

Abstract Oberseminar “Datenbanksysteme - Aktuelle Trends”

Cloud Datenbanken

Tina Neumann

24.07.2017

Einführung Cloud Computing

In [NIST] wird Cloud Computing als ein Modell definiert, welches den bedarfsgesteuerten Netzwerkzugriff zu einem geteilten Pool konfigurierbarer Ressourcen (wie Netzwerk, Server, Speicher und Applikationen) ermöglicht. Dieses Modell lässt sich in drei, aufeinander aufbauende Service Ebenen unterteilen, wobei jeweils die strukturell darunterliegenden Ebenen (bei SaaS z.B. Plattform und Infrastruktur) vor dem Nutzer verborgen bleiben. Software as a Service (SaaS) bezeichnet die Bereitstellung von Applikationen (z.B. Microsoft Office 365), Platform as a Service (PaaS) bezeichnet die Bereitstellung von Laufzeit- und Entwicklungsumgebungen oder Betriebssystemen (z.B. Google App Engine). Infrastructure as a Service (IaaS) bezeichnet die Bereitstellung von Hardware Ressourcen (z.B. Amazon S3). Es können drei Bereitstellungsmodelle von Cloud Computing unterschieden werden. Eine Public Cloud ist öffentlich über das Internet zugänglich, die Infrastruktur wird nicht exklusiv genutzt. Eine Private Cloud wird auf isolierter Infrastruktur oder in einem virtuellen Netzwerk betrieben. Hybrid Clouds sind Mischformen aus Public- und Private Cloud.

Database as a Service

[RITTER, S.425] beschreibt Database as a Service (DBaaS) als die Bereitstellung von Datenbanken für mehrerer Mandanten in der Cloud. Aufgaben wie Installation, Updates, Monitoring, Backups und Zugangskontrolle werden dabei vom Anbieter erfüllt.

Cloud Data Management Allgemein

Prinzipien und Ziele

In [KRASKA, S.122] werden Automatisierung, Virtualisierung, sowie flexible Bezahlung und Bereitstellung als Grundprinzipien von Cloud Computing genannt, welche nur einen Mehrwert bieten wenn sie gemeinsam umgesetzt werden.

Laut [SAKR, S.4-5] ist das Ziel dabei möglichst viele Eigenschaften eines Systems wie Verfügbarkeit, Skalierbarkeit, Elastizität, Performance, Mandantenfähigkeit, Last- und Mandanten-Balancierung, sowie Fehlertoleranz zu verbessern.

Anbieter

Anbieter von Cloud Services betreiben in der Regel mehrere Datenzentren über die sie Infrastruktur-, Plattform- und Softwareservices bereitstellen. Die Datenzentren befinden sich in verschiedenen geografischen Regionen der Welt, wie die Google Map in [BUTTLER] zeigt. Bei der Auswahl eines Cloudproduktes muss der Nutzer eine Region auswählen aus der er den Service beziehen möchte, diese Wahl hat u.a. Einfluss auf den Preis des Produktes und die Latenz des Services. Die drei größten Cloud-Anbieter sind Google, Amazon und Microsoft.

Es besteht die Möglichkeit eigene Hardwareressourcen unabhängig von Cloud Anbietern selbständig zu verwalten. Die Open Source Software Openstack ermöglicht z.B. das Erstellen und Verwalten von Public und Private Clouds und die Kontrolle von Rechen-, Speicher- und Netzwerkressourcen über Dashboard oder OpenStack API [OPENSTACK]. Ein weiteres Beispiel ist die Firma LinkedIn, welche eigene Datenzentren betreibt. Im offiziellen Entwickler Blog der Firma wird in [FATTAH] z.B. die Automatisierung und Steuerung eines Couchbase Clusters mit SaltStack beschrieben.

Die Verwendung von Cloud Produkten mehrerer Anbieter in einem System wird als Multi-Cloud bezeichnet. Der Administrationsaufwand steigt zwar in einer Multi-Cloud Umgebung aufgrund der Komplexität, der Nutzer geht jedoch keine Abhängigkeit mit nur einem bestimmten Anbieter ein. Weiterhin wird ihm durch Multi-Cloud ermöglicht das günstigste Produkt zu einem Workload auszusuchen und ältere Systeme zu integrieren [RACKSPACE]. Rackspace ist ein Provider der sowohl Cloud Produkte großer Anbieter wie Amazon mit zusätzlichem Support als auch Openstack-Lösungen anbietet. Verglichen mit z.B. [BRADER] ist das Multi-Cloud Angebot von Rackspace derzeit das vielseitigste.

Speicher Kategorien

Blob Storage

Blob (binary large object) Speicher dienen zur Speicherung großer Binär- oder Textdateien wie z.B. Bildern, Videos, Software oder XML-Dokumenten. Ein Nutzer kann mehrere eindeutig benannte Container anlegen. Jeder Container enthält beliebig viele Blobs bzw. Dateien. Der Zugriff erfolgt über SOAP oder REST Protokolle und die Adressierung per URI. Beispiele für Blob Storages sind Amazons Simple Storage Service (S3) und Microsofts Azure Blob Storage [JAEGER, S.689-690].

Table Storage

Der Table Storage dient zur Speicherung strukturierter Daten in großen Tabellen. Ein Datensatz besteht dabei aus einer eindeutigen ID sowie einer Menge von Attribut- und Wert-Paaren. Table Storages sind NoSQL Datenbanken ohne Vordefiniertes Schema, dieses ergibt sich dynamisch aus den Datensätzen. Operationen und Anfragen beziehen sich immer auf genau eine Tabelle, deshalb sind keine Joins zwischen Tabellen möglich.

Der Zugriff erfolgt wie beim Blob Storage über SOAP oder REST Protokolle. Die Adressierung erfolgt ebenfalls über eine URI, wobei z.B. GET Parameter verwendet werden um Datensätze der Tabelle zu filtern. Google's Big Table, Amazon SimpleDB und Azure Table Service sind Beispiele für Table Storages [JAEGER, S.691].

Datenbankserver

In der Kategorie Datenbankserver stellt ein physischer Datenbankserver virtuelle Server für mehrere Nutzer bereit, das heißt es gibt keine Garantie ob der physische Server exklusiv genutzt wird. Die Performanz wird indirekt über die Auswahl der Instanzgröße gesteuert. Der Anbieter legt Parameter wie z.B. Anzahl virtueller CPUs oder RAM in GB für jede Instanzgröße bzw. jeden Instanztyp fest. Beispiele für solche Datenbankserver sind Microsofts Azure SQL mit SQLServer oder Amazon RDS bei dem die konkrete Datenbankengine vom Nutzer festgelegt wird (Amazon bietet unter anderem MySQL und PostgreSQL an). Der Zugriff erfolgt über APIs wie JDBC oder ADO.NET (vergleichbar mit herkömmlichen relationalen Datenbankservern) und zusätzlich eine vom Anbieter generierte URL welche den Servernamen enthält [JAEGER, S.692-693].

Eigenschaften

Multi-tenancy

[RITTER, S.425-426] beschreibt Multi-tenancy oder Mandantenfähigkeit als die gemeinsame Nutzung von Ressourcen zum Zweck der Kostenminimierung und effizienter Ressourcenausnutzung durch verschiedene Mandanten. Die Herausforderung hierbei ist die Aufrechterhaltung der Isolation der einzelnen Mandanten. Multi-tenancy kann in vier Stufen unterteilt werden: privates Betriebssystem, private Datenbank, geteilte Datenbank mit privatem Schema und geteilte Tabelle. Die Stufen beschreiben welche Komponente für

jeden einzelnen Mandanten exklusiv ist, wie stark also die Mandanten isoliert sind. Bei der geteilten Tabelle ist die Isolation am geringsten, beim privaten Betriebssystem am größten.

Skalierbarkeit

Skalierbarkeit bezüglich Anfragelast und Datenvolumen ist eine Schlüsselanforderung an Cloud Datenbanken. Hierbei sind besonders die Grundtechniken Replikation, Partitionierung und Scale-Out relevant [RITTER, S.426-428]. Bei Blob und Table Storages werden REST oder SOAP Anfragen von einem Webserver entgegengenommen, die Lastenverteilung wird dabei durch den Anbieter gesteuert, der Nutzer hat durch Partitionierung nur indirekt Einfluss auf die Skalierung [JAEGER, S.696-697].

Bei Datenbankservern erfolgt vertikale Skalierung durch Erhöhung der Rechenleistung bzw. Wahl einer größeren Instanzklasse. Horizontale Skalierung, also das Hinzufügen und Entfernen von Knoten wird in NoSQL Datenbanken wie z.B. Couchbase oder Dynamic DynamoDB [AWS01] durch Shared Nothing Architektur berücksichtigt. Bei RDBS existieren unterschiedliche Ansätze der Cloud Datenbank Anbieter. Amazon RDS Datenbanken werden durch das Hinzufügen von read-only Replica-Servern horizontal skaliert, was nur für Systeme mit vielen Lese-Operationen geeignet ist. Zusätzlich ist hier der Einsatz eines Load-Balancers wie z.B. das MySQLnd Plugin für MySQL, Layer 4 mit MySQL Connector oder Layer 7 SQL-aware Load Balancer notwendig, Amazons Elastic Load Balancing (ELB) wird nicht unterstützt [AWS]. Microsoft bietet zur horizontalen Skalierung von Azure SQL mehrere Datenbankserver mit Sharding-Architektur und Elastic Database features. Dabei kommen zusätzliche proprietäre Technologien wie Elastic Database client library zur Verwaltung von Shard-Sets, Split-Merge tool für den Datentransport zwischen Shards, Elastic Database query für Abfragen über mehrere Shards sowie Elastic transactions für Transaktionen über mehrere Datenbanken zum Einsatz [AZURE1].

Elastizität

Je nach Speicherkategorie ist das Datenvolumen unterschiedlich begrenzt und elastisch. Bei Blob Storages sind Anzahl der Container je Kunde oder eine maximale Blob-Größe festgelegt, wobei in der Regel beliebig viele Blobs je Container möglich sind [JAEGER, S.694]. Bei Table Storages sind Anzahl möglicher Tabellen und die maximale Tabellengröße (z.B. Amazon SimpleDB 250 Domains x 10 GB = 2,5TB [AWS03]) begrenzt. Bei Datenbankservern wird die Größe initial festgelegt, bei Amazon beispielsweise durch Auswahl der Instanzklasse. Wenn der Speicher nicht mehr ausreicht kann eine neue Obergrenze festgelegt werden. Die Maximalgröße von SQL Azure (elastische DB, Basic Tarif) liegt bei 156GB pro Pool und maximal 500 DBs pro Pool [AZURE2].

Hochverfügbarkeit / Ausfallsicherheit

Für die Hochverfügbarkeit und Ausfallsicherheit von Cloud Speichern sind Replikationsstrategien sowie Backup und Restore Möglichkeiten ausschlaggebend. Bei Blob- und Table Storages werden die Daten in der Regel durch den Anbieter redundant auf verschiedenen Standorten vorgehalten, ein Failover bei Ausfall eines Speichers geschieht automatisch [JAEGER, S.694]. Die Verfügbarkeit wird in den SLAs geregelt. Für Amazon S3 gibt der Anbieter eine Objektlebensdauer von 99,99999999 % und eine Objektverfügbarkeit von 99,99 % an [AWS04]. Der Nutzer erhält eine Gutschrift von bis zu 25% wenn die Systemverfügbarkeit unter 99,0% im Monat sinkt [AWS05]. Bei der Einrichtung von

Datenbankservern muss der Nutzer einen Zeitraum für Wartungen festlegen (z.B. bei Amazon RDS 30 Minuten pro Woche [AWS06]) . In diesem Zeitraum spielt der Anbieter Patches und Updates ein, sodass es kurzfristig zum Systemausfall kommen kann. Das Erzeugen und Einspielen von Backup Snapshots kann automatisch durchgeführt oder durch den Nutzer initiiert werden.

Administrationsaufwand

Besonderer Vorteil von Cloud Datenbanken ist der geringe Administrationsaufwand für den Nutzer. Hardware-Administration, Software-Updates und Patches, sowie die automatische Indizierung bei Table-Storages werden vom Anbieter durchgeführt. Dem Nutzer entstehen jedoch Aufwände für das Registrieren und Anlegen von Benutzerkonten, Indizierung bei Datenbankservern, sowie Überwachung von Ressourcenauslastung und Reaktion bei Erreichen von Limits [JAEGER, S.696]. Letztere sind durch vom Anbieter bereitgestellte APIs, CLIs & SDKs automatisierbar.

Kosteneinsparung

Elastizität der Speicherprodukte und das flexible Bereitstellungsmodell des Anbieters ermöglichen es dem Nutzer bei Bedarf kurzfristig Ressourcen hinzuzufügen und freizugeben. Dadurch wird es überflüssig Hardwareressourcen auf Stand-by zu betreiben, die nur in Spitzenzeiten (z.B. bei kurzfristig ansteigender Anfragelast) tatsächlich genutzt werden. Der Preis einer Cloud Datenbank ist außerdem abhängig von der ausgewählten Region und Instanzgröße bzw. Produktklasse und unterliegt häufigen Änderungen. Dies ermöglicht dem Nutzer sein System beispielsweise in einer kostengünstigeren Region zu betreiben, was jedoch Auswirkungen auf die Latenz hat. Zusätzlich zu den Speicherkosten fallen Kosten für Datentransfer an. Bei Datenbankservern kann der Nutzer durch die vom Anbieter bereitgestellte Ausfallsicherheit und Administration Kosten einsparen [JAEGER, S.695-696].

Für Datenbankserver die permanent genutzt werden bietet Amazon Web Services sogenannte Reserved Instances an, die aufgrund ihrer Vertragslaufzeit von bis zu 3 Jahren eine Kosteneinsparung von bis zu 65% gegenüber On-Demand versprechen [AWS07].

Amazon und Microsoft ermöglichen kostenlose Tests der Produkte in Form von Freibeträgen über die Laufzeit von einem Jahr [AWS08][AZURE2]. Weiterhin können einige Produkte bis zu einem bestimmten Limit völlig kostenfrei genutzt werden, Bei Amazons SimpleDB sind beispielsweise 1GB und 25 Maschinenstunden pro Monat kostenfrei [AWS09].

Allgemeine Probleme

Performanz

Es existieren wenige aktuelle Vergleiche der Performanz äquivalenter Produkte unterschiedlicher Anbieter [KRASKA, S.125]. Besonders Drosselung und Indizierung sind für die Performanz relevant. Die Indizierung, welche üblicherweise von einem Datenbankadministrator durchgeführt wird, ist bei Table Storages beispielsweise nicht möglich. Der Nutzer hat keinen direkten Einfluss auf die Datenorganisation. Weiterhin wird die Performanz durch den Anbieter gedrosselt um Produktklassen bzw. SLAs voneinander abzugrenzen, unausgewogene Ressourcennutzung zu vermeiden und ggf. DOS-Attacken zu verhindern. [JAEGER, S.697-698]

Konsistenzkontrolle

Das CAP-Theorem nach Brewer, Gillbert und Lynch beschreibt Konsistenz (C), Verfügbarkeit (A) und Partitionstoleranz (P) als grundlegende Eigenschaften von verteilten Datenbanksystemen und besagt, dass jeweils höchstens zwei dieser drei Eigenschaften erfüllt werden können, woraus sich die drei Systemklassen AP, CP und CA ergeben. Das PACELC-Theorem erweitert CAP um die Unterscheidung zwischen Normalbetrieb (E) und Fehlerfall bzw. Netzwerkpartition (P) und ist damit ein vollständigeres Modell für verteilte Systeme. Bei vorliegender Netzwerkpartition entscheidet das System zwischen Verfügbarkeit und Konsistenz, im Normalbetrieb zwischen Latenz (L) und Konsistenz. [RITTER, S.424-425]

[MOHAMMAD, S.4] beschreibt vier Ansätze zur Umsetzung von CAP bzw. PACELC: ACID (Atomicity, Consistency, Isolation, Durability), BASE (Basically Available, Soft-state, Eventual consistency), SCLA (Strongly Consistent, Loosely Available) und Tunable consistency. Bei ACID Systemen werden Transaktionen immer vollständig oder gar nicht ausgeführt, sodass Konsistenz vor und nach der Transaktion garantiert ist. Dies kommt vor allem in relationalen DBMS zum Einsatz, welche hohe Konsistenzanforderungen haben, wie beispielsweise Amazon RDS. BASE Systeme wie z.B. Amazons SimpleDB und S3 setzen einen stärkeren Fokus auf Verfügbarkeit. Es gibt dabei keine Garantie dass alle User die selbe Version eines Datensatzes erhalten, eine Antwort ist aber garantiert. SCLA Systeme wie Google BigTable vernachlässigen die Verfügbarkeit um dafür eine stärkere Konsistenz als BASE und eine höhere Skalierbarkeit als ACID zu erreichen. Bei Systemen mit Tunable consistency wie Cassandra ist die Konsistenz und Verfügbarkeit konfigurierbar, die Client Applikation entscheidet hierbei über die Konsistenz für jede read / write Operation. Aus der Grafik [MOHAMMAD, "Figure 5: Classification of cloud data management systems based on CAP"] geht hervor, dass ACID vorwiegend in relationalen Systemen eingesetzt wird, wohingegen die anderen drei Ansätze vor allem in NoSQL Systemen zum Einsatz kommen.

Partitionierung (Sharding)

Partitionierung ist besonders für die Verfügbarkeit und Performanz relevant. Nach [MOHAMMAD, S.4] existieren verschiedene Techniken um die Verteilung der Daten auf Shard Server zu realisieren. Range Partitionierung ordnet jedem Shard Server einen geordneten zusammenhängenden Bereich von Daten zu. Bei der Hash Partitionierung erfolgt die Zuordnung mit Hashwerten die aus Primärschlüsseln berechnet werden. Eine weitere Strategie ist das Consistent Hashing, wobei Hashwerte auf einem logischen Ring angeordnet und den Shard-Servern Ringabschnitte zugeordnet werden.

Replikation

Replikationsverfahren werden in eine zeitliche und eine räumliche Dimension unterteilt, dabei sind zeitliche und räumliche Strategien in einem System beliebig kombinierbar. Zeitliche Verfahren sind Eager- und Lazy Replikation. Bei Eager Replikation werden Änderungen synchron auf alle Replikate angewandt, wohingegen bei Lazy Replikation Änderungen verzögert propagiert werden, was höhere Performanz aber geringere Konsistenz zur Folge hat. In der räumlichen Dimension wird der Primary-Copy (Master-Slave) Ansatz, bei welchem Änderungen immer über einen Master Knoten an die Replikate der Slave Knoten verteilt werden, vom Update-Anywhere (Multi-Master) Ansatz,

bei welchem jeder Knoten Änderungen durchführen darf, unterschieden. [RITTER, S.427-428]

Migration

Der Migrationsaufwand ist abhängig von Art des Quell- und Zielsystems und davon, welche Datentypen und Datenmodelle von beiden unterstützt werden. Gegebenenfalls muss die Anwendungslogik angepasst werden wenn das Zielsystem z.B. nur String Typen unterstützt, die Anwendung aber integer Werte aus dem Zielsystem ausliest und verarbeitet [JAEGER, S.699]. Bei der Migration von RDBMS nach RDBMS wird der Nutzer durch Migrationstools unterstützt, teilweise ist dies auch automatisiert möglich [AWS10][AZURE3]. Die Migration von RDBMS nach NoSQL oder umgekehrt erfordert zusätzlich eine manuelle Anpassung des Schemas. In [AWS11] werden verschiedenen Probleme und Lösungsmöglichkeiten detailliert beschrieben, darüber hinaus schlägt Amazon eine aus fünf Phasen bestehende Migrationsstrategie vor.

Sicherheit & rechtliche Aspekte

Aus dem Artikel [BITKOM] geht hervor, dass Unternehmen vor allem Bedenken hinsichtlich Datensicherheit (bzgl. Angriffen und Verlust) und unklarer Rechtslage haben. Prof. Dieter Kempf hebt in diesem Artikel hervor, dass das Sicherheitsniveau durch Nutzung von Cloud Diensten zu moderaten Kosten signifikant erhöht werden kann, wovon vorwiegend kleine und mittelständische Unternehmen profitieren.

Im Umgang mit personenbezogenen Daten ist das Bundesdatenschutzgesetz sowie die EU Richtlinie 95/46/EG relevant. Die Gefahr einer Rechtsverletzung besteht vor allem dann, wenn sich Rechenzentren im Ausland befinden. Für die anwendbaren Gesetze sind Sitz von Anbieter und Nutzer relevant, unter Umständen aber auch die Herkunft Dritter (Datensubjekte oder Subunternehmer). [NIEMANN]

Vor- und Nachteile

Ein System in der Cloud zu betreiben wirkt sich vorteilhaft auf Kosteneffizienz (durch Automatisierung und Virtualisierung), Skalierbarkeit, Verfügbarkeit und Ausfallsicherheit des Systems und die Reaktionsfähigkeit des Systembetreibers aus. Auch auf die Energieeffizienz hat die Cloud einen positiven Effekt. So zeigt [SALVOA] wie durch den Umbau des Datenverarbeitungskomplexes einer Universität zur Cloud 50% weniger Energie zur Bereitstellung einer VM benötigt wird und sogar 85% weniger um 1 Byte zu speichern. Auch Google teilt sein Wissen auf dem Gebiet der Energieeffizienz und ist Mitglied in verschiedenen Initiativen und Arbeitsgruppen.[GOOGLE]

Die Punkte Sicherheit und Rechtslage sind hingegen umstritten, da sich auf die Sicherheit des Anbieters verlassen werden muss und verschiedenste Gesetze zum tragen kommen, welche gerade hinsichtlich des Datenschutzes in der näheren Vergangenheit immer wieder Änderungen unterlagen. Auch die Abhängigkeit bezüglich Infrastruktur und Software des Anbieters ist ein Nachteil für den Nutzer. Der Vergleich zahlreicher Angebote und das schwere Verständnis detaillierter Service Level Agreements ist möglicherweise auch eine Hürde für Nutzer im Umgang mit Cloud Services.

Literaturverzeichnis

- [AWS01]** J. Barr: Auto Scale DynamoDB With Dynamic DynamoDB, Amazon Web Services, 2014. <https://aws.amazon.com/de/blogs/aws/auto-scale-dynamodb-with-dynamic-dynamodb/> (Abgerufen am: 20.07.2017)
- [AWS02]** M. Yap: Scaling Your Amazon RDS Instance Vertically and Horizontally, Amazon Web Services, 2016. <https://aws.amazon.com/de/blogs/database/scaling-your-amazon-rds-instance-vertically-and-horizontally/> (Abgerufen am: 20.07.2017)
- [AWS03]** o.A.: Amazon SimpleDB – Produktdetails, Amazon Web Services, 2017. <https://aws.amazon.com/de/simpledb/details/> (Abgerufen am: 20.07.2017)
- [AWS04]** o.A.: Amazon S3-Speicherklassen, Amazon Web Services, 2017. <https://aws.amazon.com/de/s3/storage-classes/> (Abgerufen am: 20.07.2017)
- [AWS05]** o.A.: Amazon S3 Service Level Agreement, Amazon Web Services, Stand: September 2015. <https://aws.amazon.com/de/s3/sla/> (Abgerufen am: 20.07.2017)
- [AWS06]** o.A.: Amazon RDS – Häufig gestellte Fragen, Amazon Web Services, 2017. <https://aws.amazon.com/de/rds/faqs/> (Abgerufen am: 20.07.2017)
- [AWS07]** o.A.: Amazon RDS – Preise, Amazon Web Services, 2017. <https://aws.amazon.com/de/rds/pricing/> (Abgerufen am: 20.07.2017)
- [AWS08]** o.A.: Kostenloses Kontingent für AWS, Amazon Web Services, 2017. <https://aws.amazon.com/de/free/> (Abgerufen am: 20.07.2017)
- [AWS09]** o.A.: Amazon SimpleDB – Preise, Amazon Web Services, 2017. <https://aws.amazon.com/de/simpledb/pricing/> (Abgerufen am: 20.07.2017)
- [AWS10]** o.A.: AWS Database Migration Service, Amazon Web Services, 2017. <https://aws.amazon.com/de/dms/> (Abgerufen am: 20.07.2017)
- [AWS11]** N. Slater: Best Practices for Migrating from RDBMS to Amazon DynamoDB, Amazon Web Services, 2015. Verfügbar unter: <https://d0.awsstatic.com/whitepapers/migration-best-practices-rdbms-to-dynamodb.pdf> (Abgerufen am: 20.07.2017)
- [AZURE1]** D.Dove et al.: Scaling out with Azure SQL Database, Microsoft, 2016. <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-scale-introduction> (Abgerufen am: 20.07.2017)
- [AZURE2]** o.A.: Preisrechner, Microsoft, 2017. <https://azure.microsoft.com/de-de/pricing/calculator/> (Abgerufen am: 20.07.2017)
- [AZURE2]** o.A.: Azure Free Trial, Microsoft, 2017. <https://azure.microsoft.com/en-us/offers/ms-azr-0044p/> (Abgerufen am: 20.07.2017)
- [AZURE3]** C. Rabeler et al.: SQL Server database migration to SQL Database in the cloud, Microsoft, 2017. <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-cloud-migrate> (Abgerufen am: 20.07.2017)
- [BITKOM]** o.A.: Nutzung von Cloud Computing in Unternehmen wächst, Bitkom, 2014. <https://www.bitkom.org/Presse/Presseinformation/Nutzung-von-Cloud-Computing-in-Unternehmen-waechst.html> (Abgerufen am: 20.07.2017)
- [BRADER]** L. Brader et al.: Azure and AWS for multicloud solutions, Microsoft, 2017. <https://docs.microsoft.com/en-us/azure/architecture/aws-professional/services#azure-and-aws-for-multicloud-solutions> (Abgerufen am: 20.07.2017)
- [BUTTLER]** B.Buttler: What's behind Amazon, Microsoft and Google's aggressive cloud expansions, Networkworld, 2016.

<http://www.networkworld.com/article/3131335/cloud-computing/what-s-behind-amazon-micro-soft-and-google-s-aggressive-cloud-expansions.html> (Abgerufen am: 20.07.2017)

[FATTAH] I. Fattah: Leveraging SaltStack to Scale Couchbase, LinkedIn, 2016.

<https://engineering.linkedin.com/blog/2016/04/leveraging-saltstack-to-scale-couchbase>

(Abgerufen am: 20.07.2017)

[GOOGLE] o.A.: Effizienz von Rechenzentren, Google, o.J.

<https://www.google.com/about/datacenters/efficiency/> (Abgerufen am: 20.07.2017)

[JAEGER] M.C. Jaeger, U. Hohenstein: Cloud Storage: Wie viel Cloud Computing steckt dahinter?, in: 14. Fachtagung Datenbanksysteme (BTW), Kaiserslautern, 2011.

[KRASKA] D. Kossmann, T. Kraska: Data Management in the Cloud: Promises, State-of-the-art, and Open Questions, in: Datenbank-Spektrum Bd. 10, Heft 3/Dezember 2010, Springer.

[MOHAMMAD] S. Mohammad, S. Breß, E. Schallehn: Cloud Data Management: A Short Overview and Comparison of Current Approaches, 2012. Verfügbar unter:

<https://pdfs.semanticscholar.org/0ac7/f0424724697b24ef4fe23a332dff66f6e73e.pdf>

(Abgerufen am: 20.07.2017)

[NIEMANN] P. Niemann (Hrsg): Rechtsfragen des Cloud Computing, Walter de Gruyter, 2014.

[NIST] P. Mell, T. Grance: The NIST definition of cloud computing. National Institute of Standards and Technology, 2011. Verfügbar unter:

<http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf> (Abgerufen am: 20.07.2017)

[OPENSTACK] Openstack, o.J. <https://www.openstack.org/> (Abgerufen am: 20.07.2017)

[RITTER] N. Ritter: Cloud-Datenbanken, in: T. Kudraß (Hrsg): Taschenbuch Datenbanken, 2. Auflage, Carl Hanser, 2015.

[RACKSPACE] Was ist Multi-Cloud, Rackspace, 2017.

<https://www.rackspace.com/de-de/cloud/multi-cloud> (Abgerufen am: 20.07.2017)

[SAKR] S. Sakr et al.: A Survey of Large Scale Data Management Approaches in Cloud Environments, IEEE, 2011. Verfügbar unter:

https://www.researchgate.net/publication/220250172_A_Survey_of_Large_Scale_Data_Management_Approaches_in_Cloud_Environments (Abgerufen am: 20.07.2017)

[SALVOA] A. L.A. Di Salvoa et al.: Can cloud computing be labeled as "green"? Insights under an environmental accounting perspective, in: Renewable and Sustainable Energy Reviews, Volume 69, 2016.