

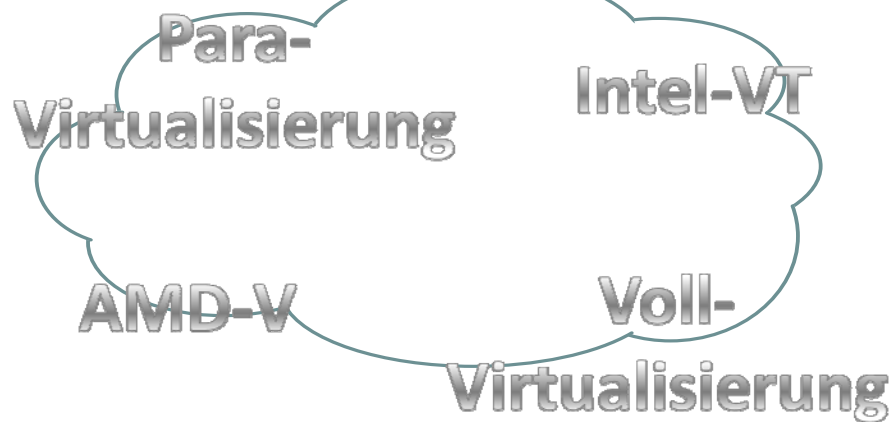


zimory

CLOUD, GRID und Virtualisierung

Benjamin Schmidt

Hintergrund



Vollvirtualisierung

- Sämtliche Hardwarekomponenten (u.a. Netzwerkkarte, SCSI-Controller) werden emuliert
- Erlaubt die Verwendung eines unmodifizierten Gastbetriebssystems
- Hohe Flexibilität
- Bei I/O-Operationen teilweise sehr langsam

Vollvirtualisierung mit Hardwareunterstützung

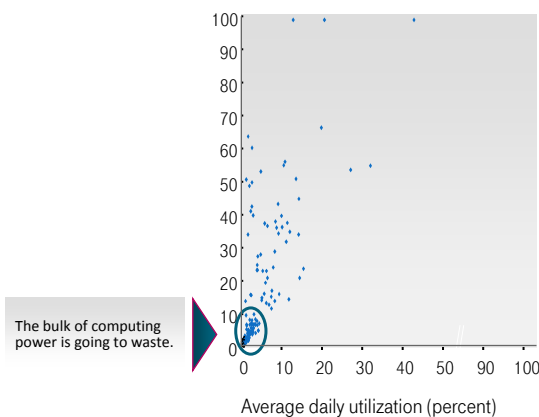
- verbesserte Virtualisierung über Befehlssatzerweiterungen für x86-Architektur
- Prozeduren direkt ("nativ") vom Prozessor des Host-Systems ausführen
- AMD-V bzw. Intel VT: nicht miteinander kompatibel
- Grafikkarte und Netzwerkkarte immer emuliert

➤ Secure Virtual Machine

Paravirtualisierung

- Hardwaretreiber des Gastbetriebssystems werden durch proprietäre Treiber ersetzt
- Interagieren mit dem Hypervisor
- Optimiert Performanceverlust verursacht durch Emulierung
- Benötigt angepasste Version des Gastbetriebssystems; nicht möglich für Windows

Rechenzentrum Virtualisierung



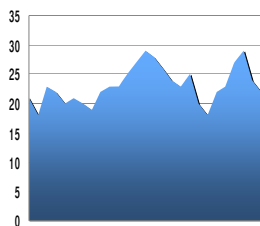
Challenge #1

- Rigid data centre structure renders most of the computing power unused most of the time.
- As maintenance costs keep running, the efficiency is decreased.
- Energy consumption for idle time affects the environmental balance.
- Virtualization technology and cloud computing allow to put unused computing power on a market.

Rechenzentrum Virtualisierung

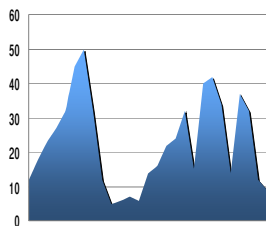
Traditional IT

- centralized application architecture
- traditional customer and user models
- high base utilization of servers



Enterprise 2.0

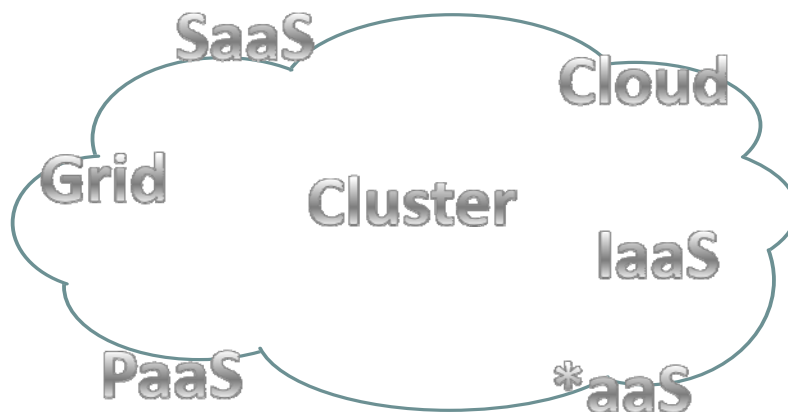
- dynamic services, software-as-a-service
- distributed applications
- dynamic usage profiles



Challenge #2

- As the business follows new patterns, customers demand increased flexibility for data centre services.
- Cloud computing enabled by reliable virtualization techniques delivers this flexibility.

Von der Virtualisierung zur Cloud



Grid/Cloud - Gemeinsamkeiten

- Dienstleistungsbezogenen Ansatz
- Produkt: „Rechenleistung“
- Vorhandene Hardware wie Recheneinheiten und Datenspeicher zu vernetzen

- Ziel: Rechenleistung wie Strom aus der Steckdose zu beziehen

Grid/Cloud – Abgrenzung Cluster

- Grid:
 - losen Kopplung
 - Heterogenität
 - geographischen Zerstreuung der Computer
 - Kommunikation über Inter-/Intranet (WAN)

- Cluster:
 - Enge Kopplung
 - Meistens homogen (sogar Hardware)
 - Von außen ein Computer
 - Kommunikation über dediziertes Netzwerk

Grid/Cloud - Rechenleistung

- Grid Computing:
 - Anbieten von paralleler Rechenleistung
 - ein Job soll einen großen Anteil der gesamten Leistung beanspruchen
- Cloud Computing:
 - Rechenleistung wird unter vielen Benutzern aufgeteilt
 - jede Anforderung soll auf einen kleinen Anteil der verfügbaren Ressourcen limitiert werden
 - „Advanced hosting platform“

Grid/Cloud - Administration

- Grid computing:
 - Fördert die Konsolidierung von zusammenarbeitenden aber administrativ getrennten Einheiten
- Cloud computing:
 - Wird normalerweise von einer einzigen administrativen Entität betrieben

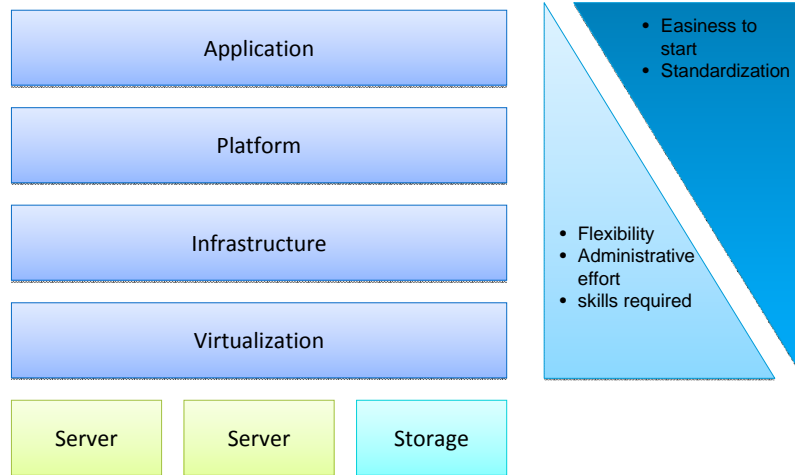
Grid/Cloud - Programmkomplexität

- Grid Computing:
 - Programme sind relativ kompliziert und müssen speziell für die verteilte Abarbeitung geschrieben werden
 - Entwickler muß sich mit Kommunikation innerhalb des Grids selbst auseinandersetzen
- Cloud computing:
 - Beinhalten eigene Messaging Systeme innerhalb der einzelnen Komponenten und bieten nach Aussen offene APIs
 - Programme kommunizieren über APIs

Cloud Computing

- Zusätzlich Abstraktion der Virtualisierung
 - IT-ähnliche Ressourcen werden als Service angeboten
 - Benutzer des Services ist darunterliegende Technologie nicht bekannt
- Wolke

Cloud Computing Layer



Infrastructure as a Service

- Roheste Form Rechenleistung als Dienst zur Verfügung zu stellen
- Benutzer „mietet“ Hardwareressourcen (CPU, Memory, Storage, Netzwerk)
- beliebiges Betriebssystem
- Abrechnung erfolgt als „Pay-as-you-go“ Modell
- Selbstverantwortung : Ausfallsicherheit, Verfügbarkeit, Skalierbarkeit und Sicherheit
- Beispiel: Amazon EC2, Eucalyptus, Zimory

Platform as a Service

- IaaS + OS (+ Laufzeit- / Entwicklungsumgebung)
- Bietet kompletten Lifecycle einer Software
- Ausfallsicherheit, Verfügbarkeit, Skalierbarkeit und Sicherheit ist Providerverantwortung
- Nachteil: Provider / Entwicklungsumgebung lock-in

- Beispiel: Google App Engine, Force.com

Software as a Service

- Komplette Applikation beim Provider gehostet
- Multi-tenant Architektur

- Beispiel: Salesforce.com CRM-solution

Data (Storage) as a Service

- Online Backup- und Archivierungsdienst
- Im weiteren Sinne: jeder Internet-Storage-Dienst
- Im engeren Sinne: Daten, die einer Cloud Applikation zugewiesen sind

- Beispiel: Amazon S3

Cloud Hürden

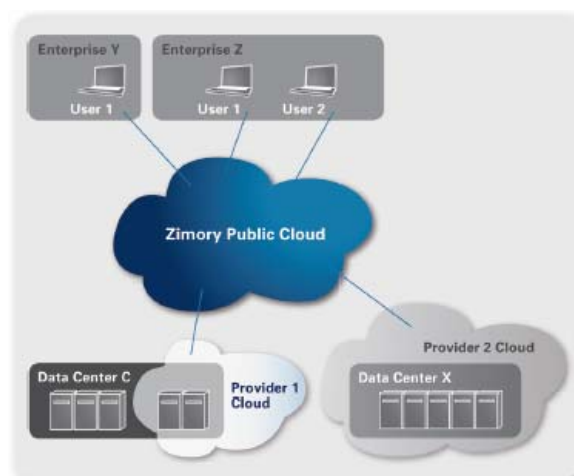
1. Availability of service
2. Data lock-in
3. Data confidentiality and auditability
4. Data transfer bottlenecks
5. Performance unpredictability
6. Scalable storage
7. Bugs in large distributed systems
8. Scaling quickly
9. Reputation fate sharing
10. Software licensing

Quelle: Michael Armbrust et al.; *Above the Clouds: a Berkeley View of Cloud Computing*

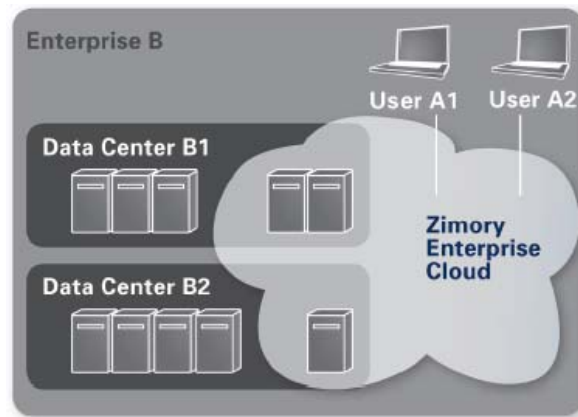
Cloud Formen

- Public Clouds
- Private/Enterprise Clouds
- Federated (Community) Clouds
- Hybrid Cloud

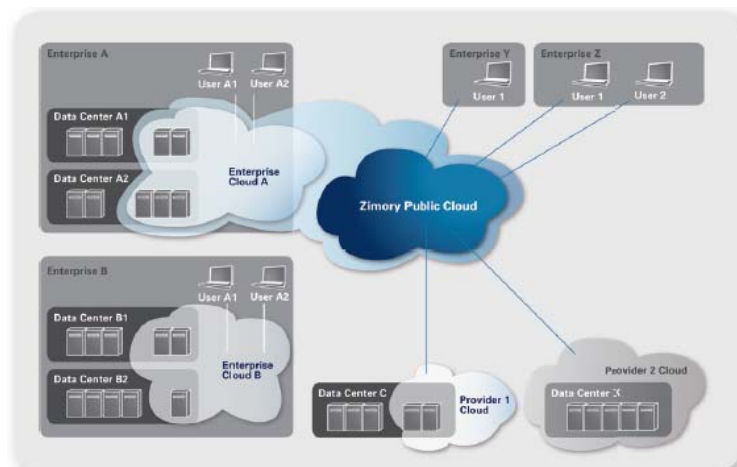
Public Cloud



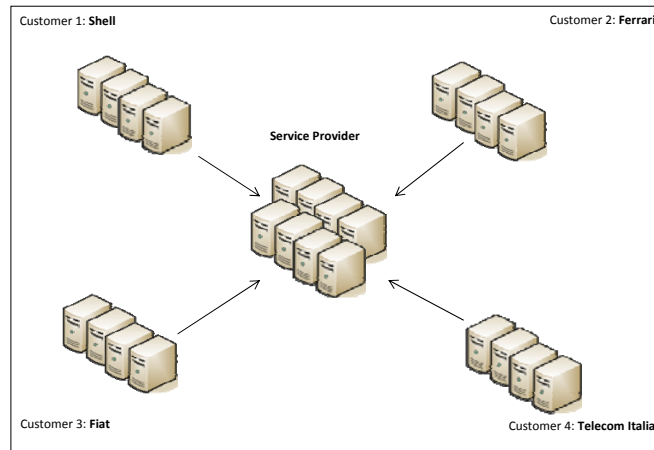
Private / Enterprise Cloud



Hybrid Cloud



Ausblick: Community Cloud



ZIMORY

schmidt@zimory.com