

Software-Defined Storage

... und das Software-Defined Datacenter

Dr. Felix Hupfeld, CTO Quobyte



Software-Defined Storage (SDS) und Datacenter (SDDC)

Agenda

- Hyperscaler - Vorbild und Konkurrenz
- Das „Geheim“-Rezept der Hyperscaler
- Wie sieht das Next-Generation Datacenter aus?

Hyperscaler - Vorbild und Konkurrenz

I wanna be like Google! I have to be like Google.

- Kosten
 - Hardware Plattform (Standard Server)
 - Betrieb (wenige Leute)
- Flexibilität
 - Große Bandbreite an Workloads auf einer Infrastruktur (keine Applikationssilos, keine Heterogenität)
- Agilität
 - Provisionierung in Sekunden (keine Wochen)

SDS/SDD Enabler

Server: Commodity ist High-Performance

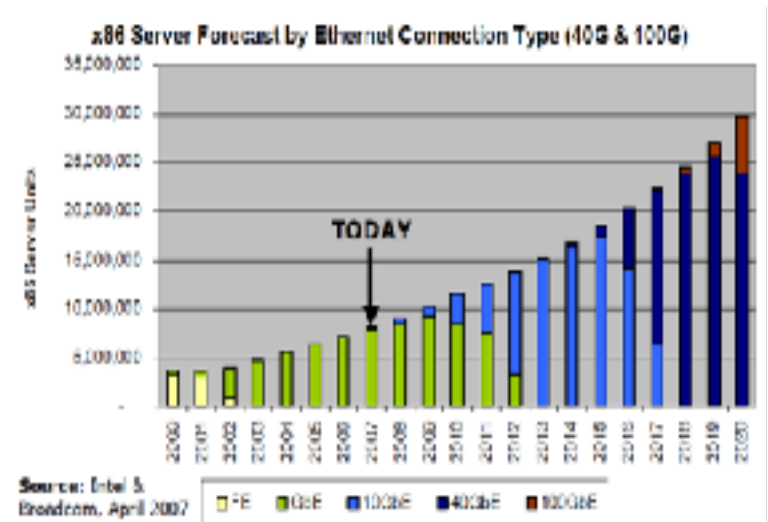
- Performance braucht keine Spezialhardware
- Basis: Standard Server Hardware



SDS/SDDC Enabler

Netzwerke: Schneller als Erwartet

- Alles ausser Ethernet (Infiniband, Omnipath) nur noch Spezialanwendungen
- Skalierbare Netzwerkfabrics
- Neue Architekturen
- Disaggregation:
 - rack-lokal obsolet



SDS/SDDC Enabler

Software-Defined Storage

- Disaggregation des Storage:
 - Lokaler Storage ist nicht schneller als entfernter Storage
 - Storage Konsumenten entkoppelt vom Storage System

NVMe 10 μ s, SSD 100 μ s, HD 10ms



NVMe 10 μ s, SSD 100 μ s, HD 10ms



10 μ s

Hyperscaler - Vorbild und Konkurrenz

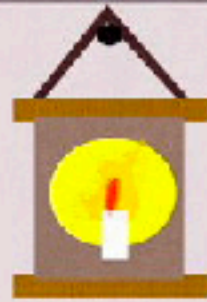
Software-Defined Datacenter - Das Rezept

Standard-Hardware

- Server
- Ethernet

+ Software und „Automatisierung“

= Software-Defined Datacenter à la Google



PHASE 1 PHASE 2 PHASE 3

Collect
underpants



Profit



Das Software-Defined Datacenter

Das „Geheim“-Rezept

- Software-Defined Datacenter
 - Software auf Standard-Hardware
 - „Automatisierung“
- Warehouse-scale Computer
 - fundamental andere Technologie
 - Verteilte Systeme, horizontal skalierbar (scale-out)
 - Entkopplung von Dienst und Hardware
 - Konsequente Fehlertoleranz
 - Disaggregation

Das „Geheim“-Rezept

Entkopplung: Fehlertoleranz

- Grundannahme: Hardware ist unzuverlässig
- Hardware ist nur noch Ressource: „pet vs. cattle“
- Was schiefgehen kann, wird schiefgehen

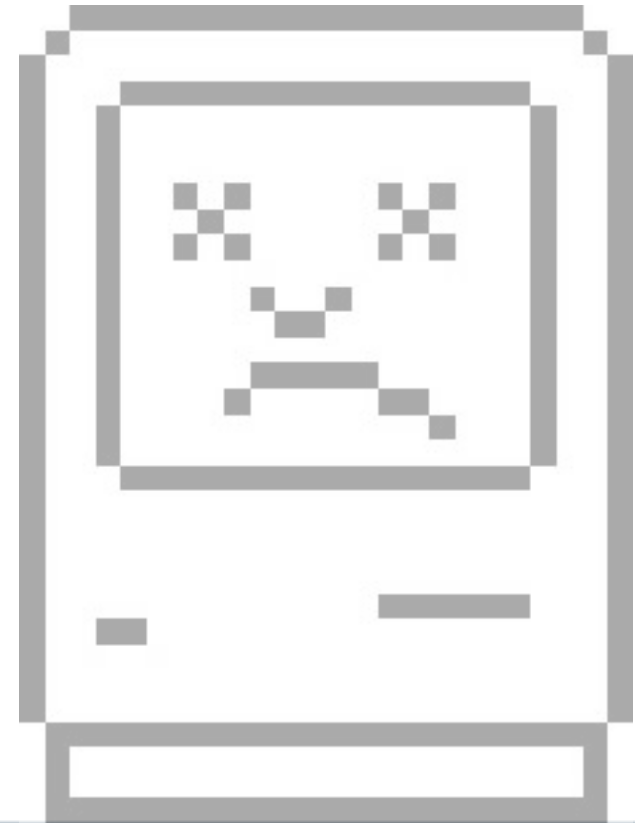
- Fehlertoleranz: umgehen eingeschränkter Fehler
 - Schutz nach rechts und links
 - Vollautomatisch
 - Ohne Fußnoten (partitionstolerant / split-brain safe)

- Schutz vor defekter Hardware
 - Schutz nach unten: Prüfsummen, Watchdogs

Das „Geheim“-Rezept

Theorie der Verteilten Systeme

- Traditionell: Idealistische Modelle
 - Synchroner Systeme
 - Perfekte Netze
 - 1990s: Realistische System Modelle
 - Fehler sind die Regel
 - Internet: Nachrichtenverlust, ...
 - Server: langsam, Abstürze, Reboot
- ➔ Alle Fehler behandeln
(ausser Boshaftigkeit)



Das „Geheim“-Rezept

Die Software

- Software, die aus vernetzten Maschinen ein System macht
 - fehlertolerant
 - verteilt
 - skalierbar
- Erste Ansätze:
 - OpenStack, Kubernetes, Docker, Mesos
 - Einzelne Software-Defined Storage Produkte
 - (Software-Defined Network für Isolation)

Next-Generation Storage

Ich bin nicht Google!

- Robust und hochverfügbar
- Einfacher
 - ... zu betreiben. Linux Applikation auf homogener Hardware
 - ... zu planen. Planungshorizont: 3 Monate
 - ... integrierter. It's a feature, not a product
 - ... universeller. Fast alle Workloads
- Flexibel
 - Viele Konfigurationen
- Skalierbar
 - ~2020 end of Moore's law - Bob Colwell (Intel, DARPA)

Vielen Dank!



google.com/+Quobyte



linkedin.com/company/quobyte-inc



twitter.com/quobyte