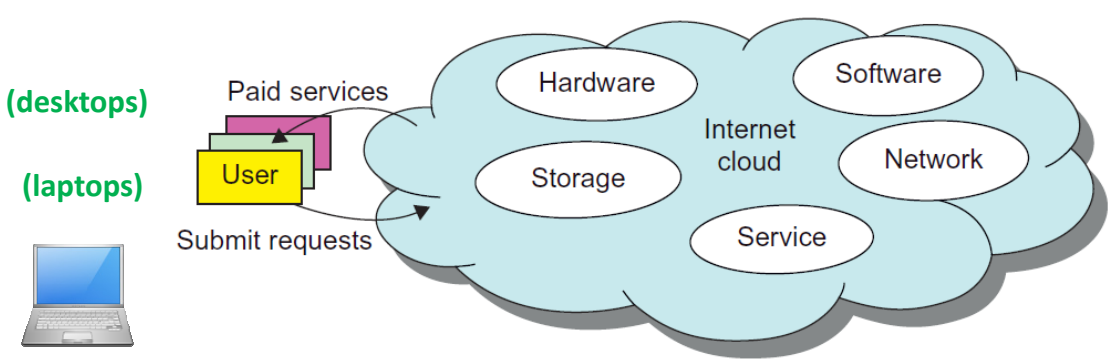


[3] Distributed & Cloud Computing Book

What is Cloud Computing from 10.000 ft (cf. Lecture 1)?

- Data Centres
  - Provide virtualized resources to form an Internet cloud
  - Provisioned with hardware, software, storage, network, and services
  - End user pay to run their applications or services after submitting requests

[3] Distributed & Cloud Computing Book

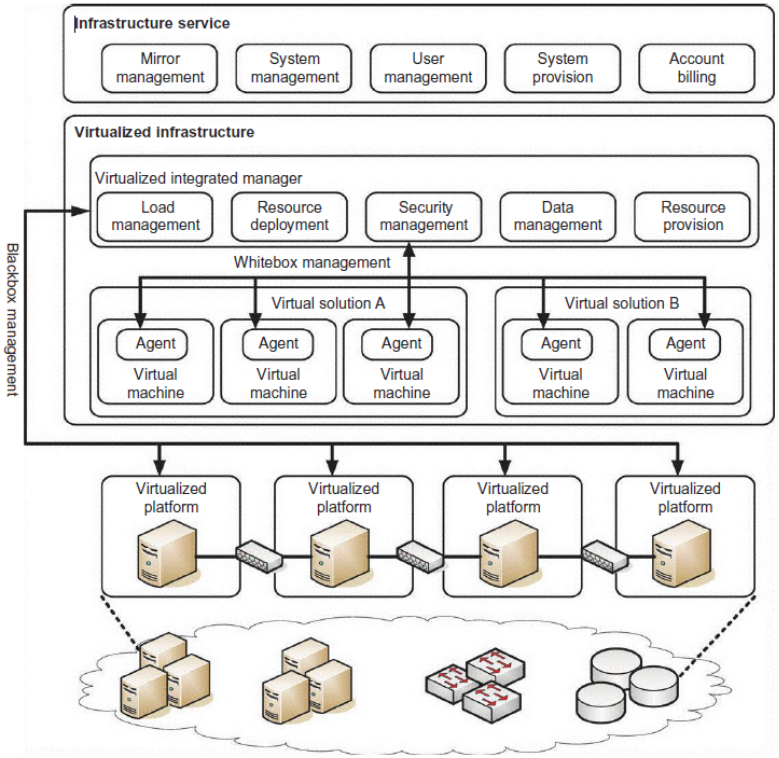


(service-oriented platform with virtualized resources)

- **Cloud computing moves desktop computing and laptop computing via the Internet to a service-oriented platform using remote large server clusters and massive storages to data centres**
- **Virtualization has enabled the cost-effectiveness and simplicity of cloud computing solutions and enabled services for multiple users**

Virtualized Resources in Data Centers

- Virtualized Infrastructure
  - Enabled by hardware virtualization concept
  - Virtualized servers, storage, and network combined in virtualized platforms
- Cloud computing
  - Enable elastic cloud systems via server virtualization
  - VMs installed on a virtualized platform are often used for hosting third-party programs (e.g. content streaming server)

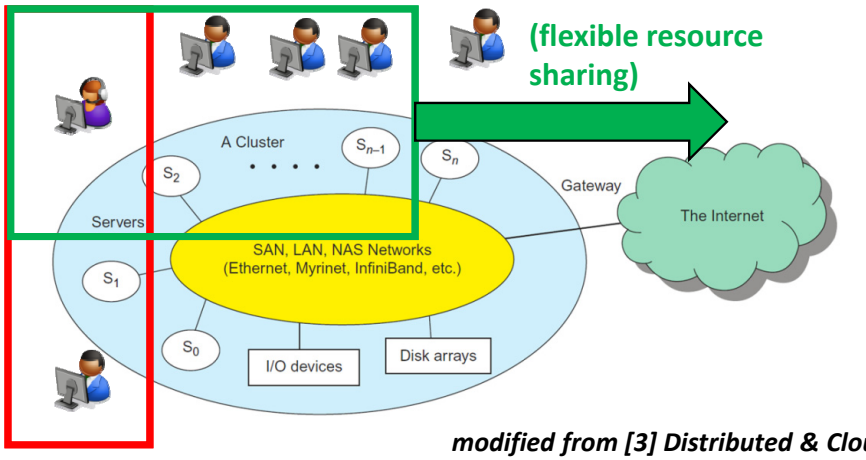


[3] Distributed & Cloud Computing Book

(basis is still physical hardware)

➢ **Lecture 13 provides more details about Cloud Operating Systems and offers insights on Cloud computing security aspects missing here**

Sharing Resources & Disaster Recovery

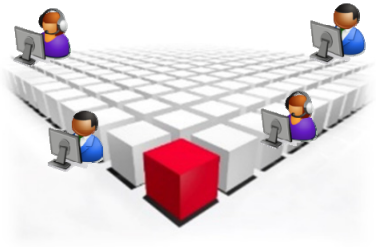


(flexible resource sharing)

- **Using VMs in a cloud computing platform ensures extreme flexibility for end users**
- **Data centre resources are shared by many users and virtualization maximizes the users‘ privileges and keep work seperate**
- **Traditional sharing of cluster resources with complex user & group mechanisms**

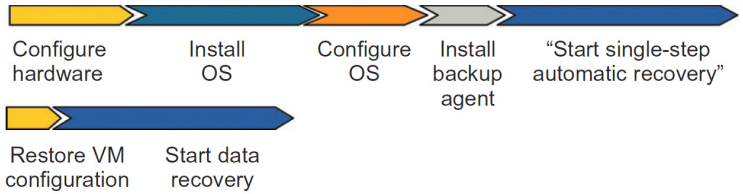
modified from [3] Distributed & Cloud Computing Book

[1] Apache Spark



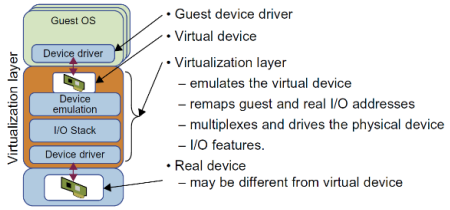
(resource sharing example for an Apache Spark cluster deployment, need for YARN scheduler, cf. Lecture 3)

- Virtualization also impacts fault tolerance
  - Recovery overhead of a conventional disaster recovery scheme high
  - Recovery overhead required to recover from live migration of VMs low

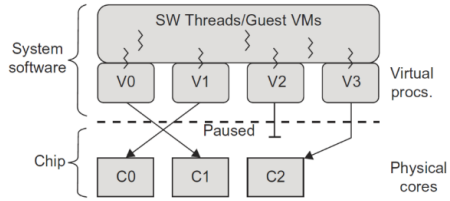


Physical vs. Virtual Processors towards Scalable Virtual Clusters

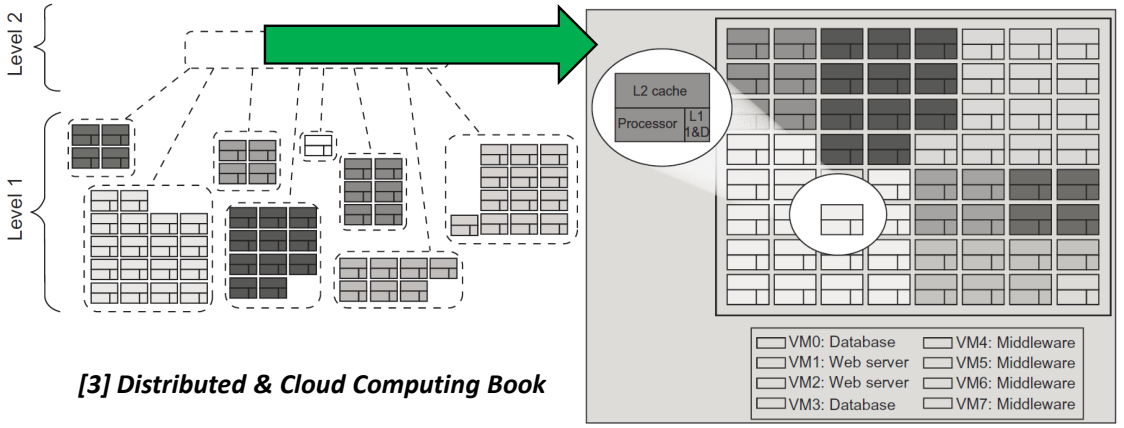
- Multi-core (cf. Lecture 1) virtualization method
  - Exposes four virtual CPUs (VCPUs) to the software application
  - Only three physical cores are actually present



- I/O devices (cf. Lecture 1) virtualization method
  - Device emulation for I/O virtualization inside the virtualization layer
  - Maps real I/O devices into the virtual devices for the guest device driver



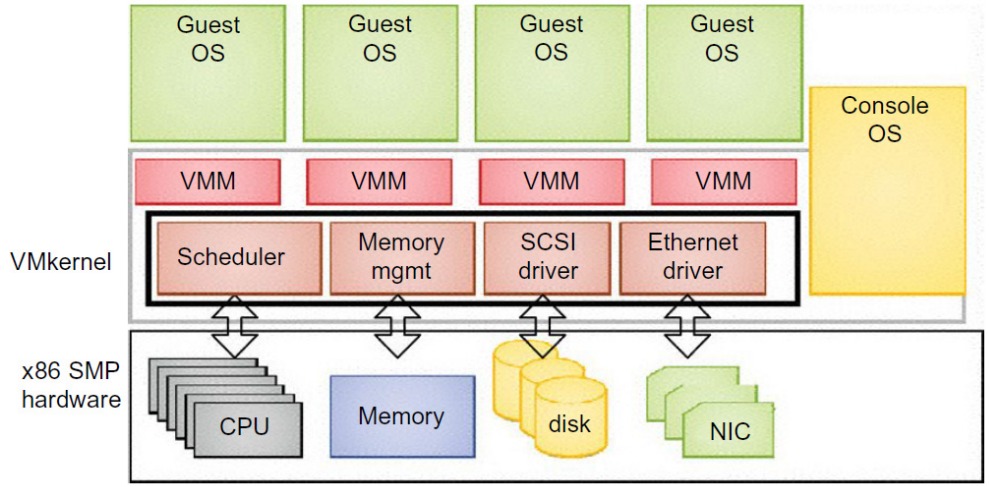
- Many-core (cf. Lecture 1) virtualization method
  - Important is the mapping of VMs into adjacent cores (performance)
  - E.g. multiple virtual clusters assigned to various workloads
  - Enables server consolidation by space-sharing of VMs
  - Future trend as many-core grows



[3] Distributed & Cloud Computing Book

Software Example: VMware ESX Server

- Server virtualization software
  - Different than VMWare Workstation (desktop virtualization)
  - Symmetric multiprocessing (SMP) servers are massively parallel systems
  - VMWare ESX is a VMM / hypervisor for bare-metal x86 SMP systems
  - Scheduler allocates CPU, memory disk, and network bandwidth resources
  - Scheduler maps resources to the virtual hardware resource set of each VM

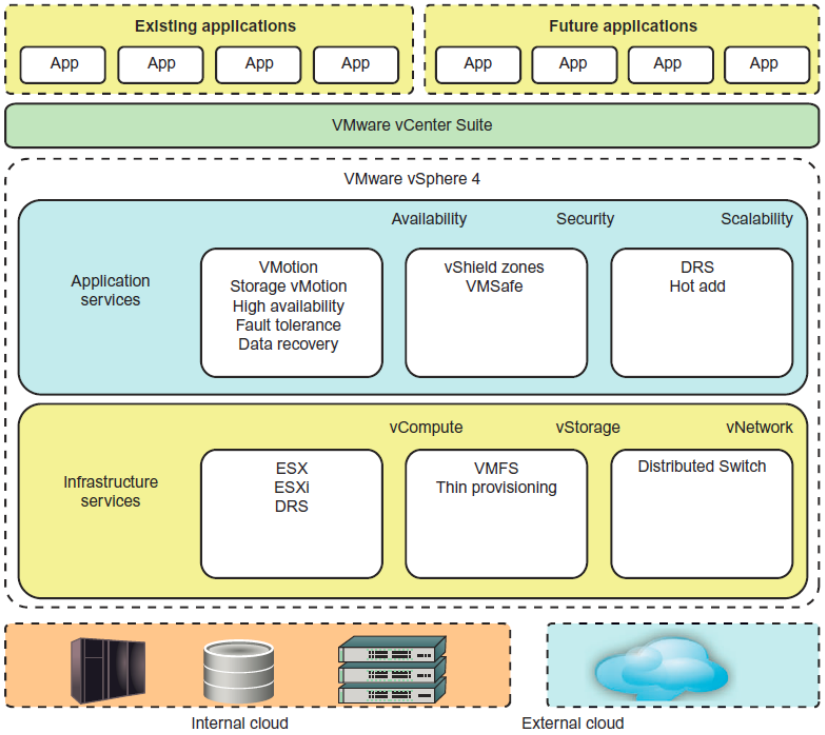


[3] Distributed & Cloud Computing Book

Software Example: VMWare vSphere & vCenter Suite

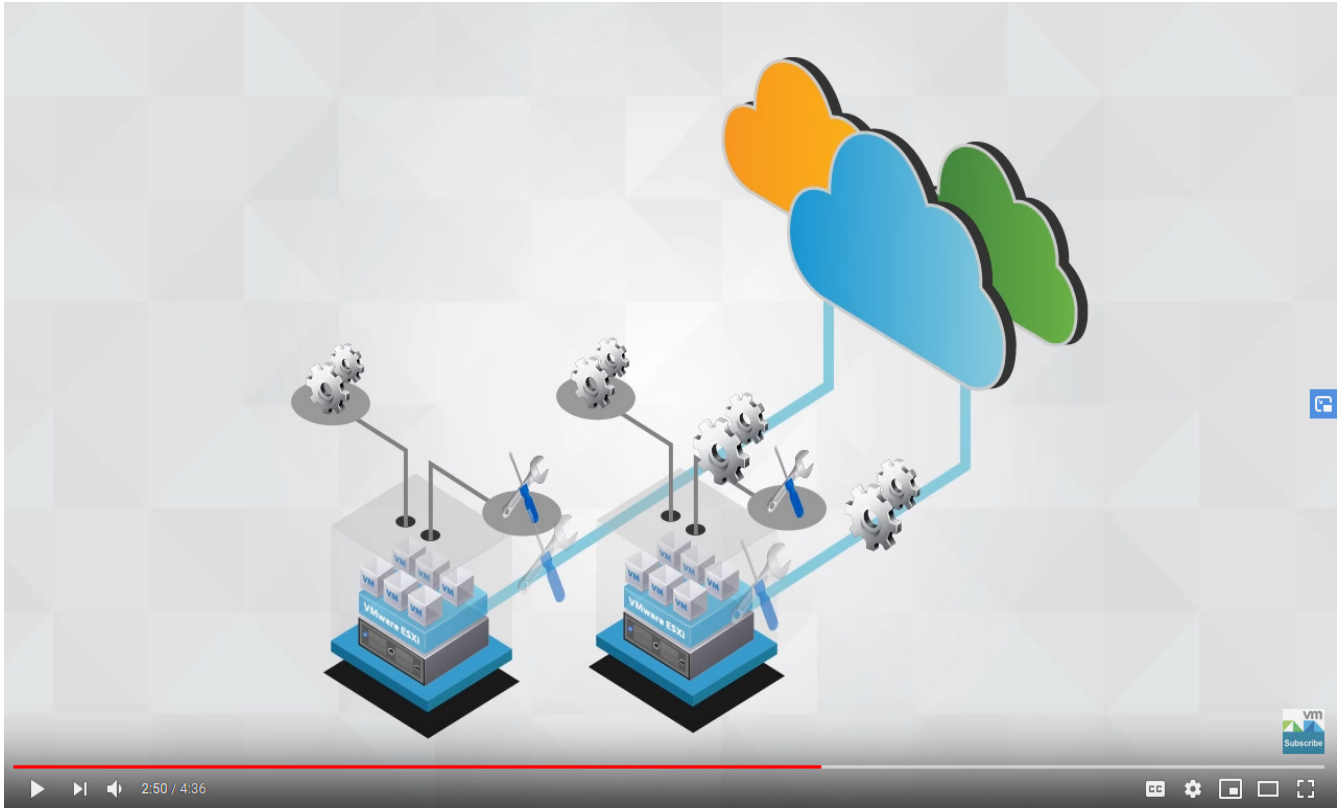
- Cloud operating system
  - Manages compute, storage, and network resources over virtualized data centers (e.g. internalcloud resources)
  - Complements internal cloud (e.g. private cloud) with external cloud (e.g. public cloud from Amazon)

[3] Distributed & Cloud Computing Book



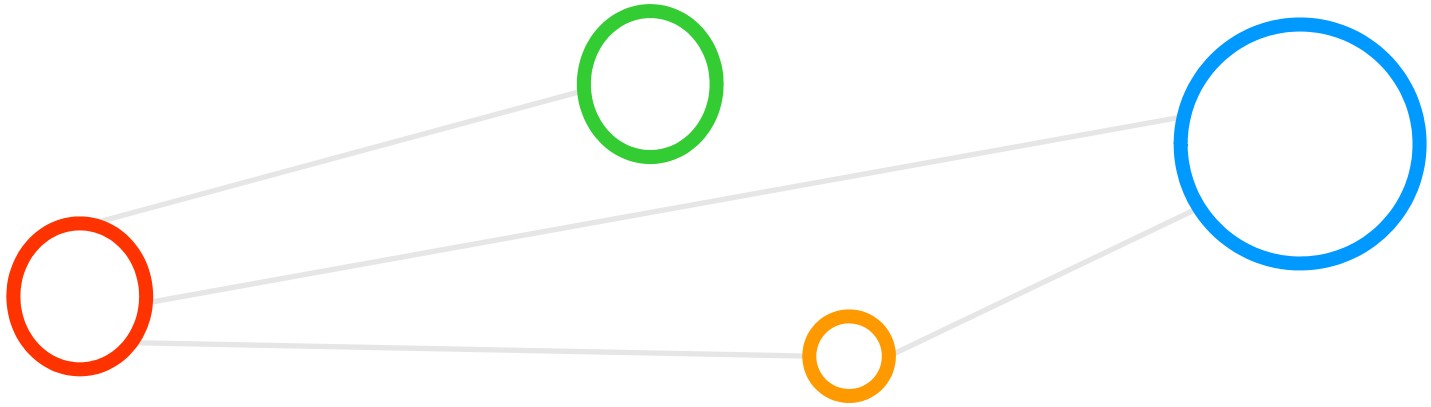
➢ **Lecture 13 provides more details about Cloud Operating Systems and offers a detailed example using the OpenStack technology stack**

[Video] Data Center Virtualization and Standardization



[10] YouTube video, Data Center Virtualization and Standardization

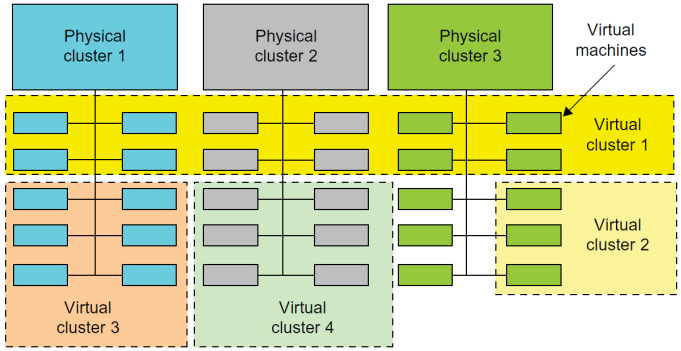
Cloud Data Center Design



Physical vs. Virtual Clusters enabling Scalable Cloud Computing Centers

- Several virtual clusters example
  - Each virtual cluster 1-4 is formed with physical machines
  - Virtual cluster 1-4 formed by a VM hosted by multiple physical clusters
  - Virtual cluster boundaries are shown as distinct boundaries as experienced by real end users
  - Physical clusters with a topology not really known by real end users but only by cloud data centre administrators
  - E.g., enables many Apache Spark worker and head nodes that are distributed in the data center itself using virtualization

[1] Apache Spark

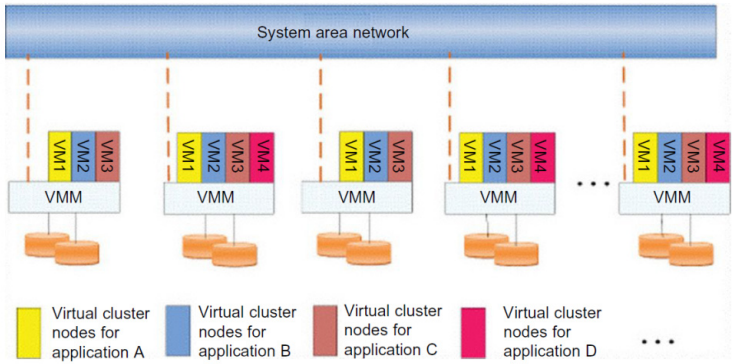


- **VM method for simplifying management & sharing of physical resources in Cloud data centres**
- **Virtual clusters are built with VMs installed at distributed servers from several physical clusters**
- **VMs in virtual clusters are interconnected logically by a virtual network across n physical networks**
- **Clustering inexpensive computers is an effective way to obtain reliable & scalable computing power**

[3] Distributed & Cloud Computing Book

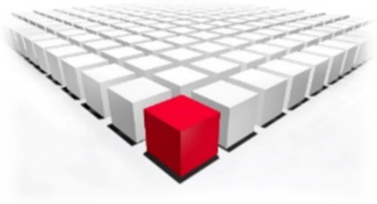
Application Partitioning on Virtual Clusters

- Selected benefits
  - Multiple VMs running with different OSes can be deployed on same physical node
  - A VM runs with a guest OS that can be different from the host OS
  - VMs can be replicated in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery
  - Failure of any physical nodes may disable only some VMs installed on them
  - But the failure of VMs will not pull down the complete host system



[3] Distributed & Cloud Computing Book

[1] Apache Spark



- **Size (number of CPU nodes) of a virtual cluster can grow and shrink dynamically on demand**
- **The purpose of using VMs is to consolidate multiple functionalities on the same server that will greatly enhance server utilization and application flexibility for a wide variety of cloud applications**

Public Cloud Computing Examples using Virtualization in Large Data Centers

Provider	AWS	Microsoft Azure	GAE
Compute cloud with virtual cluster of servers	x86 instruction set, Xen VMs, resource elasticity allows scalability through virtual cluster, or a third party such as RightScale must provide the cluster	Common language runtime VMs provisioned by declarative descriptions	Predefined application framework handlers written in Python, automatic scaling up and down, server failover inconsistent with the web applications
Storage cloud with virtual storage	Models for block store (EBS) and augmented key/blob store (SimpleDB), automatic scaling varies from EBS to fully automatic (SimpleDB, S3)	SQL Data Services (restricted view of SQL Server), Azure storage service	MegaStore/BigTable
Network cloud services	Declarative IP-level topology; placement details hidden, security groups restricting communication, availability zones isolate network failure, elastic IP applied	Automatic with user's declarative descriptions or roles of app. components	Fixed topology to accommodate three-tier web app. structure, scaling up and down is automatic and programmer-invisible

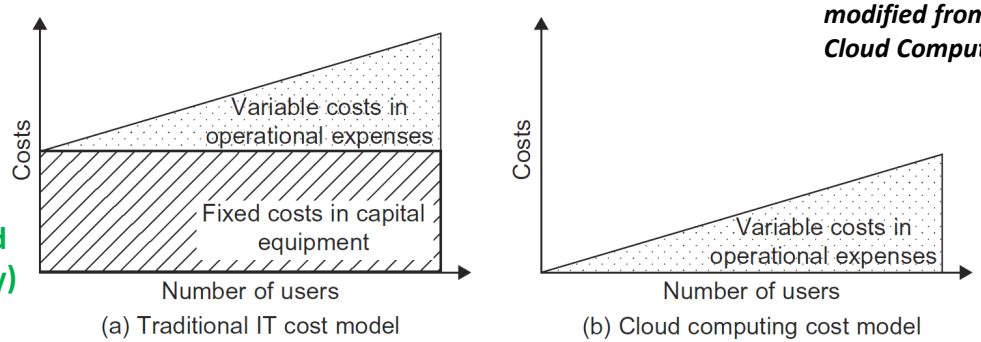
[3] Distributed & Cloud Computing Book

➢ **Lecture 8 & 9 & 10 offer more insights into concrete cloud systems and their use of virtualization on different levels of cloud services**

Understanding Cloud Computing Cost Models

- Traditional IT computing
  - Users or company owners must acquire their own computer and peripheral equipment as capital expenses
  - Users or company owners face operational expenditures in operating and maintaining the system (e.g. personnel, service costs, energy, etc.)
  - Fixed cost is main cost (reduced slightly with increasing number of users)
  - Operational costs increase sharply ('escalates') with larger number of users

(every ~3 - 5 years this fixed costs needs to be renewed with new technology)



modified from [3] Distributed & Cloud Computing Book

[1] Apache Spark

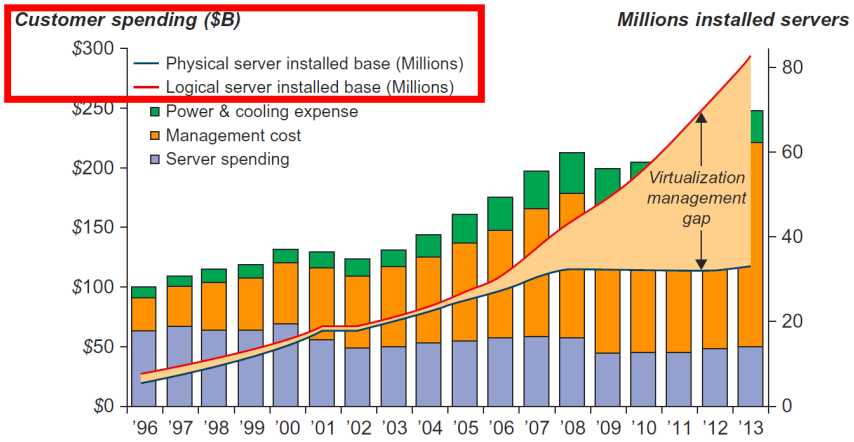
(e.g., variable cost depending on how much an Apache Spark cluster is used by different users in an enterprise or startup)

- **Cloud computing applies a pay-per-use model, in which user jobs are outsourced to data centers**
- **Cloud users have no up-front cost in hardware acquisitions, only variable costs – good for startups**

Data Center Metrics – Infrastructure Costs Overview

- Data center infrastructure capacity
  - Large data centers are typically built with thousands of servers and more
  - Small data centers are typically built with hundreds of servers
  - Costs to build and maintain data center servers has increased over years
- Understanding cost breakdown
  - Only 30% IT equipment (servers, disks)
  - 33% is for the chiller (cooling)
  - 18% emergengy power (UPS)
  - 9% air conditioning (CRAC)
  - 7% power distribution, lights, and transformer costs

(factors explain why cooling with ambient cold air like in Iceland is so cost effective)



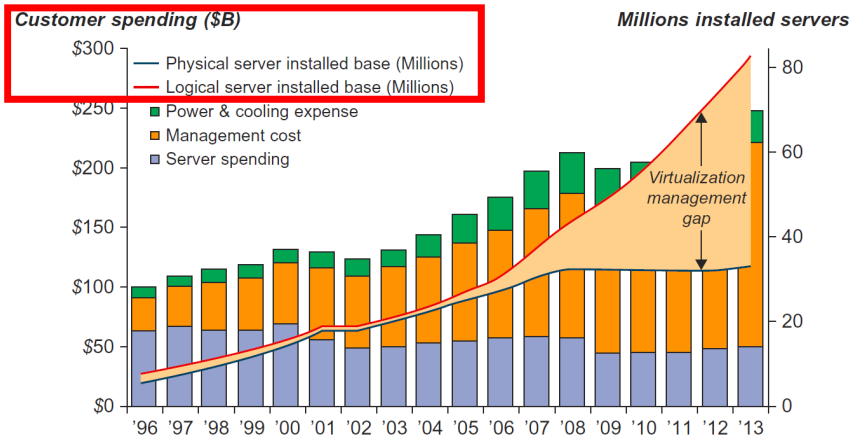
modified from [3] Distributed & Cloud Computing Book

Data Center Metrics – Infrastructure Costs

- Energy/Power costs
  - Power increases linearly w.r.t. clock frequency (i.e. 'heartbeat' CPUs)
  - Clock frequency rate can not be increased more (< 5 GHz)
  - Power increases quadratically w.r.t. voltage applied on chips
  - This includes significant costs for water/cooling/chiller
  - Key problem: cost of electricity and cooling increased from 5% to 14% (in ~15 years)

(problem explains why Iceland is attractive since energy prices still relatively low here and ambient air cooling helps to reduce costs too)

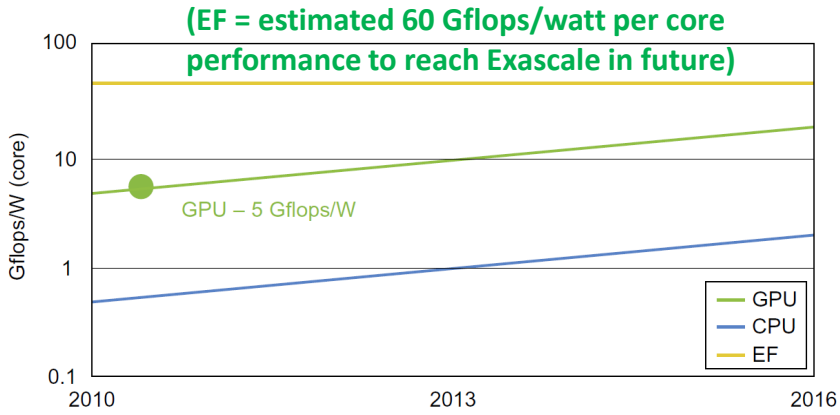
- **Power consumption, cooling, and packaging limits large data center system development**
- **Data centers use price/performance ratio: energy efficiency is more important than speed**
- **~ 60% costs to run a data center is allocated to management & maintenance (infrastructure)**



modified from [3] Distributed & Cloud Computing Book

Data Center Example: Power Efficiency of the GPU

- Key benefits
  - Power and massive parallelism of GPUs over CPUs (also for the future)
  - CPU optimized for latency in caches and memory
  - GPU optimized for throughput with explicit management of on-chip memory (cf. Lecture 1)
  - Data movement dominates power consumption → key!
  - Metric: Performance/power ratio measured in Gflops/watt per core
  - Measure: Floating point operations per second (FLOPs)



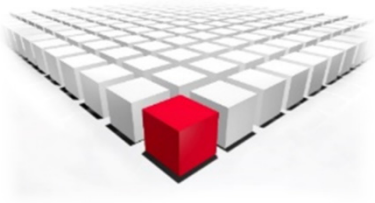
[3] Distributed & Cloud Computing Book

- **GPUs are inherently more energy efficient than other ways of computation because they are optimized for throughput and performance per watt and not absolute performance like CPUs**

Cloud Computing & Big Data Centers – Economics of Scale

- E.g. Microsoft & Cloud Computing
  - Has ~100 data centers (large/small) distributed around the globe

[3] Distributed & Cloud Computing Book



(11 times the size of a football field)

(housing 400.000 – 1 million servers)

(network cost to operate a small data center with 1000 servers is about 7 times greater / unit)

(storage cost to operate a small data center with 1000 servers is about 5,7 times greater)

- **Clouds are built on massive large-scale data centers that in turn are often built with a large number of servers connected through a huge interconnection network contributing to a 'large energy footprint'**
- **Big Data centers use 'economics of scale': the larger the data center, the lower the operational cost**

Data Center Construction & Terminology

- Off-the-shelf server hardware in a data center
  - A number of processor sockets, each with a multi-core CPU
  - CPU has an internal cache hierarchy
  - Local shared and coherent DRAM (bottleneck)
  - A number of directly attached disk storage drives
  - The DRAM and disk resources within the rack are accessible through first-level rack network switches
- Server Clusters
  - Many servers connected together
  - All resources in all racks are accessible via a cluster-level network switches

(server)

(racks)

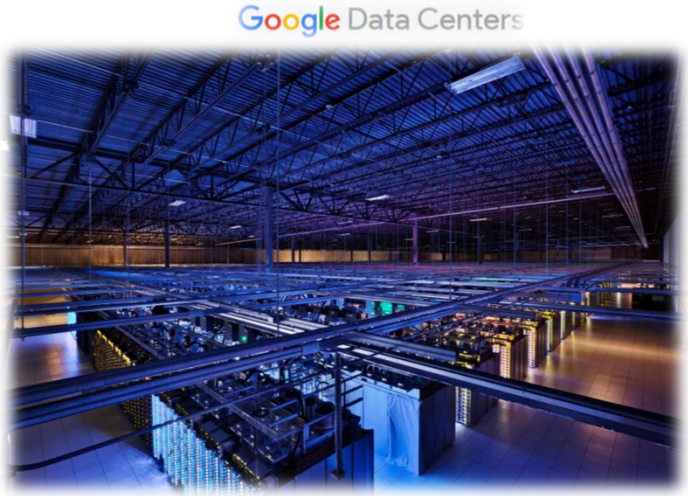
(multiple servers in racks)

- **Many data centers are often built with commercially available commodity components**
- **Components used in data centers are different from those in building supercomputer systems**
- **Instead of using many different small servers standalone and interconnected, cloud computing servers are typically organized via multiple servers in so-called racks**

Assume Failure in Data Center Design

- Example: Google Data Center, Council Bluffs, Iowa, USA
  - Scale of thousands of servers means always concurrent failure
  - Hardware failure or software failure (1 percent of nodes is common)
  - CPU failure, disk I/O failure
  - Network failure, switch error
  - Sofware failure, scheduler error
  - E.g., error in execution as part of as Apache Spark cluster job means being (partly) resubmitted for execution in an automatic way (cf. Lecture 3)
- Whole data center does not work
  - E.g. in the case of a power crash?!

[1] Apache Spark



[12] Google Data Centers, wired.com

- **Cloud data centres need to operate with failures: Computing service and user data should not be lost in a failure situation and reliability can be achieved by redundant hardware in centres**
- **Software keeps multiple copies of data in different locations and keeps data accesible during errors**

➢ **Lecture 5 provides insights into Hadoop Distributed File System (HDFS) that assumes failure by default and fits to data center practice**

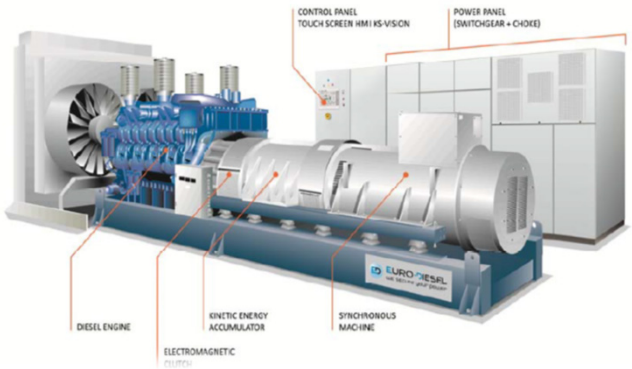
Uninterruptible Power Supplies (UPS)

- E.g. Diesel power generators kept as ‘plan B‘ for power crash
  - Refueling possible, but tried to be used as minimal as possible
  - Important since large losses in data center power distributions from UPS
- Other design hints
  - Google offers his experiences to the greater public
  - E.g. keep high voltages as close to the power supply as possible in order to reduce power line losses

[15] Google Data Center, big-data.tips

(DRUPS = Diesel Rotary UPS)

[13] Greener data center, datacentertalk.com

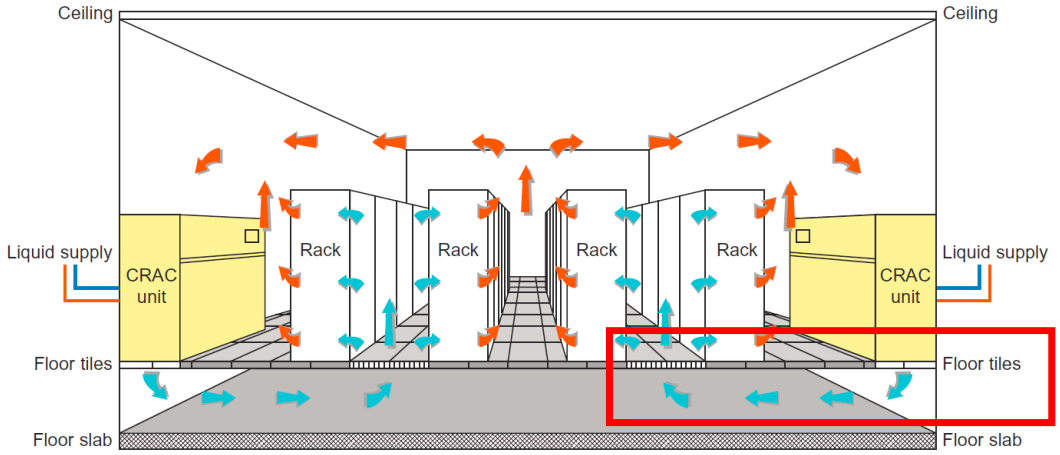


[14] DRUPS

- **Uninterruptible power supplies (UPS) enable cloud data centers to operate during power crashes so that the interrupt is not visible for end-users**
- **The machines are also known as Diesel Rotary UPS powered by Diesel fuel and can hold power up to hours/days (not a long-term solution)**

Data Center Cooling Systems

- Raised-floor data center design
  - Steel grid resting on stanchions about 2–4 ft above the concrete floor
  - Hiding cables, power lines, and cooling supplies
  - Enables hot-cold air circulation supporting water heat exchange facilities
  - Under-floor area: distribute cool air to server racks (& route power cables)
  - CRAC (computer room air conditioning) unit pressurizes the raised floor plenum by blowing cold air into the plenum and takes in hot air again



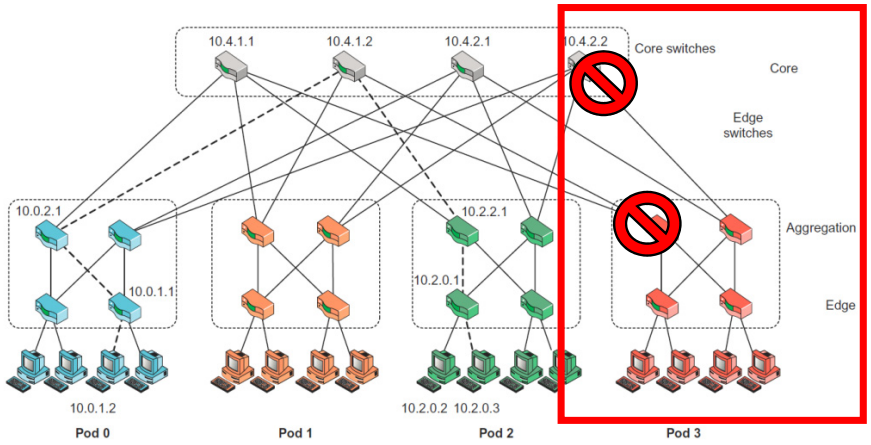
Modified from [3] Distributed & Cloud Computing Book

(perforated tiles in front of server racks)

Scalable Interconnection Network for Data Centers

- Example: Fat-tree switch network design (2 layers)
  - Applied to interconnect the server nodes that are within the racks
  - Edge switches connect nodes in bottom layer
  - Core switches provide different paths among different pods
  - Pod is a group of aggregation switches, edge switches, and ‘leaf nodes’

(multiple connected server nodes/racks)



Modified from [3] Distributed & Cloud Computing Book

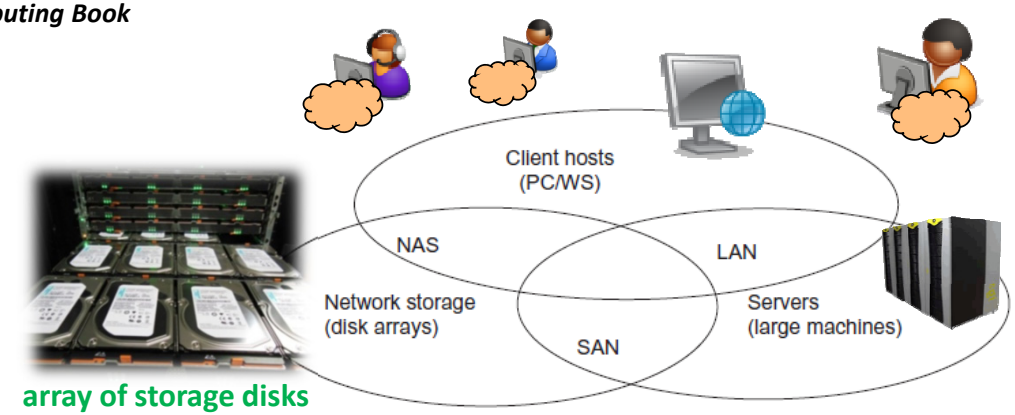
(alternate paths in case of isolated link failures exist to keep the ‘pod‘ accessible to end users)

- **The fat-tree switch network design with two layers enables a scalable interconnection network with different types of switches for data centers that consists of multiple server nodes and racks**
- **A fat-tree structure provides multiple paths between any two servers to be fault-tolerant**

Data Center System-Area Network Interconnects

- Three types of networks
  - Used in large clusters built with commercial network components

[3] Distributed & Cloud Computing Book

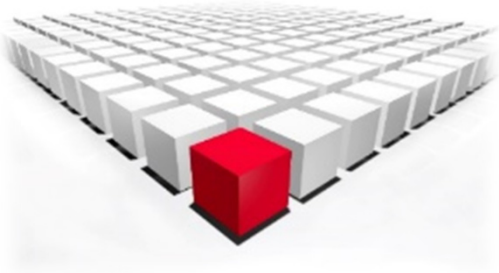


array of storage disks

- **Small cluster nodes are mostly interconnected by an Ethernet switch or local area network (LAN)**
- **A LAN is also often used to connect client hosts to big servers and large machines**
- **A storage area network (SAN) connects servers to network storage such as disk arrays**
- **Network attached storage (NAS) connects client hosts directly to the disk arrays**

Data Center Metrics – Size & Software Scalability

- Different dimensions of scalability are relevant for data centers
  - Users want to have a systems with scalable performance (cf. Lecture 1)



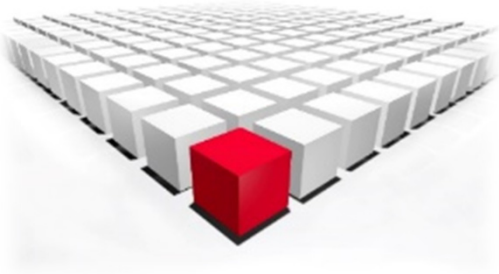
- Size scalability
  - Size: add/remove processors, cache, memory, storage, or I/O channels
  - E.g. counting the number of processors installed in one machine
- Software scalability
  - Software: OS, compilers, mathematical/engineering libraries, programming environments, existing tools and older serial software
  - E.g. New Matlab version on new GPU machines, otherwise not using them

- **Data center resources are processors, cache, memory, storage, I/O network channels, etc.**
- **Data center system scaling means increase/decrease resources depending on several factors**
- **Size scalability: achieve higher performance or more functionality by increasing machine size**
- **Software scalability: software upgrades for better performance and use of larger systems**

➢ **Lecture 13 provides more examples & applied scalability for different services using the OpenStack cloud operating system**

Data Center Metrics – Application & Technology Scalability

- Application scalability
  - Application has a problem size that affects the size of the data set or the workload increase for a data center server (problem size vs. machine size)
  - E.g. server for counting words gets one document & is 98% idle: add docs
  - E.g. storage is only filled 4%, application can further scale until storage full
  - E.g. application uses just one core from a multi-core chip, but can use more
- Technology scalability
  - Technologies: processor components, networking technologies, storages
  - E.g. new-generation CPU means impact to motherboard and power supply
  - E.g. packaging & energy concerns when porting among different suppliers
  - E.g. use of heterogenous hardware/software components from n vendors



- **Application scalability: Instead of increasing machine size, application enlarges problem size**
- **Technology scalability: Systems that can adapt to changes in building technologies (e.g. new CPUs and/or GPUs)**

➢ **Lecture 13 provides more examples & applied scalability for different services using the OpenStack cloud operating system**

Public/Private/Hybrid Cloud Deployment Models

- Public clouds are built over the Internet and can be accessed by any user who has paid for it
- Public clouds are owned by service providers and are accessible through a subscription
- Public clouds deliver a set of business processes or infrastructure resources via price-per-use

[16] Private Cloud, big-data.tips

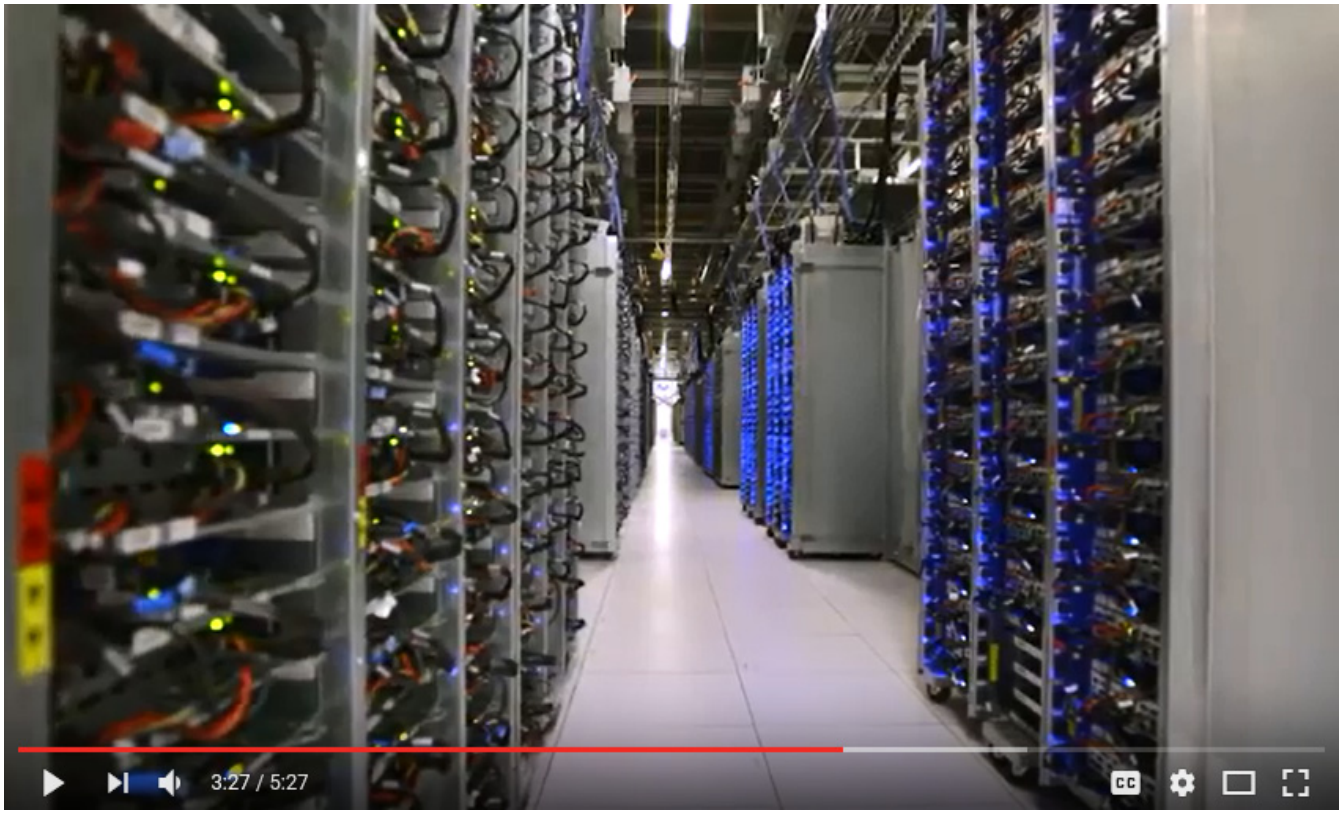
- Hybrid Clouds are a smart combination of public and private clouds supplementing local IT
- Hybrid Clouds typically employing public cloud services for many non-sensitive operations
- Supplements local infrastructure with computing capacity from an external public cloud

- Private Clouds are built within the domain of an intranet owned by a single organization
- Private Clouds are 'client owned' and managed and access is limited to owning clients
- Private Clouds do not sell resources over the Internet but provide them in the organization
- Operations only in one 'private organization' - reasons: security w.r.t. the data stored in the cloud; further potentially reasons: Cost of rent is bigger than own long-term hosting

[3] Distributed & Cloud Computing Book

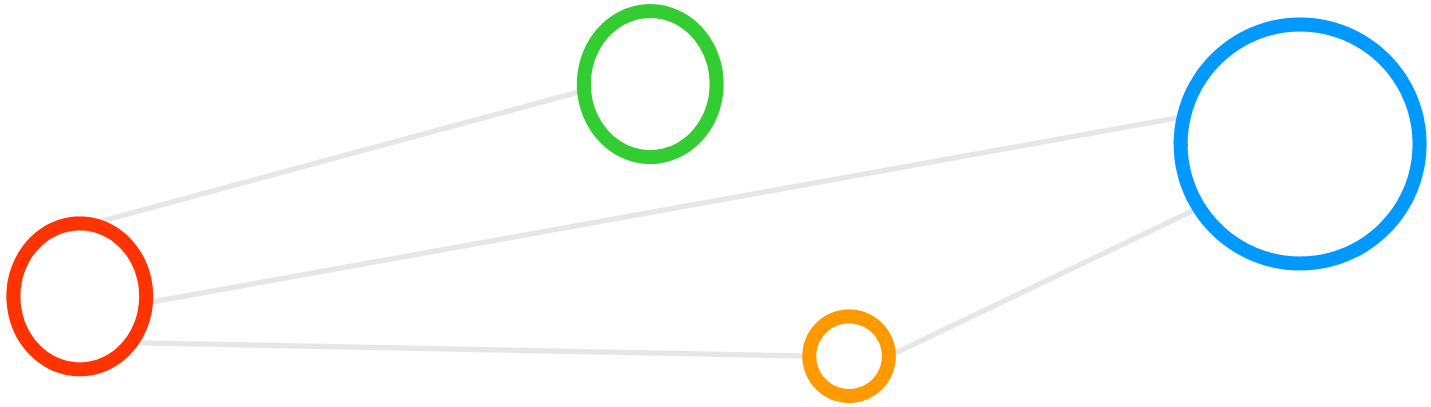
➢ Lecture 8 & 9 & 10 will clarify & compare cloud deployment models with the different cloud computing layers IAAS, PAAS, and SAAS

[Video] Google Data Center Example



[11] YouTube video, Inside a Google Data Center

Lecture Bibliography



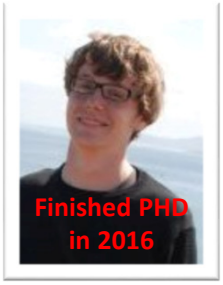
Lecture Bibliography (1)

- [1] Apache Spark Web page, Online: http://spark.apache.org/
- [2] www.big-data.tips, 'Gradient Descent', Online: http://www.big-data.tips/gradient-descent
- [3] K. Hwang, G. C. Fox, J. J. Dongarra, 'Distributed and Cloud Computing', Book, Online: http://store.elsevier.com/product.jsp?locale=en_EU&isbn=9780128002049
- [4] M. Rosenblum, The reincarnation of virtual machines, ACM QUEUE, 2004, Online: https://queue.acm.org/detail.cfm?id=1017000
- [5] Create a Windows virtual machine in the Azure portal, Online: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal?WT.mc_id=UI_empg
- [6] XEN Project Web Page, Online: https://www.xenproject.org
- [7] VMWare Workstations Web Page, Online: http://www.vmware.com/products/workstation.html
- [8] VMWare Virtualization White Paper, Online: https://www.vmware.com/pdf/virtualization.pdf
- [9] KVM Kernel Virtualization, Online: https://www.linux-kvm.org/page/Main_Page
- [10] YouTube video, Data Center Virtualization and Standardization, Online: https://www.youtube.com/watch?v=90XulQCzcoc
- [11] YouTube Video, Inside a Google Data Center, Online: https://www.youtube.com/watch?v=XZmGGAbHqa0

Lecture Bibliography (2)

- [12] ‘Google Throws Open Doors to Its Top-Secret Data Center’, wired.com, Online: https://www.wired.com/2012/10/ff-inside-google-data-center/all/
- [13] ‘Taking Greener Approach to a Data Center‘, datacentertalk.com, Online: http://www.datacentertalk.com/2011/11/taking-greener-approach-for-data-center/
- [14] ‘Diesel Rotary UPS‘, Online: http://www.greenpowerintl.com/drups.php
- [15] ‘Google Data Center‘, big-data.tips, Online: http://www.big-data.tips/google-data-center
- [16] ‘Private Cloud, big-data.tips, Online: http://www.big-data.tips/private-cloud

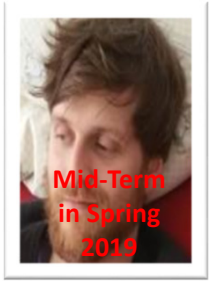
Acknowledgements – High Productivity Data Processing Research Group



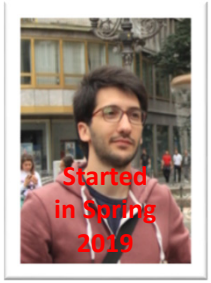
PD Dr.
G. Cavallaro

Senior PhD
Student A.S. Memon

Senior PhD
Student M.S. Memon



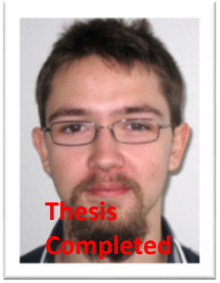
PhD Student
E. Erlingsson



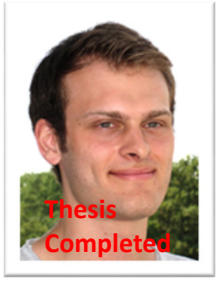
PhD Student
S. Bakarat

PhD Student
R. Sedona

Dr. M. Goetz
(now KIT)



MSc M.
Richerzhagen
(now other division)



MSc
P. Glock
(now INM-1)

MSc
C. Bodenstein
(now
Soccerwatch.tv)

MSc Student
G.S. Guðmundsson
(Landsverkjun)

This research group has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 763558 (DEEP-EST EU Project)


Data
Science
Big
Research
science
big
methods
simulations
processing
analysis
computational
Computational
technologies
TB
computing
European
Scientific
Supercomputing
climate
Centre
Computing
nodes
centers
storage
ENES
EUDAT
increasing
databases
research
HPC
Juelich
Climate
modelling
structures
datasets
performance
images
often
services
measurement
within
fusion
new
funding
concepts
Policy-based
device
Analysis
Computer
expertise
Approaches
construction
Fusion
Enable
different
forms
cross-disciplinary
resources
year
like
E.g
Test
PB
community
disciplines
results
Modelling
Cross-Disciplinary
EU
High
Environment
Key
scientific
important
files
model
international
SMAQ
engineering
References
formats
brain
etc
DKRZ
Use
basis
Infrastructure
physical
IPCC
Common
Methods
Project
MapReduce
create
via
Using
GB
Work
codes
DLCL
Level
stored
Resources
analyze
Large
Summary
data-intensive
Service
levels
external
JSC
use
e.g
Matters
Concrete
acquisition
large-scale
compute
system
Landscape
System
AR5
Hardware
distributed
cores
access
hundreds
Nutshell
Services
management
DLCLs
Understanding
Technologies
Earth
Structure
network
Germany
Simulation
directory
project
General
current
Provide
NASA
Energy
systems
day
Processing
Health