CLOUD-NATIVE PROGRAMMIERUNG

Unit 01:

Grundlagen des Cloud Computing

Stand: 13.10.2020

TECHNISCHE HOCHSCHULE LÜBECK

1

INHALTSVERZEICHNIS

Überblick über Units und Themen dieses Moduls

TECHNISCHE HOCHSCHULE LÜBECK

2

INHALTE

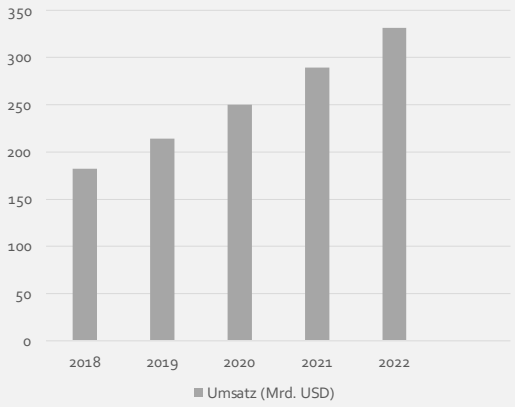
- Cloud Computing
- Cloud-Service Modelle
- Cloud-Ökonomie (Pay-as-you-Go)
- Eine kurze Architekturgeschichte der Cloud
- Cloud-native Anwendungen und Dienste
- Zusammenfassung

TECHNISCHE HOCHSCHULE LÜBECK

WACHSTUMSMARKT CLOUD-COMPUTING

*70% des Marktes teilen sich die sogenannten **Big-Five***

*„Laut Gartner wird der weltweite Markt für Public Cloud Services 2019 um 17,5 Prozent auf insgesamt 214,3 Milliarden Dollar wachsen […]. Das am **schnellsten wachsende Marktsegment** wird Infrastructure as a Service **(IaaS)** sein[…]. Die **zweithöchste Wachstumsrate** von 21,8 Prozent wird durch […] Platform as a Service **(PaaS)** erreicht."*



TECHNISCHE HOCHSCHULE LÜBECK

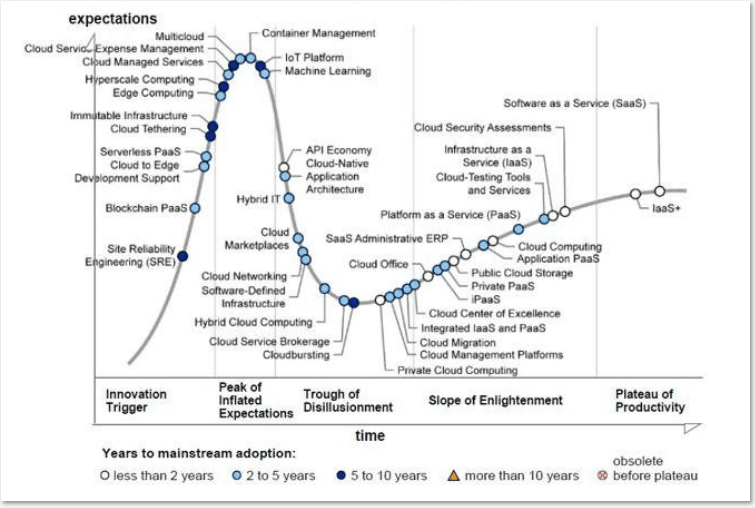
Die Entwicklung ist zwar beeindruckend, aber immer noch linear und nicht exponentiell wie manchmal behauptet wird.

BIG-FIVE:

- Amazon
- Microsoft
- Alibaba
- Google
- IBM
- Google

CLOUD COMPUTING HYPE CYCLE

Gartner, 2018



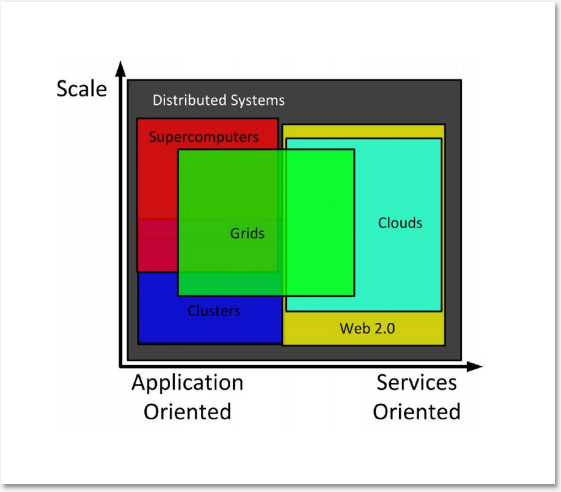
„In Zeiten des digitalen Wandels ist Cloud Computing heute die primäre Option und nicht mehr nur eine von vielen Möglichkeiten"

Gartner-Analyst Gregor Petri

TECHNISCHE HOCHSCHULE LÜBECK

CLOUD COMPUTING

Im Vergleich zu anderen Ansätzen Verteilter Systeme



Die 5 Gebote der Cloud

1. Everything fails all the time
2. Focus on MTTR and not on MTTF
3. Respect the Eight Fallacies of Distributed Computing
4. Scale out, not up
5. Treat ressources as cattle, not pets

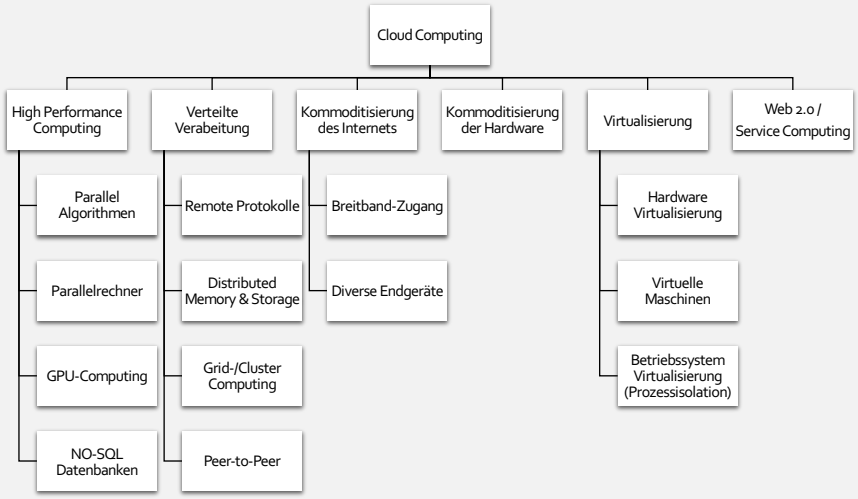
MTTR = *Mean Time to Repair*

MTTF = *Mean Time to Failure*

TECHNISCHE HOCHSCHULE LÜBECK

CLOUD COMPUTING IST KEINE ÜBERRASCHUNG

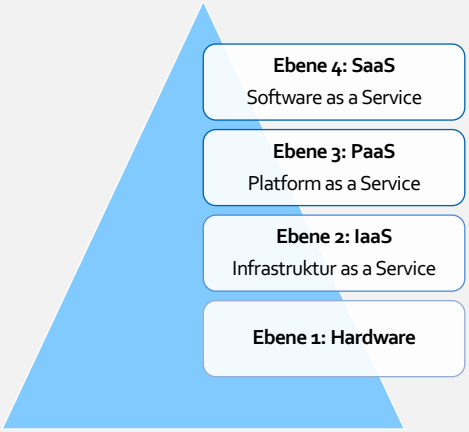
Sondern auf den Schultern von Giganten entstanden



TECHNISCHE HOCHSCHULE LÜBECK

DAS SCHICHTENMODELL DES CLOUD COMPUTINGS

Service - Modelle



Kunden, Endnutzer
- Anpassbare Software-Dienste
- XaaS (Everything as a Service)
- Transparente Updates

Entwickler
- Programmierschnittstellen (APIs)
- Plattformdienste
- Abstraktion der technischen Infrastruktur

Administratoren
- Elastizität
- Virtuelle Ressourcenpools
- Technische Infrastruktur (VM, Storage, Network)

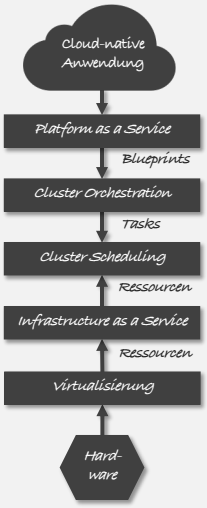
Rechenzentrum
- Rechner
- Netzwerk
- Storage

Diese Pyramide werden wir uns nach oben durcharbeiten.

TECHNISCHE HOCHSCHULE LÜBECK

TECHNOLOGIE - EBENEN

Oder: Wie kommt die Software an das Blech?

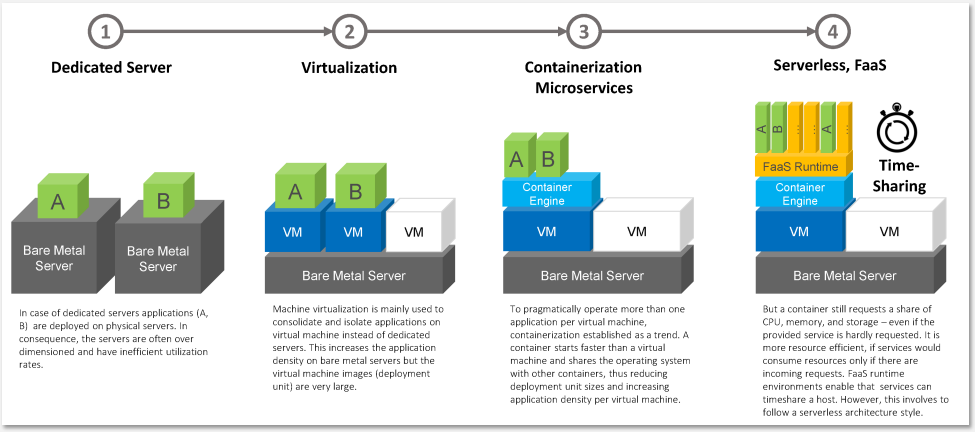


TECHNISCHE HOCHSCHULE LÜBECK

Diese Ebenen werden wir aus verschiedenen Blickwinkeln beleuchten.

HISTORISCHE ENTWICKLUNG

Eine kurze Geschichte der Cloud



Quelle: Kratzke, N. A Brief History of Cloud Application Architectures. Appl. Sci. 2018, 8, 1368.

TECHNISCHE HOCHSCHULE LÜBECK

Im Verlaufe der Zeit sind die Größen von Deployment Units geschrumpft!

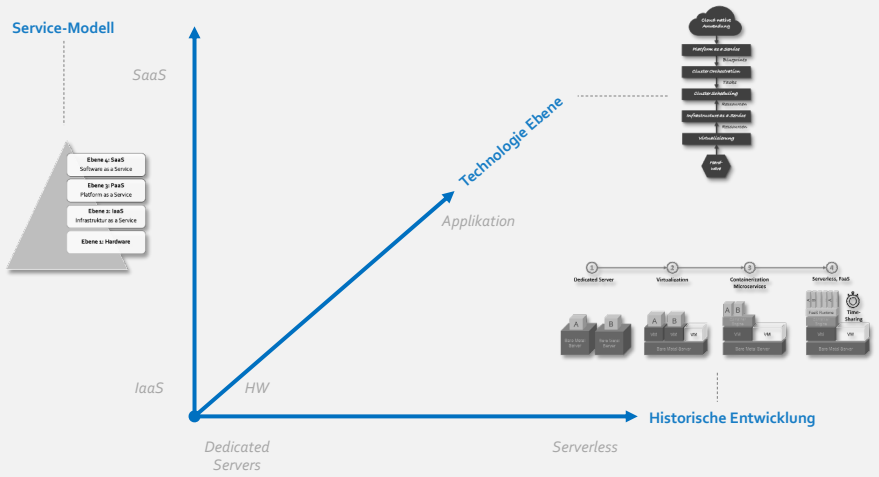
VM -> Container -> Function

Parallel dazu haben sich Servic-orientierte Architekturen (weiter)entwickelt.

Microservices + Serverless Architectures

ORIENTIERUNG IN DIESEM MODUL

Bezugssystem



TECHNISCHE HOCHSCHULE LÜBECK

Hinweis:
Dazu kommt noch eine architekturelle Ebene. Diese wird aber im Schwerpunkt im Folgemodul „Cloud-native Architekturen" behandelt werden.

INHALTE

- Cloud Computing
- Cloud-Service Modelle
- Cloud-Ökonomie (Pay-as-you-Go)
- Eine kurze Architekturgeschichte der Cloud
- Cloud-native Anwendungen und Dienste
- Zusammenfassung

TECHNISCHE HOCHSCHULE LÜBECK

CLOUD COMPUTING

The NIST Definition of Cloud Computing

Cloud computing is a model for enabling

- ubiquitous,
- convenient,
- on-demand network access

to a **shared pool of configurable computing resources** that can be

- rapidly provisioned and released
- with minimal management effort or service provider interaction.

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

TECHNISCHE HOCHSCHULE LÜBECK

This cloud model is composed of

- *five essential characteristics*
- *three service models,*
- *and four deployment models.*

Resources: *e.g., networks, servers, storage, applications, and services*

CLOUD COMPUTING

Essential Characteristics

On-demand self-service:

A consumer can provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with the cloud service provider.

Broad network access:

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous client platforms.

Rapid elasticity:

Capabilities can be elastically provisioned and released to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

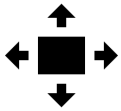
Measured service:

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Resource pooling:

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. The customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).

TECHNISCHE HOCHSCHULE LÜBECK



CLOUD COMPUTING

Service Models

Software as a Service (SaaS)

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings

Platform as a Service (PaaS)

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS)

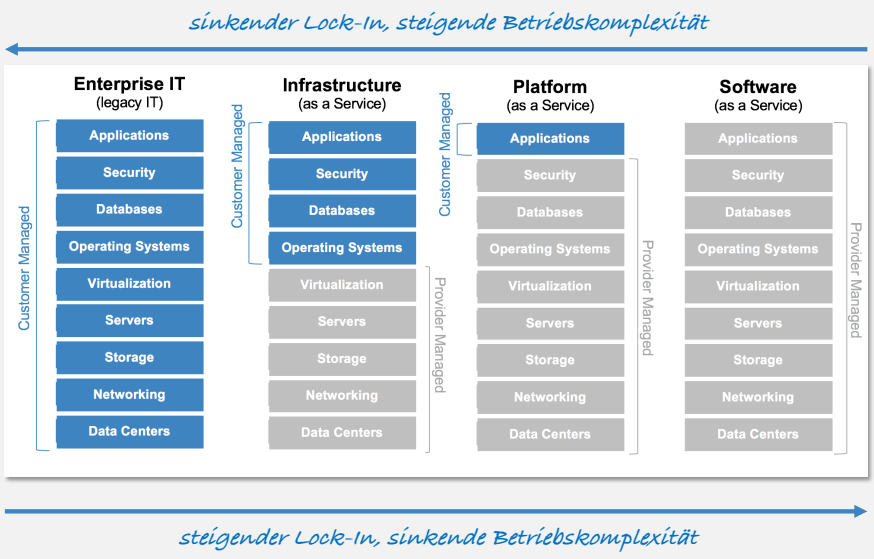
The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

TECHNISCHE HOCHSCHULE LÜBECK

CLOUD-SERVICE MODELLE

Customer Managed vs. Provider Managed Services



TECHNISCHE HOCHSCHULE LÜBECK

Mittels Cloud-Computing lassen sich Teile der IT-basierten Wertschöpfung an externe Dienstleister (Cloud-Provider) auslagern.

Von IaaS über PaaS zu SaaS wird dieser ausgelagerte Anteil dabei immer größer (ebenso wie die Gewinnmargen der Provider).

Alle Service-Modelle folgen dabei denselben wirtschaftlichen Gesetzmäßigkeiten.

CLOUD COMPUTING

Deployment Models

Private cloud

The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud

The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud

The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud

The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

TECHNISCHE HOCHSCHULE LÜBECK

Hinweis:
Wenn von Cloud Computing gesprochen wird, wird häufig Public Cloud Computing gemeint, ohne dies explizit zu sagen.

Vendor Lock-In Bedenken werden häufig durch Private Cloud Ansätze beantwortet (auch wenn das wirtschaftlich nicht immer sinnvoll ist).

INHALTE

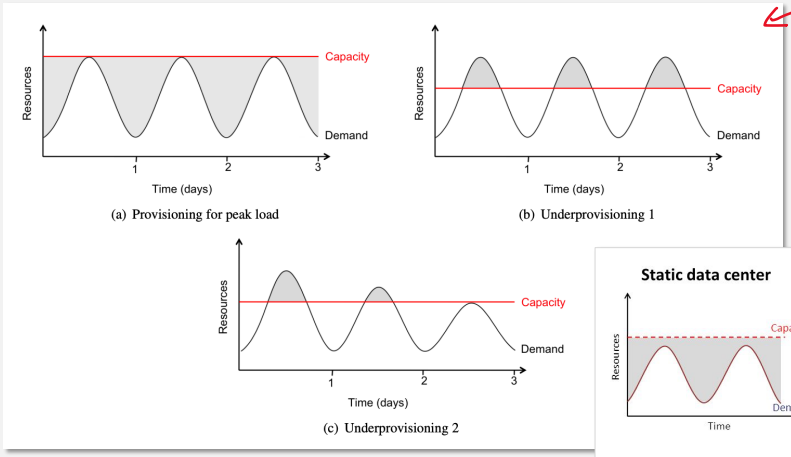
- Cloud Computing
- Cloud-Service Modelle
- Cloud-Ökonomie (Pay-as-you-Go)
- Eine kurze Architekturgeschichte der Cloud
- Cloud-native Anwendungen und Dienste
- Zusammenfassung

TECHNISCHE HOCHSCHULE LÜBECK

PAY-AS-YOU-GO

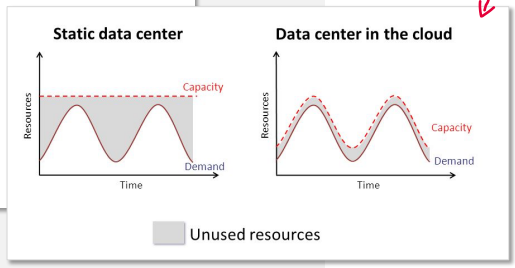
Nur für das Zahlen was man benötigt …

TECHNISCHE HOCHSCHULE LÜBECK



Das Dimensionierungs-problem in „klassischen" On-Premise Rechenzentren.

Cloud Computing ermöglicht Lastkurven enger zu folgen (und Ressourcen zu sparen).



Quelle: Above the Clouds: A Berkley View on Cloud, Computing, Feb. 2009

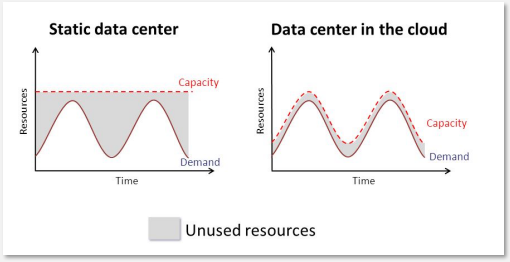
CRASHKURS IN CLOUD-ÖKONOMIE

Ist Cloud-Computing immer billiger?

TECHNISCHE HOCHSCHULE LÜBECK

Cloud-Ressourcen sind vor allem dann wirtschaftlich, wenn Lastschwankungen in einem Anwendungsfall auftreten.

Die Kosten pro Cloud-Ressource können sogar deutlich höher als die In-house Kosten liegen – solange das Verhältnis von Cloud zu In-house Kosten nicht das Verhältnis von Spitzen- zu Durchschnittslast übersteigt.



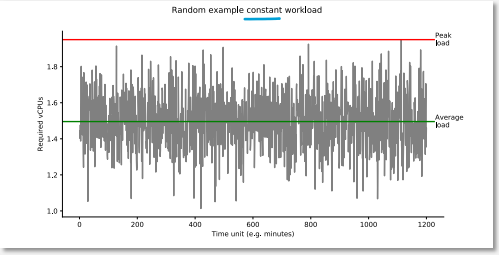
$$\frac{c}{d} < \frac{p}{a} \Leftrightarrow c < d\frac{p}{a}$$

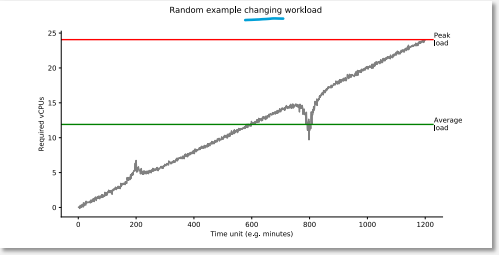
- d In-house Aufwand
- c Cloud-Kosten
- a Durchschnittslast (average)
- p Spitzenlast (peak)

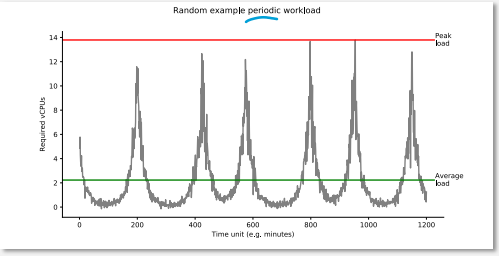
Quelle: J. Weinman, *Mathematical Proof of the Inevitability of Cloud Computing*, Jan. 2011
http://www.joeweinman.com/resources/joe_weinman_inevitability_of_cloud.pdf

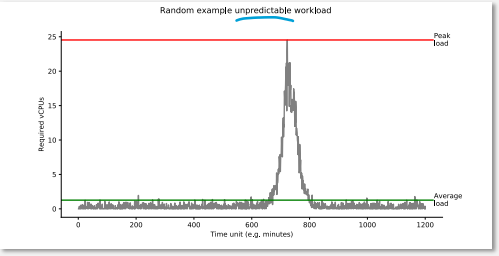
WORKLOAD KATEGORIEN

Constant, Changing, Periodic, Unpredictable (Once-in-a-lifetime)







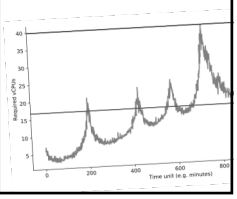


TECHNISCHE HOCHSCHULE LÜBECK

Diese Workload Kategorien werden im Cloud Computing normalerweise unterschieden.

Sie treten im „echten Leben" natürlich selten in Reinform auf.

Die meisten Workloads setzen sich aus diesen vier Grundtypen zusammen.



21

PIZZA-AS-A-SERVICE

Ein Beispiel zur Veranschaulichung der Gesetzmäßigkeiten der Cloud-Ökonomie

Legacy Service	IaaS	PaaS	SaaS
Esstisch	Esstisch	Esstisch	Esstisch
Getränke	Getränke	Getränke	Getränke
Ofen	Ofen	Ofen	Ofen
Belag	Belag	Belag	Belag
Tomatensauce	Tomatensauce	Tomatensauce	Tomatensauce
Pizzateig	Pizzateig	Pizzateig	Pizzateig
Zutaten	Zutaten	Zutaten	Zutaten
Pizza di Mama	*Kaufen & Backen*	*Pizza Service*	*Pizzeria*

TECHNISCHE HOCHSCHULE LÜBECK

selbst

fremd

Wir fragen uns nun, ob es Pizzakonsum-Gewohnheiten geben könnte, die Service-Modelle wirtschaftlich vorteilhafter erscheinen lassen könnten.

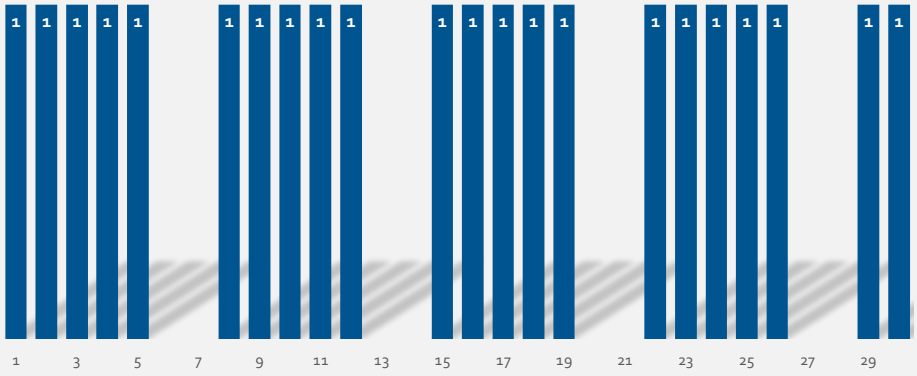
$$\frac{c}{d} < \frac{p}{a} \Leftrightarrow c < d\frac{p}{a}$$

22

STATISCHE WORKLOADS

Der regelmäßige „Fastfood" Workload

Der Cloud-Provider kann bis zu **30%** teurer sein als Ihre selbstgemachte Pizza.



TECHNISCHE HOCHSCHULE LÜBECK

Sie kaufen sich an jedem Werktag zur Mittagszeit eine Pizza am Stand gegenüber vor Ihrer Arbeitsstelle.

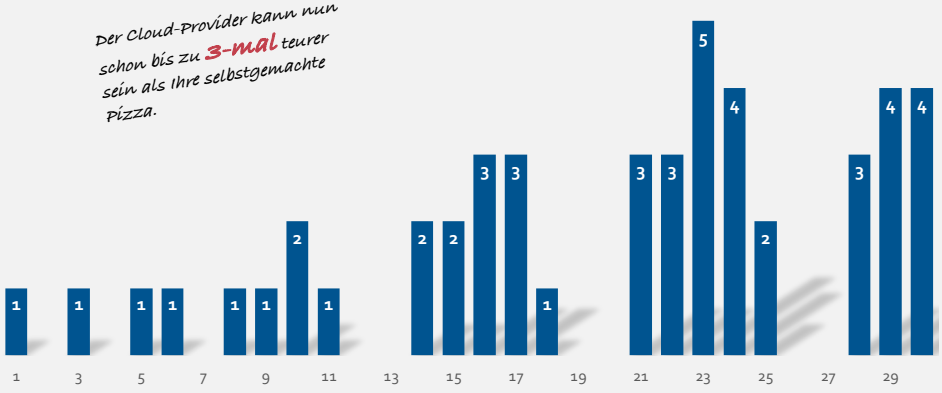
An Weochenenden tun Sie das natürlich nicht.

$$p = 1$$
$$a = {}^{22}/_{30}$$
$${}^{p}/_{a} \approx 1.3$$

CONTINUOUSLY-CHANGING WORKLOADS

Der Trendsetter-Workload

Der Cloud-Provider kann nun schon bis zu **3-mal** teurer sein als Ihre selbstgemachte Pizza.



TECHNISCHE HOCHSCHULE LÜBECK

Sie bringen Ihren Kollegen immer was vom Pizzawagen mit.

Das spricht sich rum, und Woche für Woche müssen Sie mehr Pizza besorgen.

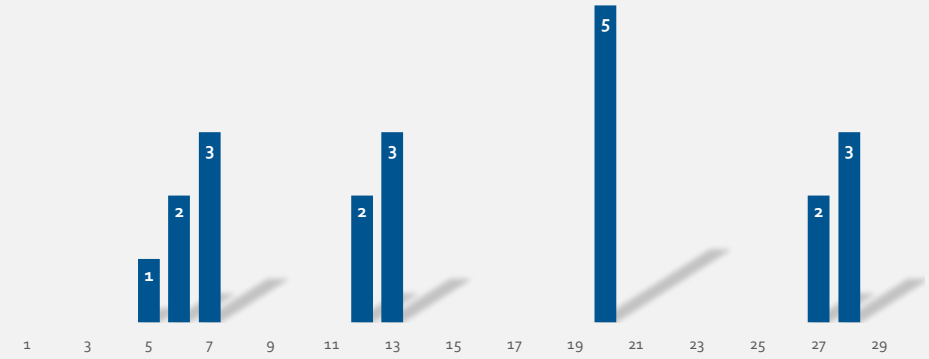
An Wochenenden arbeitet natürlich niemand.

$$p = 5$$
$$a = {}^{48}/_{30}$$
$${}^{p}/_{a} \approx 3.1$$

PERIODISCHE WORKLOADS

Der DVD/Party-Workload

Ihr Bedarf wird seltener und der Cloud-Provider kann nun schon bis zu 7-mal teurer sein als Ihre selbstgemachte Pizza.



TECHNISCHE HOCHSCHULE LÜBECK

Sie machen mit Familie und Freunden an Wochenenden DVD-Abende und reichen dazu Pizza.

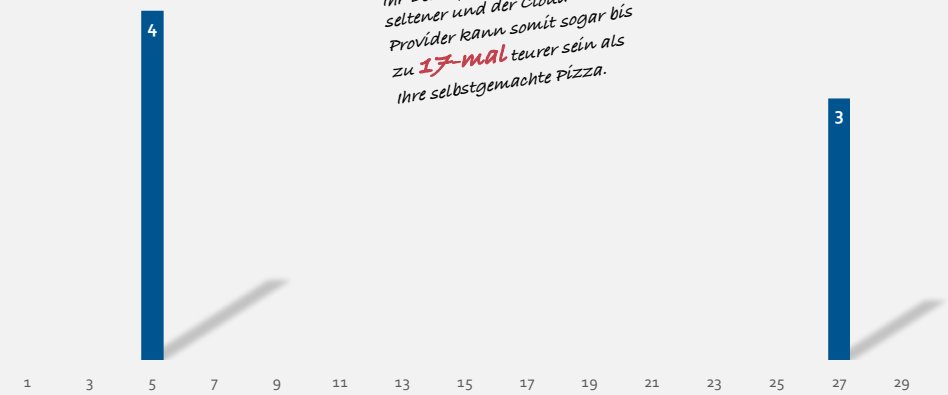
Unter der Woche haben Sie dazu natürlich keine Zeit.

$$p = 5$$
$$a = {}^{21}/_{30}$$
$$p/a \approx 7.1$$

UNVORHERSEHBARE/SELTENE WORKLOADS

Der Pizzeria-Workload

Ihr Bedarf wird nun noch seltener und der Cloud-Provider kann somit sogar bis zu 17-mal teurer sein als Ihre selbstgemachte Pizza.



TECHNISCHE HOCHSCHULE LÜBECK

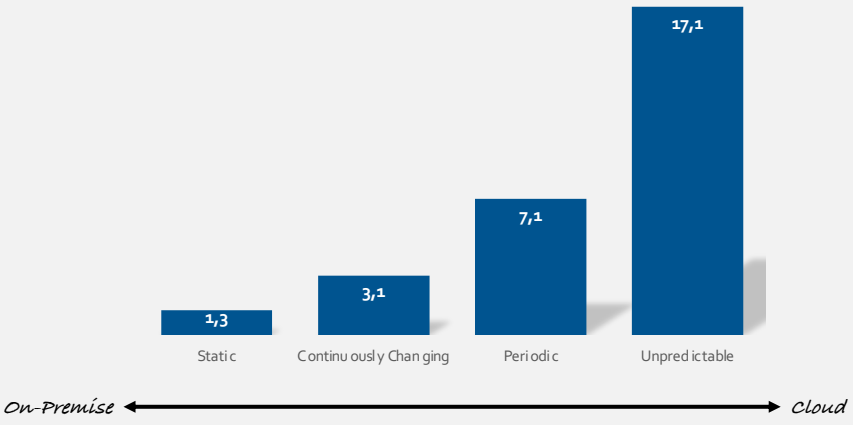
Sie laden Ihre Familie an Wochenenden ab und an in eine Pizzeria ein.

Unter der Woche haben Sie dazu natürlich keine Zeit.

$$p = 4$$
$$a = {}^{7}/_{30}$$
$$p/a \approx 17.1$$

KOSTENVORTEILE

entstehen in erster Linie durch den Workload



TECHNISCHE HOCHSCHULE LÜBECK

Kostenvorteile entstehen also in aller Regel durch den Workload und erst in zweiter Linie durch die Kostenstruktur des Dienstes.

Fun Fact:
AWS hat ca. 50% Marktanteil im Cloud Computing verlangt aber häufig bis zu 10% mehr pro Einheit für Services als bspw. Google.

Warum ist das so?

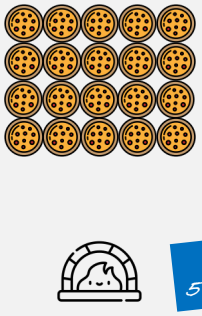
Vielleicht weil bei diesen Kostenvorteilen 10% Preisunterschiede kaum noch messbar sind?

FUN FACT

Es kostet übrigens dasselbe …

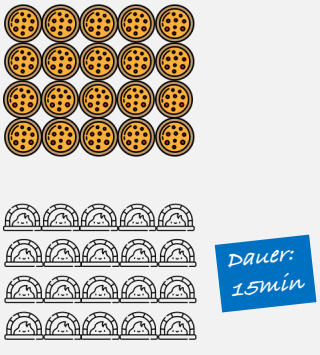
Diese Tatsache nennt sich **Kostenassoziativität**.

1 Ofen für 20 x 15 min zu mieten



Dauer: 5 Stunden

20 Öfen für 15 min zu mieten



Dauer: 15min

TECHNISCHE HOCHSCHULE LÜBECK

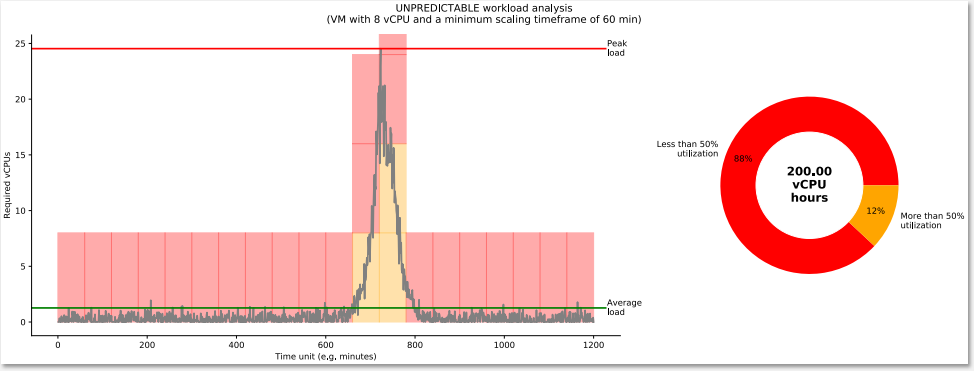
Cloud-Ökonomie kreativ nutzen

Bei welchem Lieferdienst würden Sie 20 Pizzen bestellen?

- Bei dem der Sie in 5 Stunden beliefert und bei dem 19 Pizzen kalt sind?
- Bei dem, der Ihnen nach 15 Minuten 20 warme Pizzen liefern kann?
- Wieviel Aufpreis wäre Ihnen das wert?
- Wieviel Mehraufwand kostet das den Lieferdienst?
- Wie häufig brauchen Sie als Lieferdienst wohl 20 Öfen gleichzeitig?

DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Am Beispiel der Unpredictable Workload Kategorie



Wir sehen in diesem Modell recht ineffizient genutzte VMs. Aber ungenutzte VMs kosten dasselbe wie genutzte VMs. Die Ursache liegt letztlich in zu „großen Kästen".

An welchen Stellschrauben kann man also drehen, um die Ressourcennutzung effizienter zu machen (also die Kästen kleiner zu machen)?

Das ist im Wesentlichen der Kostentreiber!!!

TECHNISCHE HOCHSCHULE LÜBECK

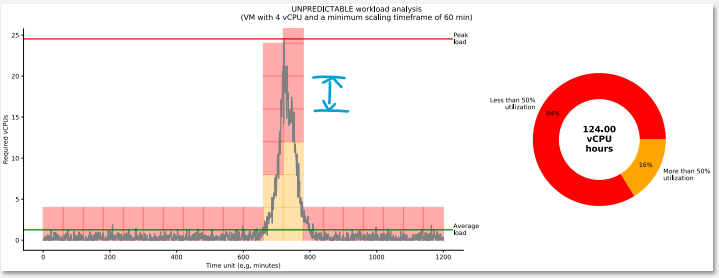
Ausgangssituation:
Nutzung von großen virtuellen Maschinen für „lange" Zeiträume

VM: 8 vCPU
Dauer: 60 min

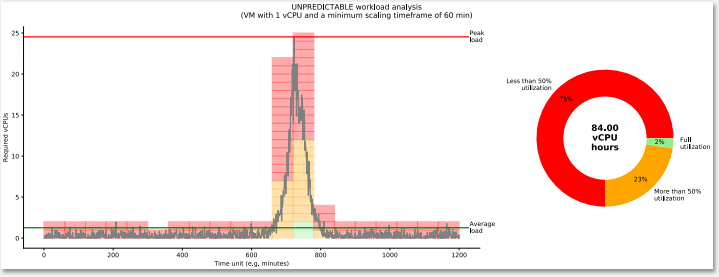
Resultierende Ressourcen-anforderung:
200 CPU-Stunden

DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Was passiert wenn wir die VM Größe reduzieren?



VM: 4 vCPU
Dauer: 60 min
=> 124 CPU-Stunden



VM: 1 vCPU
Dauer: 60 min
=> 84 CPU-Stunden

Stellschraube 1:
VM-Größe

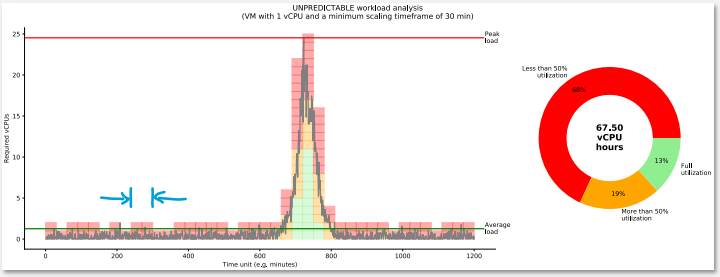
TECHNISCHE HOCHSCHULE LÜBECK

Mit kleineren VMs benötigt man zwar mehr VMs, aber fordert dennoch weniger CPU-Stunden insgesamt an.

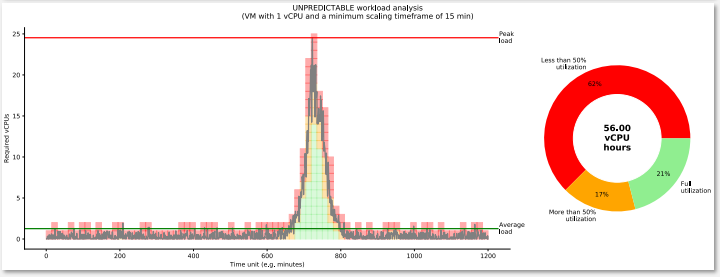
*Mit kleineren VMs kann man Lastkurven also **„enger"** folgen und verschwendet **weniger** ungenutzte Ressourcen.*

DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Was passiert wenn wir die Zuteilungsdauer reduzieren?



VM: 1 vCPU
Dauer: 30 min
=> 67.5 CPU-Stunden



VM: 1 vCPU
Dauer: 15 min
=> 56 CPU-Stunden

Stellschraube 2:
Zeit
(Abrechnungsintervalle)

TECHNISCHE HOCHSCHULE LÜBECK

*Mit kürzeren Zuteilungsdauern kann man Lastkurven also **„schneller"** folgen und verschwendet **kürzer** ungenutzte Ressourcen.*

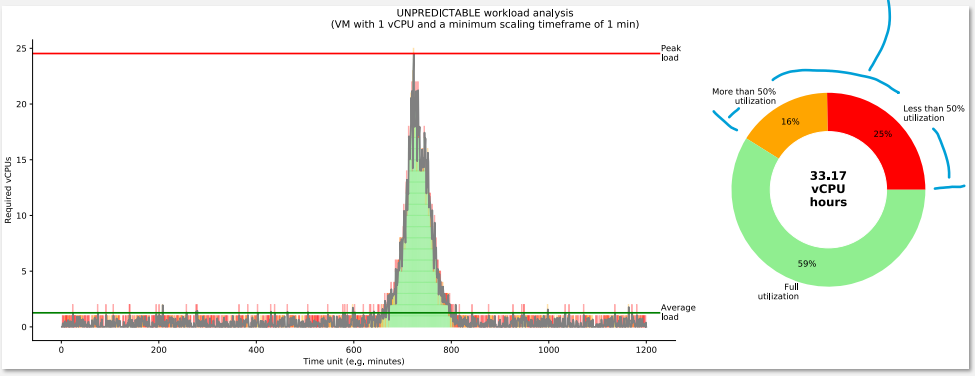
Alle Cloud Provider haben mit einer Stunden-basierten Abrechnung begonnen, und haben dann auf 30min, 15min, 5min, 1minAbrechnungsinter valle umgestellt (einige auch bereits auf Sekunden-basis).

Den maximalen „Effekt" den Sie damit erzielen können, sehen Sie auf der linken Seite!

DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Wo liegen die Grenzen dieses IaaS-fokussierten Ansatzes?

Hier scheint noch Weiteres Optimierungs-potenzial zu liegen.



TECHNISCHE HOCHSCHULE LÜBECK

VM: 1 vCPU
Dauer: 1 min

⇒ ~~33.17~~ CPU-Stunden

Um in diesen Bereich vorzudringen müssen wir allerdings noch schneller und noch feingranularer skalieren können.

- Container (Sub CPU Scale)
- FaaS (Sub Second Scale)

JUPYTER NOTEBOOK

Nur Versuch macht kluch ...

GitLab

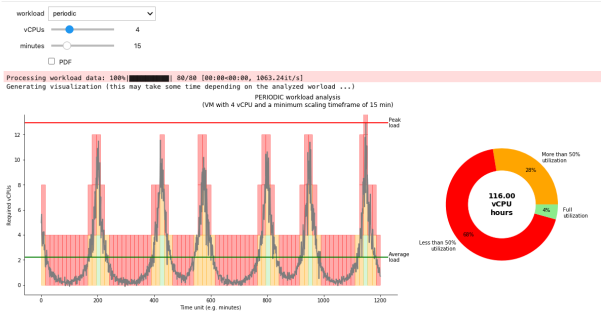
TECHNISCHE HOCHSCHULE LÜBECK

Klonen Sie dieses Repository (in Jupyter Lab, `https://jupyter.mylab.th-luebeck.de`):

```
git clone https://git.mylab.th-luebeck.de/cloud-native/lab-workload-analysis.git
```

Workload Analyse

Übung 1:
Workload Arten
Übung 2:
Periodische Workloads
Übung 3:
Unpredictable Workloads
Übung 4:
Weitere Workloads



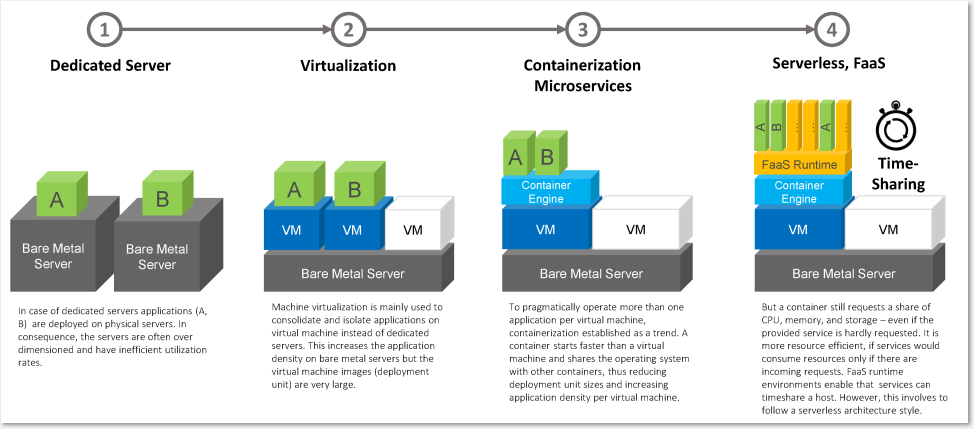
INHALTE

TECHNISCHE HOCHSCHULE LÜBECK

- Cloud Computing
- Cloud-Service Modelle
- Cloud-Ökonomie (Pay-as-you-Go)
- Eine kurze Architekturgeschichte der Cloud
- Cloud-native Anwendungen und Dienste
- Zusammenfassung

EINE KURZE GESCHICHTE DER CLOUD

Die Verkleinerung von Deployment Units im Verlaufe der Zeit



Quelle: Kratzke, N. A Brief History of Cloud Application Architectures. Appl. Sci. 2018, 8, 1368.

TECHNISCHE HOCHSCHULE LÜBECK

Im Verlaufe der Zeit sind die Größen von Deployment Units geschrumpft!

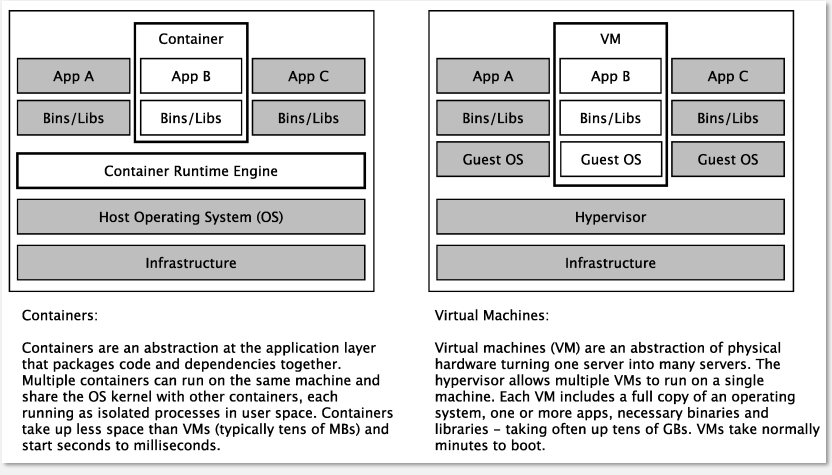
VM -> Container -> Function

Parallel dazu haben sich Servic-orientierte Architekturen (weiter)entwickelt.

Microservices + Serverless Architectures

EINE KURZE GESCHICHTE DER CLOUD

Containerisierung



Containers:

Containers are an abstraction at the application layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (typically tens of MBs) and start seconds to milliseconds.

Virtual Machines:

Virtual machines (VM) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, one or more apps, necessary binaries and libraries – taking often up tens of GBs. VMs take normally minutes to boot.

TECHNISCHE HOCHSCHULE LÜBECK

Container nutzen Betriebssystem-Virtualisierung (Prozessisolation) anstelle von Maschinen-Virtualisierung.

Nachteil:
Container-OS und Host-OS müssen identisch sein (meist Linux).

Vorteil:
Container starten sehr viel schneller und können wesentlich kleiner sein.

INHALTE

- Cloud Computing
- Cloud-Service Modelle
- Cloud-Ökonomie (Pay-as-you-Go)
- Eine kurze Architekturgeschichte der Cloud
- Cloud-native Anwendungen und Dienste
- Zusammenfassung

TECHNISCHE HOCHSCHULE LÜBECK

THE EIGHT FALLACIES OF DISTRIBUTED COMPUTING

Bill Joy + Tom Lyon (1 – 4) , *Peter Deutsch (5- 7)*, *James Gosling (8)*

Diese Liste der sogenannten **Irrtümer der verteilten Datenverarbeitung** sind eine Sammlung eigentlich trivialer, aber doch häufiger fehlerhafter Annahmen, die Entwickler (häufig unbewusst) voraussetzen, wenn sie insbesondere das erste Mal eine verteilte Anwendung entwickeln.

Die Liste der Fallacies of Distributed Computing kommt ursprünglich von **Sun Microsystems** und wurde dort von Bill Joy und Tom Lyon mit vier Trugschlüssen eröffnet. Weite Bekanntheit erlangten sie 1994 durch Peter Deutsch, der sie zu sieben Trugschlüsse erweiterte und als "The Seven Fallacies of Distributed Computing" veröffentlichte. James Gosling, ebenfalls von Sun, setzte ungefähr 1997 noch den achten Punkt dazu.

(1) Das Netzwerk ist ausfallsicher
(2) Die Latenzzeit ist gleich null
(3) Der Datendurchsatz ist unbegrenzt
(4) Das Netzwerk ist sicher
(5) Die Netzwerktopologie wird sich nicht ändern
(6) Es gibt immer nur einen Netzwerkadministrator
(7) Die Kosten des Datentransports können mit null angesetzt werden
(8) Das Netzwerk ist homogen

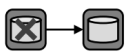
TECHNISCHE HOCHSCHULE LÜBECK

SUN Microsystems

ist die Firma, die Java (1996, V1.0) entwickelt hat und 2009/2010 von Oracle übernommen wurde.

CLOUD-NATIVE APPLIKATIONEN

Das IDEAL-Modell (Von Service-orientierten Architekturen inspiriert)



I solated State

A maximimum of cloud application components should be stateless. They do not handle:

Session State: State of the communication with the application.

Application State: Business data handled by the application.

D istribution

Cloud applications are split up into multiple components

to utilize multiple cloud resources

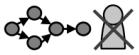
because the cloud itself is a large distributed system.

E lasticity

Cloud applications are scaled by adjusting resource numbers (scaling out) – not by scaling up:

Scale out (horizontally): Increase performance by adding more resources.

Scale up (vertically): Increase performance by improving existing resources.



A utomated

Management tasks during runtime have to be handled quickly.

Example: Cost reduction by adjusting pay-per-use resource numbers automatically.

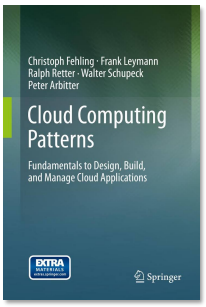
Example: Automatic reaction to resource failures.

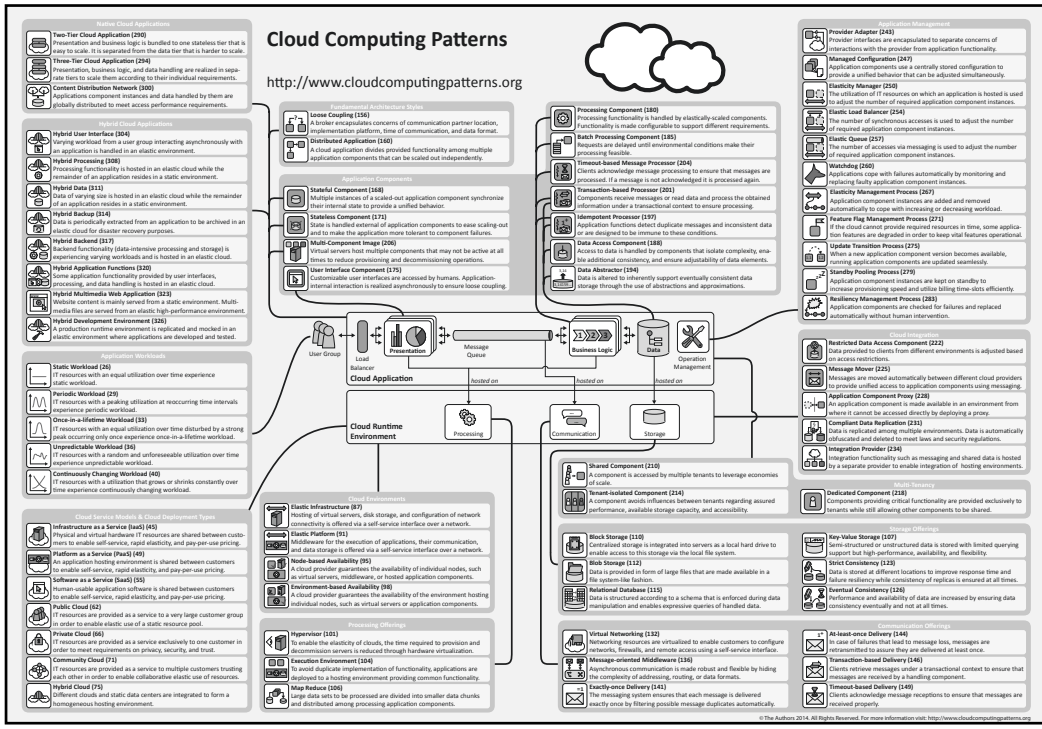
L oose Coupling

Cloud-native application components should not influence each other regarding factors such as availability, data format, data exchange rate.

Example: Failure of one application component does not cause failure of other components.

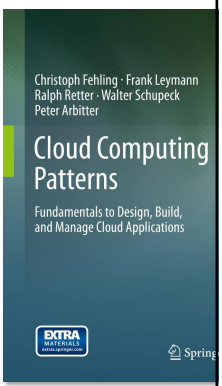
TECHNISCHE HOCHSCHULE LÜBECK





TECHNISCHE HOCHSCHULE LÜBECK

Das IDEAL-Modell „nur" etwas detaillierter ;-)



CLOUD-NATIVE APPLIKATIONEN

Definition

CNCF Cloud Native Definition v1.0

Cloud native Technologien ermöglichen es Unternehmen, skalierbare Anwendungen in modernen, dynamischen Umgebungen zu implementieren und zu betreiben. Dies können öffentliche, private und Hybrid-Clouds sein. Best-Practises, wie Container, Service-Meshs, Microservices, immutable Infrastruktur und deklarative APIs, unterstützen diesen Ansatz.

Die zugrundeliegenden Techniken ermöglichen die Umsetzung von entkoppelten Systemen, die belastbar, handhabbar und beobachtbar sind. Kombiniert mit einer robusten Automatisierung können Softwareentwickler mit geringem Aufwand flexibel und schnell auf Änderungen reagieren.

Source: Cloud-native Computing Foundation, https://github.com/cncf/toc/blob/master/DEFINITION.md

Industry point of view

Understanding Cloud-native Applications after 10 Years of Cloud Computing

A cloud-native application (CNA) is a distributed, elastic and horizontally scalable system composed of loosely-coupled (micro)services. A CNA isolates state in a minimum of stateful components.

The application and each self-contained deployment unit of that application are designed according to cloud-focused design patterns and operated on a self-service elastic platform.

Source: Kratzke, Nane and Quint, Peter-Christian. Understanding Cloud-native Applications after 10 Years of Cloud Computing - A Systematic Mapping Study, in Journal of Systems and Software, Elsevier, 2017, DOI: 10.1016/j.jss.2017.01.001

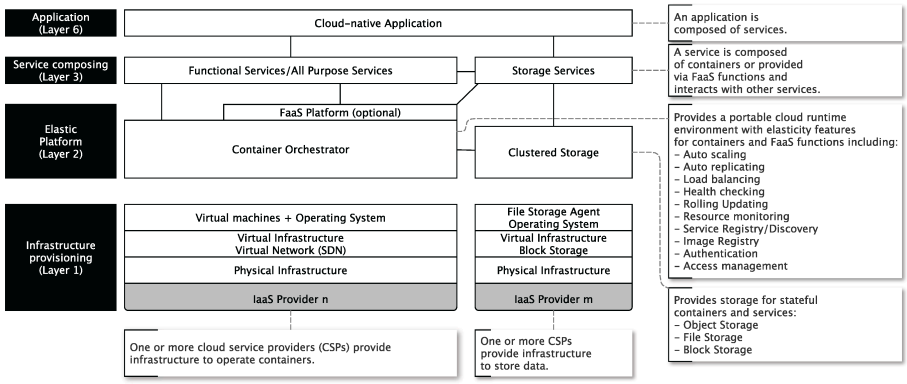
Software engineering research point of view

TECHNISCHE HOCHSCHULE LÜBECK

Oft zu hören: *Cloud-native Applikationen sind solche, die absichtsvoll für Cloud-Infrastrukturen und -Plattformen entwickelt werden (also bewusst nirgendwo anders laufen sollen).*

CLOUD-NATIVE APPLIKATIONEN

Referenzmodell einer Cloud-native Applikation



Quelle: Kratzke, N. A Brief History of Cloud Application Architectures. Appl. Sci. 2018, 8, 1368.

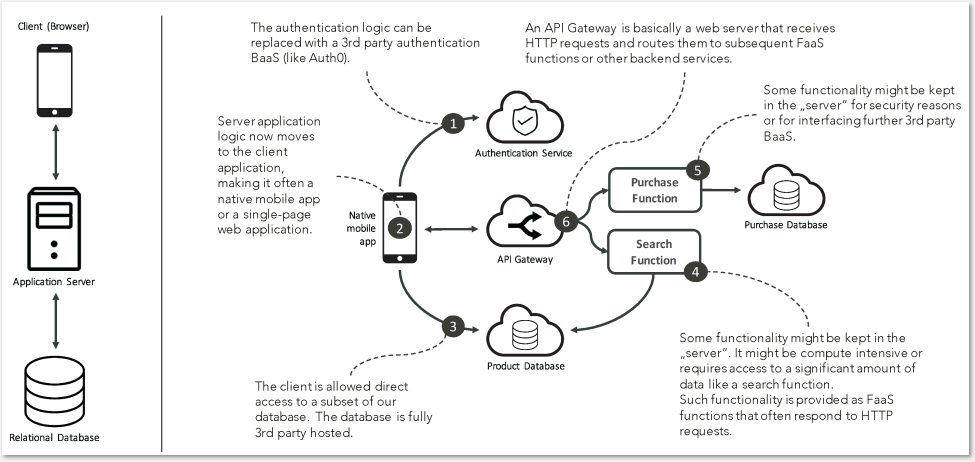
TECHNISCHE HOCHSCHULE LÜBECK

CNA folgen häufig diesem Referenzmodell:

- *Service of Services*
- *Isolierter Storage*
- *Container oder FaaS Functions*
- *Microservice oder Serverless Architektur*

SERVERLESS ARCHITEKTUREN

Containerisierung in Cloud-nativen Anwendungen zu Ende gedacht



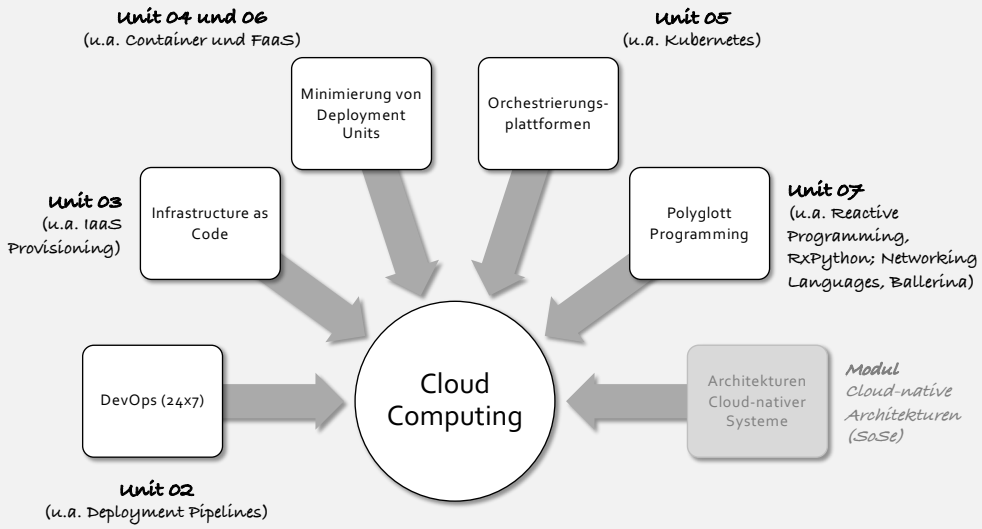
Quelle: Kratzke, N. A Brief History of Cloud Application Architectures. Appl. Sci. 2018, 8, 1368.

INHALTE

- Cloud Computing
- Cloud-Service Modelle
- Cloud-Ökonomie (Pay-as-you-Go)
- Eine kurze Architekturgeschichte der Cloud
- Cloud-native Anwendungen und Dienste
- Zusammenfassung

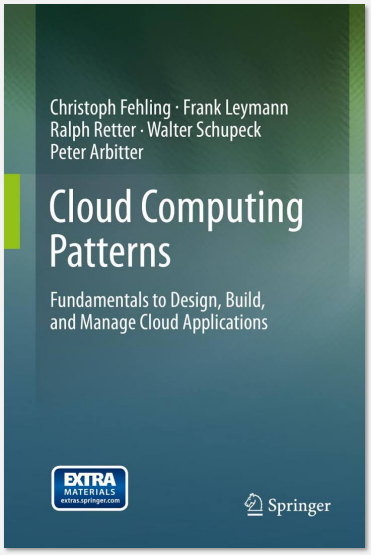
ZUSAMMENFASSUNG

und Ausblick auf die Module Cloud-native Programmierung und Cloud-native Architekturen



TECHNISCHE HOCHSCHULE LÜBECK

ZUM NACHLESEN



TECHNISCHE HOCHSCHULE LÜBECK

ZUM NACHLESEN

Paper

- Armbrust et al.; **Above The Clouds – A Berkley View on Cloud Computing (2009)**
  https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf
- Mell, Grance; **NIST Definition of Cloud Computing (2011)**
  https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
- Weinman; **Mathematical Proof of the Inevitability of Cloud Computing (2011)**
  http://www.joeweinman.com/resources/joe_weinman_inevitability_of_cloud.pdf
- Kratzke, Quint; **Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study (2017)**
  https://doi.org/10.1016/j.jss.2017.01.001
- Kratzke; **A Brief History of Cloud Application Architectures (2018)**
  https://www.mdpi.com/2076-3417/8/8/1368

TECHNISCHE HOCHSCHULE LÜBECK

TOOL-CHAIN

Dieses Moduls

Unit 01:
- Python
- Jupyterlab (THL Managed Service)

Unit 02:
- Gitlab CI (.gitlab-ci.yml, THL Managed Service)

Unit 03:
- VirtualBox
- Vagrant
- Google Cloud Platform (Managed Service)
- Terraform

Unit 04:
- Docker

Unit 05:
- Kubernetes
- Google Kubernetes Engine (Managed Service)

Unit 06:
- Kubeless
- Google Cloud Functions (Managed Service)

Unit 07:
- RxPython (Reactive Programming)
- Ballerina (Network Programming)

TECHNISCHE HOCHSCHULE LÜBECK

Fast alle Programmierbeispiele basieren in diesem Modul auf Python.

Sie müssen aber kein Python-Experte sein, um diesen folgen zu können.

Komplexe Tool-Chain (typisch für CNA).

In den Praktika erhalten Sie die entsprechenden Quellen, wie diese zu installieren sind.

KONTAKT

Disclaimer

Nane Kratzke

+49 451 300-5549

nane.kratzke@th-luebeck.de

kratzke.mylab.th-luebeck.de