

Vom Utility Computing zum Cloud Computing

Lothar Wieske

DB Systel GmbH
Kleyerstr. 27
60326 Frankfurt am Main
lothar.wieske@deutschebahn.com

Abstract: Virtualisierung, Standardisierung und Automation waren und sind die Treiber für Utility Computing; beim Cloud Computing kommen Simplification, Commoditization und Federation dazu. Wer Cloud Computing erfolgreich betreiben und benutzen will, muss sich viel stärker mit den Geschäftsmodellen und ihren technischen und wirtschaftlichen Zusammenhängen beschäftigen. Zunächst werden Beispiele für Amazon und Salesforce vorgestellt. Abschließend werden Ideen und Konzepte von Public Clouds und Private Clouds abgeglichen insbesondere mit Blick auf Commodity Clouds und Enterprise Clouds.

1 Einleitung und Gedankengang

Cloud Computing lässt sich entlang dreier Dimensionen gliedern [MG09]. Fünf Schlüsselmerkmale beschreiben das WIE: On-Demand Self-Service, Broadband Network Access, Resource Pooling, Rapid Elasticity, Measured Service. Drei Angebotsformen gliedern das WAS: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS). Vier Bereitstellungsmodelle fächern schließlich das VON WEM und FÜR WEN auf: Private Cloud, Public Cloud, Community Cloud, und Hybrid Cloud.

Angebotsformen und Bereitstellungsmodelle spannen eine Matrix auf. In dieser Matrix betrachtet der vorliegende Beitrag einen Ausschnitt: Public/Private und IaaS/SaaS.

Für Pioniere von Public IaaS (Amazon) und Public SaaS (Salesforce) werden wichtige Kostenstrukturen umrissen; zweiseitige Märkte liefern den wirtschaftlichen Hintergrund für werbegetriebene Anbieter wie Facebook, Google und Twitter. Die technische Begründung für die hohe Fertigungstiefe von Public Clouds mit ihren Architekturen und eigenen Dateisystemen und Datenbanken liefert das CAP-Theorem.

Wirtschaftliche und technische Treiber der Public Cloud Provider ebenso wie deren Architekturen und Lösungen übertragen sich nicht einfach und direkt auf Private Clouds. Utility Computing hat mit Virtualisierung, Standardisierung und Automatisierung zur verbesserten Bereitstellung geführt, aber Konzepte rund um die Ressourcen nicht wesentlich angetastet. Cloud Computing geht hierbei unterschiedliche neue Wege und stellt Infrastrukturen, Plattformen und Software in weiter gefasste Zusammenhänge, insbesondere mit der Unterscheidung Commodity Cloud vs. Enterprise Cloud.

2 Wirtschaftliche Faktoren des Cloud Computing

Dieser Abschnitt trägt Daten und Fakten zusammen und identifiziert drei wirtschaftliche Faktoren: Größenvorteile bei Public IaaS (Amazon), Verbundvorteile bei Public SaaS (Salesforce) und zweiseitige Märkte für werbefinanzierte Provider (Facebook, Google).

2.1 Größenvorteile bei der Infrastruktur – Public IaaS

James Hamilton hat im Jahr 2010 die jährlichen Gesamtkosten eines Rechenzentrums mit ca. 46.000 Servern auf ca. 42,3 Mio. \$ geschätzt und kommt zu drei Kernaussagen. Die Anschaffungskosten für Server und Netzwerk dominieren mit 65%. Die Anschaffungskosten der Ausstattung für den Strom (Verteilung und Kühlung) folgen mit 18%. Die Verbrauchskosten für den Strom folgen an dritter Stelle mit 13%. ([Ha10])

Die Betrachtungen von James Hamilton ergeben klare Hinweise auf Größenordnungen und Fertigungstiefen. Viele Cloud Provider bauen ihre Server selbst oder in Auftragsfertigung und kaufen nicht von der Stange. In den Servern von Facebook beispielsweise findet man keine Videokarten, weil Herstellkosten und Verbrauchskosten verschwendet werden, wenn gar kein Monitor angeschlossen wird. Auf die Platinen der Server von Google werden einzelne Akkus gesetzt, die eine große zentrale und teure unterbrechungsfreie Stromversorgung ersetzen. PUE steht für Power Usage Efficiency und Werte oberhalb von 1,0 stehen für den Energieanteil, der für die Verteilung bzw. die Kühlung anfällt bzw. wegfällt; die besten Cloud Rechenzentren liefern Werte unter 1,1.

2.2 Verbundvorteile in den Anwendungen – Public SaaS

Salesforce legt im 10-K Formular der SEC Filings die Verhältnisse von Kosten und Umsatz für sein Customer Relationship Management (CRM) offen. Bei Kosten von ca. 127 Mio. \$ für die Serviceerbringung wurde ein Umsatz von ca. 1 Mrd. \$ erzielt [Sf09].

Salesforce bediente seine 55.000 Unternehmenskunden mit 1.500.000 Benutzern auf nur 1.000 Servern [TC09]. Für diese enorme Packungsdichte auf den Servern ist die Mandantenfähigkeit der Architektur maßgeblich, die Salesforce in besonderer Weise perfektioniert und für die entwickelten Technologien auch Patente angemeldet hat.

Salesforce betreibt keine eigenen Rechenzentren und setzt keine Virtualisierung ein. Stattdessen verbaut Salesforce in Colocation-Facilities von Equinix sogenannte Pods. Ein Pod bündelt jeweils eine Java Enterprise Plattform; in jedem Pod findet sich ein Oracle Real Application Cluster mit 8 Knoten und etwa 40 Resin Application Server stellen die Funktionen der Anwendung zur Verfügung. Jedem Pod werden solange weitere Unternehmenskunden hinzugefügt, wie es die Erfahrungswerte von Salesforce zulassen. Schrittweise werden dann neue Pods für neue Unternehmenskunden aufgebaut.

Anbieter von SaaS, die nur auf Größenvorteile der Infrastruktur bauen, werden im Markt gegenüber Anbietern, die Verbundvorteile bei Anwendungen ausnutzen, wirtschaftlich nicht bestehen können.

2.3 Netzwerkeffekte mit zweiseitigen Märkten

Zweiseitige Märkte finden auf Plattformen statt, die zwei unterscheidbare Nutzergruppen zusammenführen. Beispiele für solche zweiseitigen Märkte gibt es viele: Einkaufszentren, Kreditkarten, Betriebssysteme, Facebook, Google und andere.

Mit der Nutzung der Plattform durch die beiden Nutzergruppen entstehen zweiseitige, indirekte Netzwerkeffekte. Je mehr Teilnehmer der einen Gruppe die Plattform verwenden, desto attraktiver wird sie für die Nutzer der anderen Gruppe und umgekehrt.

Für den Nutzer eines Betriebssystems (Nutzergruppe A) entsteht ein Zusatznutzen, wenn möglichst viele Entwickler (Nutzergruppe B) dafür entwickeln; denn Programmauswahl und Anzahl der Nutzer steigen und damit auch der Anreiz für die Entwickler.

Werbefinanzierte Dienste bilden immer einen zweiseitigen Markt. Auch wenn Internetdienste ein API einführen, entsteht ein zweiseitiger Markt: Entwickler und Nutzer profitieren vom entstehenden reichhaltigen Ökosystem.

Letztlich sind alle IaaS/PaaS/SaaS-Angebote zweiseitige Märkte. Bei SaaS mag das auf den ersten Blick weniger offensichtlich sein, weil der Anbieter sein Angebot unmittelbar für die Nutzer entwickelt und bereitstellt – das spricht zunächst für einen einseitigen Markt. Weil jedoch die Anbieter meist nicht genügend Breite besitzen (wollen), eine umfassende Abdeckung für Standardsoftware (CRM, SCM, ERP, HRM, ...) selbst bereitzustellen, bieten sie oft eine komplementäre Plattform (PaaS) an, über die Partner das Portfolio des Anbieter erweitern – mit zusätzlichen Lösungen und Einbindungen.

3 Technische Rahmenbedingungen für Availability und Scalability

Im Jahr 2000 stellte Eric Brewer in einem Vortrag eine Vermutung bezüglich Consistency, Availability und Partition Tolerance in verteilten Systemen vor: "You can have at most two of these properties in any shared-data system." ([Br00]). Im Jahr 2002 haben Seth Gilbert und Nancy Lynch vom MIT die Vermutung formal bewiesen und sie damit zum CAP-Theorem (Anfangsbuchstaben der Merkmale) erhoben. ([GL02])

Weil Partition Tolerance im Internetumfeld mit instabilen Verbindungen immer gefordert ist, bleibt eigentlich nur die Abwägung zwischen Availability und Consistency.

Mit lahmenden Webseiten verlieren Unternehmen im Internet wichtige Kunden und Umsätze. In einem Vortrag von Martin Kliehm findet man konkrete Zahlen ([K110]):

- Amazon: 100 ms Verzögerung => -1 % Umsatz
- Bing: 2000 ms Verzögerung => -4,3 % Umsatz
- Google: 400 ms Verzögerung => -0,59 % Suchen pro Anwender
- Yahoo!: 400 ms Verzögerung => -5-9 % Datenverkehr

Seien es Warenumsätze oder Werbeeinnahmen, bei Unternehmen dieser Größenordnung geht es bei diesen Rückgängen um viel Geld und erhebliche Verluste. Da muss die Infrastruktur rund um die Uhr laufen und durch hohe Verfügbarkeit und niedrige Verzögerungen glänzen, im direkten Interaktionsverhalten genauso wie bei den Services im Hintergrund. Die virtuellen Warenkörbe der Kunden eines Internethändlers müssen jederzeit gelesen und geschrieben werden können, beim Wegfall eines einzelnen Servers wie nach dem Wegbrechen eines ganzen Rechenzentrums.

Und deswegen wägt die Internet-IT eher in Richtung Availability ab, während sich die Unternehmens-IT eher in Richtung Consistency ausrichtet – und diese eigentlich wirtschaftlichen Überlegungen ziehen weitreichende technische Folgen nach sich.

Denn in der Folge hat die Internet-IT eigene Dateisysteme und Datenbanken entwickelt. Mit Amazon Dynamo legen Kunden jederzeit Waren im Einkaufskorb ab. Das verteilte Google File System ist für die Internetsuche mit hohen Datendurchsätzen optimiert. Und bei Facebook unterstützt Cassandra in einem Cluster mit mehr als 600 Prozessorkernen und einem Plattenumfang von mehr als 120 TB die Inbox Search.

Diese Dateisystemen und Datenbanken beruhen auf Entwürfen, die mit herkömmlichen Konzepten von Relationen und Transaktionen brechen, und Neuland betreten ([Vo08]).

5 Private Clouds vs. Public Clouds

Der Übertragbarkeit der Konzepte von Amazon und Salesforce auf Private Clouds im Unternehmen ist begrenzt; die Grenzziehung verläuft für IaaS und SaaS unterschiedlich.

5.1 Private IaaS Cloud

In der Rechnung für IaaS wurden die jährlichen Gesamtkosten eines Rechenzentrums mit 46.000 Servern auf ca. 42,3 Mio \$ geschätzt. Für die oben aufgezählten Optimierungen ergeben sich folgende wesentlichen Unterschiede bei Private Clouds:

- Private Clouds sind deutlich kleiner.
- Private Clouds sind personalintensiver.

Die meisten auch größeren Unternehmen dürften bei ihren Rechenzentren nicht über 5.000-10.000 Servern hinauskommen und damit bei einem Fünftel bzw. einem Zehntel der Serverzahlen liegen. Damit greifen die beispielhaft angesprochenen Optimierungen deutlich weniger und ihre Amortisation liegt deutlich niedriger.

In seiner Schätzung blendet James Hamilton die Personalkosten aus, weil sie mit einer sehr weitgehenden Standardisierung und Automatisierung nur 3% zu den Gesamtkosten beitragen – bei den besten Anbietern. In klassischen Unternehmensrechenzentren liegt der Anteil der Personalkosten bei ca. 30-40% und dürfte auch in der Anfangsphase von Private Clouds nicht schnell und nicht wesentlich sinken.

Jenseits dieser beiden Erwägungen zur Wirtschaftlichkeit von Private Clouds ist die grundsätzlich unterschiedliche Architektur von Servern, Netzwerk und Storage im Rechenzentrum von Amazon und im klassischen Rechenzentrum eines Unternehmens das wesentliche Merkmal zur Unterscheidung und Gestaltung einer Private IaaS Cloud.

In den weltweit verteilten Rechenzentren von Google stehen nach einschlägigen Schätzungen mittlerweile mehr als eine Million Server; ebenso gibt es Schätzungen, dass die drei größten Rechenzentrumsbetreiber zusammen auf mehr als 2 Millionen Server kommen. Bei einer derart hohen Anzahl von Server sind tägliche Ausfälle nicht mehr nur bedauerliche Zufälle, sondern Regelgeschäft. Auch bei Speichern und bei Netzen gehen die Ingenieure solcher Großrechenzentren von alltäglichen Ausfällen aus.

Deswegen wandert bei diesen Infrastrukturen die Redundanz von der Hardware in die Software und die Anwendungen. Einzelne Server bekommen dann keine redundanten Netzwerkkarten oder Stromteile mehr und der Ausfall eines Geräts führt in der Folge möglicherweise zu weiteren Ausfällen einer ganzen Anzahl von Servern.

Die Infrastruktur wird daher in sogenannte Availability Zones unterteilt, die die Reichweite/Tragweite eines einzelnen Ausfalls isolieren. Und ähnlich wie RAID seine Speicherblöcke über mehrere Festplatten verteilt, wird eine Anwendung durch Application Striping über mehrere Availability Zones gespreizt und erhält eine gute Widerstandsfähigkeit gegen den Ausfall einer ganzen Availability Zone (Resiliency).

Auch den Ausfall einer gesamten Availability Zone überlebt die Anwendung deswegen, weil sie vorher mit weiteren Anwendungsteilen in anderen Availability Zones vorgesorgt hat. Es ist aber gar nicht so einfach, alle Anwendungsteile mehrfach vorzuhalten und über die Availability Zones zu spreizen. Für Loadbalancer und Webserver können Availability Zones und Application Striping einfach umgesetzt werden. Dazu müssen Sticky Sessions vom Loadbalancer verbannt werden und die Komponenten auf dem Appserver zustandslos programmiert werden (REST – Representational State Transfer).

Für Datenbanken gibt es jedoch theoretisch wie praktisch keine einfachen Lösungen. Die theoretischen Hürden ergeben sich aus dem CAP-Theorem, weil es ja beim Database Striping um die Balance aus Consistency, Availability und Partition Tolerance geht. Praktisch hat Database Striping bei Shared-Disk Architekturen keine Chance, weil die geteilte Festplatte ja in einer einzigen Availability Zone liegen würde/müsste.

Die Anwendungsteile müssen für Availability Zones und Application Striping genügend klein geschnitten werden können, denn sonst gehen mit den neuen Prinzipien erhebliche Verteuerungen einher. Gerade hier kann Java Enterprise Edition nicht punkten, denn die einschlägigen Application Server sind einfach groß und ressourcenhungrig. Ein Ausfall einer Availability Zone kann durch Anwendungsteile in anderen Availability Zones ausgeglichen werden; eine Lastspitze kann durch Anwendungsteile in einer Burst Zone ausgeglichen werden. Mit kleinen Anwendungsteilen geht beides: Heilen und Dehnen.

Für dieses Bauprinzip mit Availability Zones und Application Striping hat sich die Bezeichnung Commodity Cloud im Gegensatz zur Enterprise Cloud herausgebildet; die Enterprise Cloud verbaut Infrastrukturen und Architekturen mit klassischer Redundanz.

5.2 Private SaaS Cloud

Bei der Mandantenfähigkeit von Salesforce CRM stellt sich natürlich die Frage nach dem Nutzen für die interne Verwendung im Unternehmen. Die Notwendigkeit unterschiedlicher und mehrerer Mandanten verweist eigentlich eher auf eine zu geringe Standardisierung. Jenseits der Geschäftsprozessunterstützung mit Enterprise Applications (CRM, SCM, ERP usw.) gibt es zwei weitere Anwendungsklassen für Public/Private SaaS: Social Applications und Productivity Applications. Unter Social Applications fallen Funktionen wie Wikis, Blogs und Foren; insgesamt ist der Markt hierfür erst im Entstehen. Die prominentesten Funktionen der Productivity Applications sind Calendar, Contacts und Emails ebenso wie Textverarbeitung und Tabellenkalkulation und Präsentationen. Hier dürften sich in den Private Clouds üblicherweise die gleichen Hersteller finden (IBM, Microsoft) wie auch bei der klassischen Bereitstellung.

6 Zusammenfassung und Ausblick

Was Utility Computing mit Virtualisierung, Standardisierung und Automatisierung begonnen hat, führt Cloud Computing mit Simplification, Commoditization und Federation weiter. Im Zuge des Cloud Computing verlieren die Utilities (Server, Storage, Network) jedoch mit dem Übergang zu Commodities ein Stückweit ihre Einfachheit; die Ressourcen müssen nunmehr im architektonischen Verbund sorgsam verstanden (Commoditization) und gezielt zusammengesetzt (Federation) werden. Neues Denken im Gesamtzusammenhang von Commodities (IaaS), Elasticity (PaaS) und Tenants (SaaS) birgt auch neue Komplexität, die mit den Ansprüchen von On-Demand, Self-Service und Network-Access jedoch beherrschbar bleibt (Simplification).

Literaturverzeichnis

- [Br00] E. A. Brewer: Towards Robust Distributed Systems. PODC Keynote. Juli 2000. <http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf>
- [GL02] S. Gilbert, N. Lynch: Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. <http://lpd.epfl.ch/sgilbert/pubs/BrewersConjecture-SigAct.pdf>
- [Ha10] J. Hamilton: Overall Data Center Costs. <http://perspectives.mvdirona.com/2010/09/18/OverallDataCenterCosts.aspx>
- [Kl10] M. Kliehm: Web Performance Optimierung. Vortrag am 17.5.2010 beim Webmontag in Frankfurt, <http://www.slideshare.net/kliehm/performancewmfra>
- [MG09] P. Mell, T. Grance: The NIST Definition of Cloud Computing. Version 15, Juli 2009. <http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc>
- [Sf09] Salesforce.com: Form 10-K für das Fiskaljahrende am 31. Januar 2009. <http://www.sec.gov/Archives/edgar/data/1108524/000119312509048665/d10k.htm>
- [TC09] TechCrunch: The Efficient Cloud: All Of Salesforce Runs On Only 1,000 Servers. Blog Eintrag. März 2009. <http://techcrunch.com/2009/03/23/the-efficient-cloud-all-of-salesforce-runs-on-only-1000-servers/>
- [Vo08] W. Vogels: Eventually Consistent – Revisited. Dezember 2008. http://www.allthingsdistributed.com/2008/12/eventually_consistent.html