

Seminararbeit

## **Relationale Cloud-DB**

**Seminar Cloud Data Management (WS09/10)**

Autor: B. Sc. Sebastian Wienecke (Mat.-Nr. 1741663)  
Studiengang: Master of Science Informatik (3. Semester)

Betreuer: Dipl.-Inf. Lars Kolb

Eingereicht am: 30.03.2010

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Motivation</b> .....	<b>3</b>
<b>2</b>	<b>Grundbegriffe</b> .....	<b>4</b>
2.1	Einführung in Cloud Computing .....	5
2.2	Möglichkeiten der Datenspeicherung in Cloud Computing Umgebungen	7
2.3	Relationale Datenbanksysteme in der Cloud .....	9
2.4	Kurzvergleich MapReduce-Ansatz und relationale Datenbanksysteme	10
2.5	Hybridsystem: Verbindung MapReduce und relationales DBMS .....	11
<b>3</b>	<b>Praxisbeispiele</b> .....	<b>12</b>
3.1	Microsoft SQL Azure .....	13
3.1.1	Windows Azure .....	13
3.1.2	Windows Azure AppFabric.....	14
3.1.3	SQL Azure.....	14
3.1.4	Fazit .....	16
3.2	Amazon RDS (Relational Database Service).....	16
3.3	Amazon EC2 Relational Databases AMIs.....	18
<b>4</b>	<b>Zusammenfassung</b> .....	<b>19</b>
	<b>Literatur- und Quellenverzeichnis</b> .....	<b>20</b>

## 1 Einleitung und Motivation

Das Internet ist in den letzten Jahren zum Inbegriff unserer Zeit geworden, das 21. Jahrhundert wird bisweilen schon als „Internetzeitalter“ bezeichnet [Goo10a]. Es ist mittlerweile für viele Menschen, vor allem in den Industrienationen, genauso selbstverständlich wie Radio, Fernsehen oder das Automobil. In diesem hart umkämpften Telekommunikationsmarkt versuchen sich die Provider durch immer neue Technologien und sinkende Preise gegenseitig zu überbieten. Die dadurch in den letzten 7-10 Jahren massiv gesteigerte Bandbreite und Marktdurchdringung von Breitbandanschlüssen [OEC09] macht es den Konsumenten mittlerweile problemlos möglich, in Webseiten oder Webapplikationen genauso schnell zu navigieren und zu arbeiten, als würden diese lokal auf dem eigenen Rechner vorhanden sein.

Das Besuchervolumen auf einzelnen Internetseiten verteilt sich selten gleichmäßig über den gesamten Tag, sondern unterliegt häufig erheblichen Schwankungen. Auslöser dafür sind beispielsweise Berichterstattungen in frequentierten Medien oder die ungleichmäßige Bevölkerungsverteilung auf der Erde in den verschiedenen Zeitzonen. Diese Belastungsspitzen zwingen die Seitenbetreiber dazu, große Mengen an Rechenleistung ständig auf Abruf vorzuhalten.

An diesen beiden Aspekten setzt die Technik des Cloud Computings an. Große Unternehmen wie Google, Amazon oder Microsoft bieten dabei Kunden die Möglichkeit an, Rechen- oder Speicherleistung gegen entsprechende Gebühren für eigene Zwecke zu mieten. Während anfänglich nur ungenutzte Rechenkapazität vermietet wurde, entstand aus der Idee rasch ein rentables Geschäftsmodell und die Anbieter investieren mittlerweile gezielt in umfangreiche Rechenzentren zur alleinigen Vermietung. Dies wird als „Infrastructure as a Service“, oder kurz als „IaaS“ bezeichnet. Für den Benutzer erscheint die dadurch geschaffene Plattform als „Wolke aus vernetzten Computern“ – eine „Cloud“. Eine Abrechnung erfolgt dann je nach Anbieter oder Technik nach Rechenzyklen, Speicherplatz oder Übertragungsvolumen. Effizient nutzbar wird Cloud Computing aber erst durch die eingangs angesprochene Steigerung der Übertragungsbandbreite.

Gerade für kleine oder mittelständische Unternehmen kann dieser Dienst interessant sein, durch die Verlagerung der Daten und Software „in die Cloud“ muss selbst keine, oder nur eine minimale, Serverinfrastruktur bereitgestellt werden. Auf diese Weise kann ein Unternehmen 6 oder 7-stellige Beträge initial einsparen. Bei wachsender Unternehmensgröße oder gestiegenem Rechenaufwand können schnell und problemlos zusätzliche Kapazitäten in der Cloud dazu gebucht werden. Allerdings birgt dieser Ansatz neben datenschutzrechtlichen Bedenken auch das Risiko, bei einem Ausfall des Internets oder der Cloud komplett arbeitsunfähig zu sein. Außerdem entsteht eine Abhängigkeit zum Betreiber, ein Wechsel zu einem anderen ist unter Umständen nur schwer oder überhaupt nicht möglich.

Die Entwicklungen und Tendenzen in diesem Gebiet sind noch lange nicht beendet. Nachdem zunächst nur die Möglichkeit bestand, Dateien in speziellen Dateisystemen der Cloud zu speichern, konnten alsbald auch einfache Programme direkt dort ausgeführt werden. Es entstanden neue Datenbanksysteme, welche nur die Möglichkeit bieten, Key-Value-Paare zu speichern. Diese Systeme sind aber gezielt auf eine parallele Datenverarbeitung von hunderten Rechnern optimiert, unterstützten allerdings keine Relationen zwischen den einzelnen Datensätzen. Gerade diese Einschränkung ist aber hinderlich für viele Anwendungsszenarien. In dieser Arbeit soll daher untersucht werden, inwieweit es derzeit möglich ist, relationale Datenbanksysteme in Cloud Computing Umgebungen zu betreiben, wo die Vor- und Nachteile liegen und inwieweit diese auf einen Parallelbetrieb optimiert sind. Im zweiten Teil der Arbeit werden dann einige Anbieter betrachtet und diese miteinander verglichen.

## **2 Grundbegriffe**

In diesem Kapitel werden die Grundlagen für die spätere Betrachtung der Praxisbeispiele gelegt. Dazu ist es nötig, zunächst relevante Begriffe zu betrachten und in einen Kontext einzuordnen.

In dieser Arbeit wird weitestgehend auf eine vollständige Betrachtung von Cloud Computing verzichtet, da dies in anderen Arbeiten des Seminars bereits intensiv geschieht. Einige knappe Ausführungen sind aber zum weiteren Verständnis der Arbeit notwendig. Anschließend wird der Aspekt der relationalen Datenbanksysteme in der Cloud beleuchtet.

## 2.1 Einführung in Cloud Computing

Etwa im dritten Quartal 2007 begann ein Begriff durch das Internet und die Fachpresse zu geistern – Cloud Computing [Goo10b]. Mittlerweise vergeht kaum ein Tag, an dem nicht in einschlägigen IT-News-Portalen (etwa heise.de oder golem.de) eine neue Schlagzeile zu diesem Thema veröffentlicht wird. Was also ist „diese Wolke“?

Unter dem Begriff verbirgt sich die Umsetzung eines langjährigen Traumes für Programmierer und Anwender: Rechenleistung, wann und so viel man möchte für möglichst wenig Geld [BFG08]. Ein Anbieter stellt eine scheinbar unendliche Menge an Servern bereit, welche ein Kunde flexibel und bedarfsgerecht mieten kann. Das steht im Gegensatz zu den bisher üblichen Server- oder Hostingangeboten, bei denen physische oder virtuelle klar definierte Maschinen vermietet werden und die die jeweiligen Mieter größtenteils selbst administrieren müssen. Wird mehr Rechenleistung oder Speicherplatz benötigt, muss meist umständlich der Vertrag geändert werden. Bei den dem Autor bekannten Cloud-Computing-Anbietern hingegen kann der Kunde über ein Web-Interface per Klick zusätzliche Leistung hinzubuchen, die Abrechnung erfolgt einzig nach in Anspruch genommenen Kapazitäten.



*Abbildung 1: Ein Container mit ca. 2500 Servern für Cloud Computing am Beispiel von Microsoft Windows Azure [Lyl09]*

Wie bereits erwähnt stellen die Anbieter eine große Anzahl von Servern bereit um diesen Dienst zu ermöglichen. Bis zu 200.000 werden in großen Rechenzentren miteinander vernetzt. Um möglichst platz- und kosteneffizient zu operieren sind diese als Blades in ein-

heitlichen Containermodulen untergebracht. Ein solcher Container ist in der Abbildung 1 zu sehen.

Für den Betreiber des Rechenzentrums senkt diese starke Konzentration der Computer an einem Ort die durchschnittlichen Unterhaltungskosten. Diese Preisvorteile können direkt an den Endkunden weitergegeben werden.

Technisch betrachtet ist die Cloud alleine nicht wirklich neu. Jeder Nutzer erhält einen Zugang zu einem oder mehreren virtuellen Computern, welche in der Serverfarm betrieben werden. Je nach Dienst oder Anbieter stehen dem Kunden verschiedene Leistungen zur Verfügung, reinen Speicherplatz (Amazon S3), Datenbanken (Amazon SimpleDB, Google BigTable), Laufzeitumgebungen für eigene Anwendungen (Google AppEngine) oder komplette Betriebssysteme (Microsoft Azure). Je nach Ausprägung wird dies als „Software-as-a-Service“, „Platform-as-a-Service“ oder „Infrastructure-as-a-Service“ bezeichnet. Auch das gab es bereits vorher. Neu ist beim Cloud Computing aber die zugrunde liegende Software, welche vollständig auf diesen Einsatzzweck optimiert wurde. Virtuelle Maschinen werden nicht mehr ausschließlich von einem oder mehreren statischen Servern betrieben, sondern können sich in der Cloud mehr oder weniger frei bewegen. Spezielle Lastverteiler verschieben Teile des Betriebssystems oder einzelner Anwendungen auf andere momentan weniger belastete Server, um die Rechner gleichmäßig auszulasten, Antwortzeiten zu minimieren und eine nahezu lineare Skalierbarkeit zu gewährleisten [BFG08].

Ebenfalls neu sind spezielle Cloud-Dateisysteme. Diese bieten schier unbegrenzten Speicherplatz bis in den Petabyte-Bereich hinein, meist mit mehreren Replikaten pro Speicherblock auf mehreren Servern oder physikalischen Platten. Dies geschieht für Anwendungen oder Benutzer vollständig transparent, der gesamte Speicher erscheint als „ein großes Ganzes“.

Cloud Computing kann wegen der oben und in der Einführung genannten Vorteile sowohl für kleinere Unternehmen als auch für große Konzerne interessant sein. Einerseits wird der Unterhalt einer eigenen Server-Infrastruktur minimiert und technische Administratoren können eingespart werden, andererseits erhält man vollständige Datensicherheit und eine Plattform, welche mit den eigenen Anforderungen problemlos mitwachsen kann

(„pay-as-you-grow“ – Kosten richten sich proportional zur genutzten Leistungsmenge). Erkauft wird dieser Service aber mit einer Abhängigkeit zu einem Anbieter, welchem man unter Umständen sehr viele unternehmenskritische Daten anvertraut.

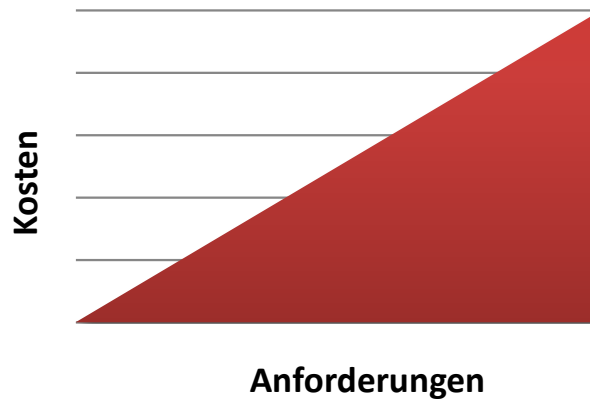


Abbildung 2: Linearer Kostenanstieg basierend auf genutzten Leistungen

## 2.2 Möglichkeiten der Datenspeicherung in Cloud Computing Umgebungen

Prinzipiell gibt es in der Cloud die gleichen Möglichkeiten der Datenspeicherung wie in anderen On- oder Offlineumgebungen: in Dateien oder in (meist relationalen) Datenbanken. Der Unterschied besteht darin, dass die zu speichernden Datenmengen meist um ein vielfaches größer sind. Dies stellt natürlich ganz besondere Herausforderungen an die verwendete Soft- und Hardware und die zugrundeliegenden Algorithmen. Als Beispiel sei hier das Social Network Facebook.com erwähnt, welches Ende 2009 etwa 2,5 PB an Daten gespeichert hatte, mit einem täglichen Zuwachs von über 15 TB [ABA09 S. 2; Whi09 S. 415]. Die Daten werden ausschließlich von den derzeit ungefähr 400 Mio. Benutzern [Rot10] generiert und müssen jederzeit bei minimalen Antwortzeiten abrufbar sein.

Wie gerade erwähnt ist die einfachste Möglichkeit der Datenspeicherung die Dateiablage. Ein großer Anbieter dieses Dienstes ist Amazon mit seinem S3 („Simple Storage Service“) [Ama10a]. Die Anwendungsgebiete sind in erster Linie reine Datenbackups oder das Speichern von großen unstrukturierten oder semistrukturierten Textdateien (z. B. Protokoll-dateien). In der Cloud wird die einfache Dateiablage, wie im vorhergehenden Abschnitt angedeutet, durch spezielle Mechanismen unterstützt. Je nach Anbieter oder gebuchter Leistungsklasse werden die Daten mehrfach auf unterschiedlichen Servern redundant gespeichert. Die Auswertung der so gespeicherten Daten ist relativ trivial: um beispiels-

weise bestimmte Protokolleinträge zu finden muss die Anwendung die jeweilige Datei zeilenweise durchlaufen, gefundene Datensätze extrahieren und anschließend interpretieren. Dieser Prozess kann durch die redundante Ablage auf verschiedenen Servern parallel stattfinden um die Antwort- oder Suchzeiten zu verkürzen. Im weiteren Teil dieser Arbeit soll die Dateiablage nicht weiter betrachtet werden.

Neben unstrukturierten Text- oder Binärdaten gibt es natürlich noch die relational abbildbaren strukturierten Daten. Das klassische Beispiel ist eine Kunden-Verträge-Datenbank, wie sie wahrscheinlich in jedem Unternehmen vorliegt. Zur Speicherung dieser Daten werden meist relationale Datenbanksysteme (MySQL, Oracle, DB2, MS SQL, usw.) genutzt. Diese Systeme wurden jahrzehntelang weiterentwickelt und bieten höchst effiziente Algorithmen zur Filterung, Sortierung und Vereinigung gespeicherter Datensätze. Da diese Form der Datenspeicherung das Hauptthema der vorliegenden Arbeit darstellt, wird darauf im nachfolgenden Abschnitt 2.3 detaillierter eingegangen.

In Cloud-Umgebungen gibt es neben den beiden bisher genannten Möglichkeiten zur Datenspeicherung noch eine dritte – die Key-Value-Stores. Hierbei handelt es sich um eine einfachere Form eines Datenbanksystems, welches nur schwach normalisiert ist und keine Transaktionen unterstützt. Datensätze werden immer in der Form „Index und zugehörige Werte“ abgelegt. Diese Designentscheidungen werden bewusst zur Geschwindigkeitsoptimierung beim Speichern und Lesen getroffen. Es finden selten Löschungen statt, neue Daten werden einfach am Ende der Tabellen angehängt und durch eine höhere Versionsnummer als aktuell markiert. Durch die Kopplung der Key-Value-Stores an die Cloud-Dateisysteme können Synergien genutzt werden. Eine Tabelle wird beispielsweise physikalisch in mehreren Dateiblöcken gespeichert, diese werden mehrfach repliziert und auf unterschiedlichen Servern gespeichert. Bei einem Lesezugriff kann die Anfrage somit parallel von mehreren Rechnern bedient und somit die Antwortzeit verkürzt werden. Dieser Parallelismus ist ein weiterer Grund, wieso auf eine Normalisierung verzichtet wird: Jeder Server kann in vielen Fällen autonom mit seinen Daten arbeiten, ohne auf andere warten zu müssen. Durch die nahezu vollständige Vermeidung von Änderungen und Löschungen von Datensätzen werden Sperr- und Commitvorgänge obsolet, was sich wiederum positiv auf die Lesegeschwindigkeit auswirkt.



Im weiteren Verlauf dieser Arbeit werden nur noch die relationalen Datenbanksysteme betrachtet, da diese das Hauptthema dieser Arbeit darstellen.

### **2.3 Relationale Datenbanksysteme in der Cloud**

Wie bereits im vorhergehenden Abschnitt 2.2 beschrieben ist ein relationales Datenbanksystem eine Möglichkeit der Datenspeicherung in Cloud-Umgebungen. Der Einsatz eines solchen Systems bietet für Entwickler viele Vorteile. Auf der einen Seite stehen natürlich die zahlreichen Funktionen, die das System bieten kann: Joins, Indizes, Grouping, ACID-Eigenschaften, usw. Über die Relationen sind viele Modelle und mögliche Szenarien abbildbar. In jahrelanger Entwicklung haben die etablierten Hersteller, wie Oracle, IBM und Microsoft, effiziente Algorithmen geschaffen und die Sprache SQL größtenteils standardisiert. Auf eine vollständige Definition und Erklärung eines RDBMS soll in dieser Arbeit verzichtet werden.

Auf der anderen Seite steht die nahezu unendliche Rechenleistung, welche eine Cloud-Computing-Plattform bietet. Wie bereits erwähnt können hunderte oder tausende Rechner zusammengeschaltet werden, um ein Problem zu lösen oder eine Software zu betreiben.

Leider haben relationale Datenbanksysteme aber mit gewissen Altlasten zu kämpfen, welche einem Einsatz der Software in der Cloud im Weg stehen. Ein Betrieb auf mehreren Servern ist zwar von vielen Herstellern vorgesehen, allerdings über eigene, interne Kommunikationsprotokolle: Clustering. Dieser Verbund skaliert aber nur sehr begrenzt linear und ist keinesfalls auf einen Betrieb auf hunderten oder gar tausenden von Rechnern vorgesehen. ACID-Eigenschaften und Transaktionen führen zu kurzzeitigen Datensatzsperrungen, welche die Antwortzeiten in Summe vergrößern oder gar zu Dead-Locks führen. Eine verteilte Datenbank über viele Server benötigt auch umständliche Mehrphasen-Commit-Protokolle, um die Konsistenz über alle Knoten sicherzustellen. Auch das wirkt sich sehr negativ auf die Performanz des Gesamtsystems aus.

Diese Nachteile resultieren aus grundsätzlichen Designentscheidungen: Fehler treten selten auf und Datenbankserver waren immer teure High-End-Maschinen. In Cloud-Rechenzentren wird aber meist billige Standardhardware eingesetzt, hier gilt das Prinzip

„Masse statt Klasse“ – Hardwareausfälle werden bewusst in Kauf genommen, ein anderer beliebiger Rechner übernimmt nahtlos die Aufgaben des defekten Gerätes [ABA09 S. 3]. Damit können die althergebrachten DBMS nur schlecht umgehen, es kommt zu Transaktionsabbrüchen oder gar fehlenden Datenblöcken.

Weiterhin laufen die Datenbanksysteme meist auf normalen Server-Betriebssystemen (Windows, Unix, z/OS). In der Cloud werden diese notgedrungen in virtuellen Maschinen betrieben, welche die Komplexität und die Mechanismen der Cloud-Umgebung vor dem DBMS verbirgt. Es kann daher zu keinen Synergien kommen, wie beispielsweise automatische transparente Replikation von Datenblöcken auf mehrere Server und Lastverteilung und Ausfallsicherheit über und mit diesen Replikaten.

Bisher gibt es erst einen dem Autor bekannten Hersteller, welcher sein Produkt direkt für die Cloud optimiert hat: Microsoft betreibt seinen MS-SQL-Server als SQL Azure in der eigenen Cloud (siehe Abschnitt 3.1). Über technische Hintergründe hält sich Microsoft aber wie immer sehr bedeckt. Weitere Hersteller wie beispielsweise Oracle scheinen an der Optimierung ihrer Software zu arbeiten [Ora10; Gee10]: Unter dem Stichwort „Grid Computing“ entwickelt Oracle angepasste Versionen der Datenbanksysteme. Als Zielgruppe sieht Oracle derzeit aber eher große Unternehmen mit einer privaten, eigenen Cloud und scheinbar nicht die öffentlich verfügbaren Dienste wie beispielsweise Amazon Web Services (siehe auch Abschnitte 3.2 und 3.3). Daher hat der Autor diese Systeme in dieser Arbeit nicht näher betrachtet.

## **2.4 Kurzvergleich MapReduce-Ansatz und relationale Datenbanksysteme**

In Cloud-Umgebungen gibt noch eine weitere, bisher nicht genannte Methode, gespeicherte Daten zu extrahieren: MapReduce. Dies ist eine von Google erfundene und mittlerweile patentierte Methode [Hei10] unstrukturierte oder semistrukturierte Daten (meist Textdateien) hochgradig parallel zu durchsuchen und auszuwerten.

Die Idee dabei ist, zunächst auf sehr vielen Rechnern die Grundmenge an Daten parallel durchzugehen und relevante Daten herauszusuchen. Im zweiten Schritt erfolgt die Aggregation zur endgültigen Ergebnismenge. Auch das geschieht wieder parallel. MapReduce wurde gezielt für Cloud-Dateisysteme entworfen und nutzt die zugrunde liegenden Tech-

niken, wie beispielsweise die mehrfache Replikation von Datenblöcken: Da der Datensatz x-mal vorliegt, kann er auch x-mal gleichzeitig durchlaufen werden [ABA09 S. 2f].

Tabelle 1 zeigt die Unterschiede zwischen einem klassischen relationalen Datenbanksystem und dem MapReduce-Ansatz. Weiter ins Detail soll in dieser Arbeit nicht eingegangen werden.

Eigenschaft	relationales DBMS	MapReduce
verwaltete Datengröße	GB	PB
Zugriffsmethoden	interaktiv, Batch-Jobs	Batch-Jobs
Datenänderungen	oft lesen und schreiben	einmal schreiben, oft lesen
Struktur	statisches Schema	dynamisches Schema
Integrität	hoch	gering
Skalierung	nicht linear	linear
Datenspeicherung	im DBMS	in Textdateien, Key-Value-Stores

*Tabelle 1: Vergleich RDBMS mit MapReduce [Whi09 S. 5], übersetzt und ergänzt*

## 2.5 Hybridsystem: Verbindung MapReduce und relationales DBMS

An der Universität Yale wurde eine Verbindung zwischen dem MapReduce-Ansatz und einem relationalen Datenbanksystem hergestellt: HadoopDB. Die Technik basiert auf der freien MapReduce-Implementierung Hadoop und den freien Datenbanksystemen MySQL und PostgreSQL. Grundidee hierbei ist, Vorteile beider Systeme miteinander zu kombinieren. Im Regelfall sind die lang entwickelten Algorithmen in einem RDBMS schneller und zuverlässiger als selbst implementierte. Aus diesem Grund sollen diese für Suchanfragen genutzt werden. Wie bereits im Abschnitt 2.3 erwähnt verhindern aber globale Sperr- und Transaktionsprotokolle bei diesen Systemen eine lineare Skalierung. Dies soll über den MapReduce-Ansatz umgangen werden. Bedingung ist natürlich, dass in das Gesamtsystem selten geschrieben aber häufig aus ihm gelesen wird.

Auf jedem Server (Knoten) in der Cloud wird eine Instanz des Datenbanksystems installiert und über einen speziellen Connector mit einem Master-Knoten verbunden. Dieser weiß, welche Daten sich auf welchem Knoten befinden und empfängt vom Benutzer oder einer Anwendung gestellte SQL-Anweisungen. Über einen Interpreter wird dieser Befehl in eine Reihe von Map und Reduce-Kommandos übersetzt und an die einzelnen Knoten verteilt. Die Datenbankconnectoren übersetzen diese wiederum in SQL-Queries und fragen damit die lokale Datenbankinstanz ab. Anschließend vereint der Masterknoten die

einzelnen Ergebnismengen wieder [ABA09 S. 2ff]. Dieser Aufbau ist in Abbildung 3 dargestellt.

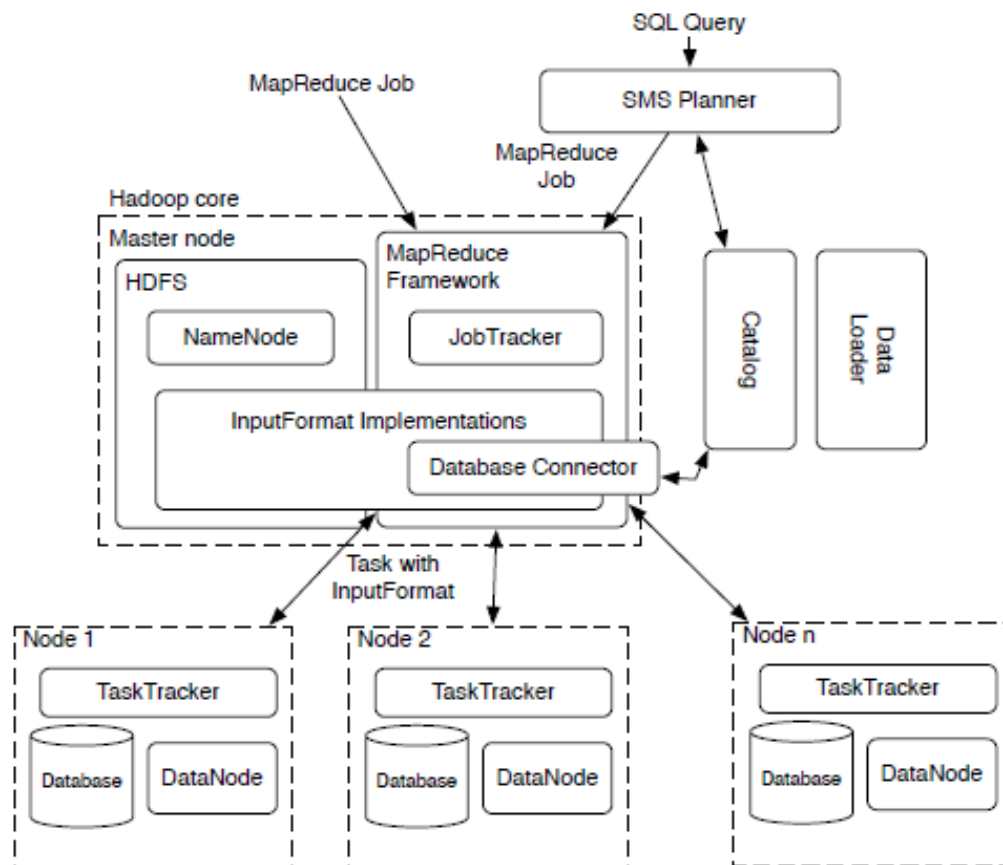


Abbildung 3: Aufbau von HadoopDB [ABA09 S. 4]

HadoopDB befindet sich derzeit noch in der Entwicklung und dem Autor ist kein Anbieter bekannt, der dies Produktiv einsetzt.

### 3 Praxisbeispiele

In diesem Kapitel werden einige Systeme und Techniken relationaler Cloud-DBS vorgestellt, welche bereits auf dem Markt verfügbar sind. Um nicht den Rahmen dieser Arbeit zu sprengen wurden bei der Auswahl der betrachteten Anbieter nur diejenigen berücksichtigt, welche sich bereits im Cloud-Computing-Umfeld etabliert haben und produktiv genutzt werden können.

### 3.1 Microsoft SQL Azure



Als erstes Praxisbeispiel wird der Cloud-Computing-Dienst Microsoft Windows Azure Plattform betrachtet. Unter diesem Namen stellt Microsoft ein mehrere Komponenten umfassendes Produkt zur Verfügung. Zur Verdeutlichung des Zusammenhangs und -spiels der diversen Einzelteile innerhalb der Plattform wird zunächst auf jede knapp eingegangen, bevor SQL Azure anschließend ausführlich beschrieben wird. Abbildung 4 zeigt einen Überblick über die Azure Plattform:



Abbildung 4: Komponenten der Windows Azure Plattform [MS10a]

#### 3.1.1 Windows Azure

Die Basis von Microsofts Azure Plattform stellen zahlreiche Rechenzentren mit jeweils mehreren 100.000 Blade-Servern weltweit dar (siehe auch Abbildung 1). Auf diesen Rechnern wird eine speziell angepasste Version von Windows – Windows Azure – betrieben, welche sich um die Verteilung von Speicherplatz und Rechenzeit kümmert. Eine zusätzliche Komponente, die „Fabric“, verbirgt die zugrundeliegende Anzahl von CPU-Kernen vor den oberen Schichten und ein Load-Balancer verteilt anfallende Arbeit an physikalische Prozessoren.

Der Kunde selbst kann über ein Web-Interface oder direkt aus dem Visual Studio (ab Version 2008) eigene Anwendungen in die Cloud laden. Möglich sind hierbei zahlreiche Sprachen, neben den .NET-Sprachen auch Microsoft-fremde wie Ruby, PHP oder Java. Hochge-

ladene Programme können ebenfalls über das Web-Interface gestartet oder wieder gestoppt werden. Auch mehrere Instanzen derselben Anwendung sind möglich. Jede Programminstanz wird in einer eigenen virtuellen Maschine mit Windows Server 2008 R2 in einer 64 Bit-Umgebung ausgeführt. Durch die Wahl eines nahezu „normalen“ Windows als Betriebssystem wird es den Entwicklern ermöglicht, beliebige Software nahezu ohne Anpassungen in der Microsoft-Cloud zu betreiben.

Neben der Anwendungsverwaltung können auch Parameter wie die Anzahl zugrundeliegender physikalischer Computer oder die Datenbankgröße per Mausclick mit sofortiger Wirkung geändert werden.

### *3.1.2 Windows Azure AppFabric*

Ein weiteres Element der Windows Azure Plattform ist die AppFabric. Sie untergliedert sich im Wesentlichen in zwei Teile: den Service Bus und das Access Control.

Über den Service Bus können sowohl Cloud-Anwendungen mit der Außenwelt interagieren als auch umgekehrt. Über eine eindeutige URI kann der Entwickler eine hierarchische Struktur aufbauen, welche von externen Programmen angesprochen werden kann. Über den Service Bus wird demzufolge eine eigene Form von WebServices bereitgestellt [Mäe09 S. 3f; Cha09 S. 7f].

Der zweite Teil, das Access Control, stellt Entwicklern einheitliche Methoden zur Benutzerauthentifikation und Zugriffsverwaltung bereit. Diese Funktionen können auch für den Service Bus genutzt werden [Mäe09 S. 3f; Cha09 S. 7f].

### *3.1.3 SQL Azure*

Das dritte wesentliche Element der Plattform ist SQL Azure. Ähnlich wie Windows Azure eine angepasste Version von Windows ist, ist SQL Azure eine Cloud-optimierte Version vom MS-SQL-Server. Dieser ist fest in das Gesamtkonzept integriert und bietet nahezu alle Möglichkeiten eines eigenständigen relationalen Datenbanksystems. So können beispielsweise Indizes, Fremdschlüssel, Sichten und Funktionen angelegt werden und Transaktionen werden unterstützt [MS09a S. 7].

Durch die Integration in den Cloud-Dienst kann SQL Azure Synergien nutzen: jede Datenbankinstanz wird dreifach repliziert und bietet so ein hohes Maß an Ausfallsicherheit und

Skalierbarkeit. Auf die Datenbank selbst kann sowohl von Cloud-Anwendungen als auch über die AppFabric von externen Programmen aus zugegriffen werden. Ein Unternehmen mit weltweiten Standorten kann somit eine einzige Datenbank nutzen, auf die von allen möglichen Orten aus zugegriffen werden kann – sofern dort Internet verfügbar ist. Über den globalen Load Balancer werden auch Zugriffe auf die Datenbank optimal auf die einzelnen Server verteilt, um niedrige Antwortzeiten zu ermöglichen. Abbildung 5 zeigt schematisch die Integration von SQL Azure in die Cloud, der Entwickler kann sich auf die einzelnen Abfragen konzentrieren, ohne sich um Konfiguration oder andere administrative Dinge der Datenbank kümmern zu müssen [MS09b; LMM09 S. 5].

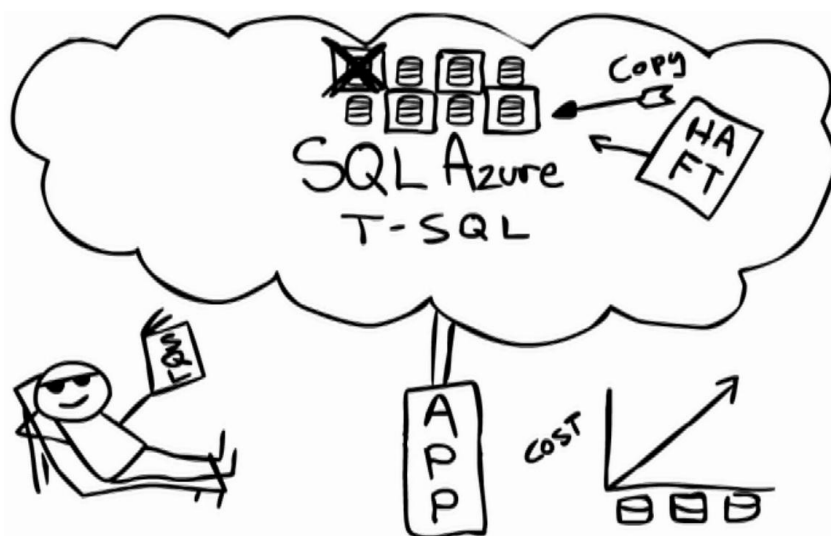


Abbildung 5: SQL Azure in der Microsoft Windows Azure Platform [MS09b]

Pro Datenbankinstanz sind bei Microsoft SQL Azure 10 GB möglich, Kunden können aber problemlos per Mausklick neue Instanzen hinzubuchen. Auf diese Weise kann ein Entwickler Datenbankoptimierungen wie Parallelität oder Partitionierung vornehmen. Dies wird allerdings derzeit nicht direkt unterstützt, sodass die unterschiedlichen Zugriffe auf die verschiedenen Datenbankinstanzen und das Zusammenführen der Daten in der Programmlogik selbst abgebildet werden muss [LMM09 S. 12]. Eine integrierte Partitionierung wie bei Key-Value-Stores ist allerdings derzeit in Planung [MS09c].

Eine Abrechnung erfolgt bei SQL Azure ebenso wie bei anderen Cloud-Diensten diverser Anbieter: einheitlich nach tatsächlich genutzter Leistung. Eine Übersicht bietet Tabelle 2.




 Windows Azure	Rechenzeit	\$ 0,12 pro h
	Speicherplatz	\$ 0,15 pro GB und Monat
	Speicherzugriff	\$ 0,01 pro 10T Transaktionen
 Microsoft SQL Azure	Web Edition (bis 1 GB)	\$ 9,99 pro Monat
	Business Edition (bis 10 GB)	\$ 99,99 pro Monat
 Windows Azure platform AppFabric	Access Control	\$ 1,99 pro 100T Transaktionen
	Service Bus	\$ 3,99 pro Verbindung
<b>Allgemein</b> (bei allen Diensten)	Datentransfer	\$ 0,10 in pro GB \$ 0,15 out pro GB

Tabelle 2: Kostenübersicht Windows Azure Platform [MS10b]

### 3.1.4 Fazit

Microsoft Windows Azure Platform mit integriertem SQL Azure bietet eine vollständige Lösung zum Betrieb von Anwendungen und relationalen Datenbanken in der Cloud. Durch die Bündelung zu einer kombinierten Lösung entfallen für den Entwickler Technologieübergänge und herstellereigene Eigenheiten. Nach Informationen des Autors ist Microsoft damit der derzeitige einzige Anbieter, welcher alles aus einer Hand bietet.

Anzumerken bleibt, dass einige Funktionen, die der Anwender möglicherweise von eigenständigen Serverprodukten her kennt, noch nicht in Azure verfügbar sind. An einigen arbeitet Microsoft angeblich bereits, über andere wird keine Aussage getroffen. Derzeit ist der Dienst aber auch noch nicht vollständig in den Produktivbetrieb übergegangen.

Speichergrößen von mehreren PetaBytes sind weder in Azure selbst noch in SQL Azure möglich. Das liegt vor allem an dem zu Grunde liegenden reinen RDBMS und dem Fehlen einer gesonderten Key-Value-Store-Implementierung, welche mit solchen Datenmengen umgehen könnte.

## 3.2 Amazon RDS (Relational Database Service)



Amazon, früher nur als versandkostenfreier Buchhändler bekannt, hat sein Geschäftsfeld mittlerweile stark erweitert. Über die Amazon Web Services, kurz AWS, werden zahlreiche Dienstleistungen angeboten, darunter auch typische Cloud-Computing-Dienste. So gibt es neben reinem Online-Speicherplatz, dem bereits erwähnten S3, auch eine Laufzeitumgebung für Anwendungen (Elastic Compute Cloud EC2), ein MapReduce-Framework (Elastic MapReduce) und verschiedene Datenbanksysteme.



Ein solches ist der in diesem Abschnitt vorgestellte Relational Database Service. Über diesen Dienst stellt Amazon ein vollständiges MySQL 5.1 Datenbanksystem zur Verfügung. Ähnlich wie bei SQL Azure kann der Benutzer über ein Webinterface oder über WebServices die Konfiguration steuern [Ama09]. Somit unterscheidet sich Amazons Produkt auf den ersten Blick nur unwesentlich von vielen anderen Angeboten im Internet. Bei nahezu jedem Hosting-Service erhält der Kunde Zugriff auf eine oder mehrere MySQL-Datenbanken.

Der Unterschied besteht im Wesentlichen in drei Punkten: Erstens kann bei Amazon auch von lokalen Clientprogrammen aus auf die Datenbank zugegriffen werden, während bei vielen Hosting-Services nur der Zugriff vom Hosting-Server aus möglich ist. Damit steht die Datenbank bei Amazon nicht nur als Datenquelle für eine Homepage zur Verfügung, sondern kann von jedem denkbaren Programm genutzt werden.

Der zweite Unterschied besteht darin, dass bei Amazon die Geschwindigkeit und die Datenbankgröße in mehreren Klassen anpassbar sind [Ama09; Ama10b]:

Name	RAM	CPU	Plattform	Kosten
Small DB Instance	1,7 GB	1 ECU	64 Bit	\$ 0,11 pro h
Large DB Instance	7,5 GB	4 ECU	64 Bit	\$ 0,44 pro h
Extra Large DB Instance	15 GB	8 ECU	64 Bit	\$ 0,88 pro h
Double Extra Large DB Instance	34 GB	13 ECU	64 Bit	\$ 1,55 pro h
Quadruple Extra Large DB Instance	68 GB	26 ECU	64 Bit	\$ 3,10 pro h

1 ECU = virtuelle CPU mit Leistung eines 1,0 – 1,2 GHz Opteron oder Xenon

Sonstige Kosten		
Speicherplatz	5 GB – 1 TB	\$ 0,10 pro GB und Monat
	I/O Zugriffe	\$ 0,10 pro 1 Mio. Zugriffe
Backup-Speicher	aktive DB	kostenlos
	inaktive DB	\$ 0,15 pro GB und Monat
Datentransfer	in	\$ 0,10 pro GB (kostenlos bis 30. Juni 2010)
	out	\$ 0,15 - \$ 0,08 pro GB (gestaffelt)
	zw. AWS	kostenlos (im selben Rechenzentrum)

*Tabelle 3: Übersicht über die verschiedenen RDS-Pakete und Kosten [Ama10b]*

Und der dritte Unterschied besteht darin, dass der RDS zumindest teilweise mit anderen Amazon Web Services gekoppelt ist. So wird beispielsweise regelmäßig ein Backup der Datenbank angelegt und im S3 gespeichert. Datentransfer zur oder von der Datenbank innerhalb eines Amazon Rechenzentrums (konfigurierbar), beispielsweise vom EC2-Dienst aus, ist kostenlos.

Ähnlich wie bei SQL Azure ist auch bei Amazon der Speicherplatz pro Datenbankinstanz begrenzt (siehe Tabelle 3). Ein Kunde kann aber auch hier mehrere Instanzen buchen, steht dann aber vor dem gleichen Problem: Datenpartitionierung oder parallele Zugriffe müssen in der eigenen Programmlogik abgebildet werden [Ama09]. Eine Datenbankinstanz ist als eigenständiges vollwertiges DBMS zu sehen, wie viele Datenbanken und Tabellen der Benutzer anlegt, bleibt ihm überlassen. Eine automatische Replikation der Instanz über mehrere Server ist im Moment nicht implementiert, nach Aussage Amazons aber in Planung [Ama09].

Abschließend lässt sich sagen, dass der RDS nicht das Niveau von SQL Azure erreicht. Er ist nur mäßig in das Gesamtkonzept integriert und bietet mit dem mittlerweile überholten MySQL 5.1 nicht den Funktionsstand wie Microsofts MS-SQL. Amazon kam hier scheinbar vorerst nur dem Wunsch der Kunden nach einem relationalen DBMS nach, ohne viel Aufwand zu betreiben. Wobei Microsoft als Softwareentwicklungsunternehmen hier sicherlich gewisse technische Vorteile hat.

### 3.3 Amazon EC2 Relational Databases AMIs



Neben RDS bietet Amazon noch eine zweite Möglichkeit eines relationalen DBMS an. Dieser Dienst heißt voll ausgeschrieben „Amazon Elastic Compute Cloud Relational Databases Amazon Machine Images“ und zeigt schon die zugrunde liegende Technik: Der Kunde kann eine klassische vorkonfigurierte virtuelle Maschine in der Amazon-Cloud betreiben [Ama10c]. Zur Auswahl stehen diverse Datenbanksysteme verschiedener Versionen und Hersteller, welche auf verschiedenen Betriebssystemen betrieben werden:

- IBM DB2
- Oracle DB 11g Release 1
- MySQL 5
- MS SQL Server
- PostgreSQL
- Sybase
- Vertica



Leider stellt Amazon nicht viele Informationen über diese vorkonfigurierten AMIs bereit. Somit ist davon auszugehen, dass ein Kunde diese virtuellen Computer per Klick starten

kann, er aber spezifische Einstellungen, wie Backups, Replikationen, Zugriffskontrollen, usw. selbstständig direkt auf der Maschine vornehmen muss. Auf eine Preisübersicht wird in dieser Arbeit verzichtet, da jedes Produkt eine individuelle Preisgestaltung hat.

## 4 Zusammenfassung

Leider konnte der Autor wenig konkrete Literatur über vorhanden relationale Datenbanksysteme in der Cloud finden. Das liegt zum einen daran, dass Cloud Computing ein sehr aktuelles und noch in der Entwicklung befindliches Thema ist und es noch nicht viele Studien darüber gibt. Zum anderen halten sich die Anbieter und Hersteller noch sehr über die zugrunde liegende Technik bedeckt und präsentieren im Großen und Ganzen nur Marketingtexte und -folien.

Außerdem gibt es für das Cloud Computing bisher zwei grundverschiedene Zielgruppen. Auf der einen Seite sind es große Netzwerke wie Facebook oder eBay mit ihren mehrere PetaBytes umfassenden DataWareHouses. Diese Datenmenge kann derzeit kein relationales DBS, innerhalb oder außerhalb einer Cloud, bewältigen. Auf der anderen Seite sind es Unternehmen, welche ihre Anwendungen und Datenbanken an einer Stelle zentral betreiben möchten, ohne sich um räumliche Verteilung, Synchronisationsprobleme und Backups kümmern zu müssen.

Nur die zweite Zielgruppe benötigt derzeit ein komplettes relationales DBS. Aufgrund von noch nicht abschließend geklärten Fragen, wie beispielsweise Datenschutz, Datensicherheit und garantierten Service-Level-Agreements, schrecken aber viele Unternehmen noch davor zurück, ihre gesamten Unternehmensdaten in die Cloud zu verlagern. Die großen Datenbankhersteller verspüren also scheinbar noch nicht den Druck, spezielle cloudoptimierte Produkte auf den Markt zu bringen. Bisher hat nur ein Hersteller – Microsoft – ein aufeinander abgestimmtes Gesamtpaket veröffentlicht.

Die Zukunft wird zeigen, wie sich der Markt entwickeln wird. Wahrscheinlich werden beide der in der Arbeit vorgestellten Ansätze koexistieren: Hybridsysteme wie HadoopDB für Abfragen auf riesige Datenmengen und cloudoptimierte relationale Datenbanksysteme für hochperformante relationale Datenbestände.

## Literatur- und Quellenverzeichnis

- [ABA09] **Abouzeid, Azza; Bajda-Pawlikowski, Kamil; Abadi, Daniel; et al.** *HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads*. [Online] 2009. [Zitat vom: 29. Dezember 2009.] <http://db.cs.yale.edu/hadoopdb/hadoopdb.pdf>.
- [AFG09] **Armbrust, Michael; Fox, Armando; Griffith, Rean; et al.** *Above the Clouds: A Berkeley View of Cloud Computing*. [Online] 10. Februar 2009. [Zitat vom: 24. März 2010.] <http://d1smfj0g31qzek.cloudfront.net/abovetheclouds.pdf>.
- [Ama09] **Amazon Web Services LLC.** *Amazon Relational Database Service FAQs*. [Online] 2009. [Zitat vom: 2. Januar 2010.] <http://aws.amazon.com/rds/faqs>.
- [Ama10a] **Amazon Web Services LLC.** *Amazon Simple Storage Service (Amazon S3)*. [Online] 2010. [Zitat vom: 23. März 2010.] <http://aws.amazon.com/s3/>.
- [Ama10b] **Amazon Web Services LLC.** *Amazon Relational Database Service (Amazon RDS)*. [Online] 2010. [Zitat vom: 28. März 2010.] <http://aws.amazon.com/rds/#pricing>.
- [Ama10c] **Amazon Web Services LLC.** *Amazon EC2 Relational Databases AMIs*. [Online] 2010. [Zitat vom: 24. Februar 2010.] [http://aws.amazon.com/running\\_databases/#relational](http://aws.amazon.com/running_databases/#relational).
- [BFG08] **Brantner, Matthias; Florescu, Daniel; Graf, David; et al.** Building a database on S3. *Proceedings of the 2008 ACM SIGMOD international Conference on Management of Data*. 2008, S. 251-264.

- [CDG06] **Chang, Fay; Dean, Jeffrey; Ghemawat, Sanjay; et al.** *Bigtable: A Distributed Storage System for Structured Data.* [Online] November 2006. [Zitat vom: 29. Dezember 2009.] <http://labs.google.com/papers/bigtable-osdi06.pdf>.
- [Cha09] **Chappel, David.** *Introducing the Windows Azure Platform.* [Online] Dezember 2009. [Zitat vom: 27. März 2010.] <http://go.microsoft.com/fwlink/?LinkId=158011>.
- [Gee10] **Geelan, Jeremy.** *An Exclusive Interview with Oracle.* [Online] 7. Januar 2010. [Zitat vom: 28. März 2010.] <http://cloudcomputing.systemcon.com/node/1153768>.
- [Goo10a] **Google Inc.** *Google Ergebnisliste für Internetzeitalter.* [Online] 21. März 2010. [Zitat vom: 21. März 2010.] <http://www.google.de/search?q=%22Internetzeitalter%22+%2221.+jahr+hundert%22>.
- [Goo10b] **Google Inc.** *Google trends - Cloud Computing.* [Online] 22. März 2010. [Zitat vom: 22. März 2010.] <http://trends.google.com/trends?q=cloud+computing>.
- [Hei10] **Heise Zeitschriften Verlag GmbH & Co. KG .** *Google lässt Map/Reduce patentieren.* [Online] 20. Januar 2010. [Zitat vom: 28. März 2010.] <http://www.heise.de/newsticker/meldung/Google-laesst-Map-Reduce-patentieren-908531.html>.
- [LMM09] **Lee, Jason; Malcolm, Graeme; Matthews, Alistair.** *Overview of Microsoft SQL Azure Database.* [Online] September 2009. [Zitat vom: 30. Dezember 2009.] Whitepaper. <http://go.microsoft.com/?linkid=9686976>.
- [Lyl09] **Lyle, Andrew.** *Inside a Windows Azure server container.* [Online] 17. November 2009. [Zitat vom: 22. Februar 2010.] <http://www.neowin.net/news/inside-windows-azure-server-container>.

- [Mäe09] **Mäenpää, Jouni.** *Cloud computing with the Azure platform.* [Online] 27. April 2009. [Zitat vom: 23. März 2010.]  
[http://www.cse.tkk.fi/en/publications/B/5/papers/Maenpaa\\_final.pdf](http://www.cse.tkk.fi/en/publications/B/5/papers/Maenpaa_final.pdf) .
- [MS09a] **Microsoft Corporation.** *Similarities and Differences of SQL Azure and SQL Server.* [Online] September 2009. [Zitat vom: 1. Januar 2010.]  
 Whitepaper. <http://go.microsoft.com/?linkid=9692818>.
- [MS09b] **Microsoft Corporation.** *Windows Azure - Azure Services.* [Online] 2009. [Zitat vom: 30. Dezember 2009.]  
<http://www.microsoft.com/germany/net/WindowsAzure/AzureServices.aspx>.
- [MS09c] **Microsoft Corporation.** *Windows Azure Platform - Frequently Asked Questions (FAQ).* [Online] 2009. [Zitat vom: 30. Dezember 2009.]  
<http://www.microsoft.com/windowsazure/faq/#sql>.
- [MS10a] **Microsoft Corporation.** *Windows Azure Platform.* [Online] [Zitat vom: 21. Februar 2010.] <http://www.microsoft.com/germany/net/WindowsAzure/images/windowsazureplatform.png>.
- [MS10b] **Microsoft Corporation.** *Windows Azure Pricing.* [Online] 2010. [Zitat vom: 25. März 2010.]  
<http://www.microsoft.com/windowsazure/pricing>.
- [OEC09] **OECD Broadband statistics.** *Households with broadband access (2004-2008).* [Online] Mai 2009. [Zitat vom: 22. März 2010.]  
<http://www.oecd.org/dataoecd/20/59/39574039.xls>.
- [Ora10] **Oracle.** *Oracle Cloud Computing.* [Online] 10. März 2010. [Zitat vom: 27. März 2010.]  
<http://www.oracle.com/us/technologies/cloud/index.htm>.
- [Rot10] **Roth, Philipp.** *Facebook Infografik und Statistiken.* [Online] 23. März 2010. [Zitat vom: 25. März 2010.]  
[http://facebookmarketing.de/zahlen\\_fakten/facebook-infografik-und-statistiken](http://facebookmarketing.de/zahlen_fakten/facebook-infografik-und-statistiken).

[Whi09] **White, Tom.** *Hadoop: The Definitive Guide*. First Edition. Sebastopol: O'Reilly, 2009. ISBN 978-0-596-52197-4.