Extended Abstract Obserseminar: Datenbanksysteme - Aktuelle Trends

Cloud-Datenbanken

Franz Anders

02.07.2015

Dies ist das erweiterte Abstract zum Vortrag "Cloud-Datenbanken"für das Oberseminar "Datenbanksysteme - aktuelle Trends". Der Zweck dieses Schriftstückes ist es, einen Überblick über die im Vortrag besprochenen Themen zu bieten. Für jedes Kapitel werden Links und Verweise zu weiterführenden Artikeln gegeben, die eine tiefere Einarbeitung ermöglichen. Diese dienen gleichzeitig als Quellenangaben.

1 Einführung in das Cloud-Computing

Cloud-Computing beschreibt die Lieferung von Computing-Services über das Internet. Anstatt die Hardware und Software des eigenen PCs oder des firmeninternen Netzwerkes zu verwenden, wird ein Service von einem darauf spezialisierten Anbieter zur Verfügung gestellt und der Zugang zu diesem über das Internet ermöglicht. Es is schwierig, im Internet bereitgestellte Dienste eindeutig dem Cloud-Computing zuzuordnen. Es gibt keine scharfe Definition. Es gibt jedoch Eigenschaften, die dem Cloud-Computing zugeschrieben werden. Diese werden im folgenden vorgestellt.

1.1 1. Services über Cloud-Computing

Cloud-Services lassen sich in drei Kategorien einteilen. **Software-as-a-Service (SaaS)** beschreibt die Anmietung von Hardware, einem Betriebssystem und einer Anwendung. Der Nutzer hat nur Zugriff auf eine Softwareanwendung, die in der Cloud läuft. **Platform-as-a-Service (Paas)** beschreibt die Anmietung von Hardware und einem Betriebssystem. Dazu werden Programmier- und Laufzeitumgebungen angeboten, mit deren Hilfe

Software programmiert werden kann. **Infrastructure-as-a-Service (IaaS)** beschreibt die reine Anmietung von Hardware. Das Betriebssystem und die Anwendungen werden vom Benutzer selber installiert.

1.2 Hosting

Es gibt drei verschiedene Arten von Cloud-Anbietern: Public Clouds werden von Firmen angeboten, bei denen die Wirtschaftlichkeit im Vordergrund steht. Die Cloud ist jedem Nutzer zugänglich, der einen Vertrag mit dem Anbieter eingehen möchte. Private Clouds hingegen sind Clouds, die von einem Unternehmen aufgebaut werden und nur diesem über das Intranet zur Verfügung stehen. Community Clouds sind Clouds, die nur einem ausgewählten Nutzerkreis zur Verfügung stehen, das heißt sie sind offener als Private Clouds, aber geschlossener als Public Clouds.

1.3 Eigenschaften

Die folgenden Eigeschaften zeichnen Cloud-Computing aus:

- Gemanaged: Instandhaltung, Updating, Backups etc. der Software- und Hardwarekomponenten werden vom Cloudanbietern durchgeführt und müssen nicht vom Nutzer organisiert werden.

- Virtualisierung: Server werden zu Server-Farmen zusammengeschlossen, auf denen virtuelle Maschienen betrieben werden. Diese können frei die Hardwareressourcen nutzen. Die Software, die in den virtuellen Maschinen läuft, ist nicht mehr fest an einen Server gebunden.

- Skalierbarkeit: Meint die Eigenschaft einer Software- oder Datenbankinstanz, mit steigenden Anforderungen mitzuwachsen. Wird auch als Elastizität bezeichnet.

- "Pay as you go": Beschreibt das Bezahlmodell. Der Nutzer zahlt nur für die Ressourcen, die er beim Anbieter in Anspruch nimmt.

- Ausfalltoleranz: Der Cloudanbieter hat sicherzustellen, dass die Daten nicht verloren gehen können.

Weitere allgemeine Informationen zum Thema Cloud-Computing: `http://www.explainthatstuff.com/cloud-computing-introduction.html`
`https://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf`
`http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Cloud-Computing`

2 Database-as-a-Service

Klassische relationale Datenbanksystemen können den Bedürfnissen von Anwendungen mit besonders hohen Anforderungen bezüglich des Datenvolumens oder der Nutzerzahl nicht nachkommen. Es gibt daher ein Verlangen nach Datenbanken, die die Vorteile des Cloud-Computing (Skalierbarkeit, Elastizität, Ausfallsicherheit etc.) für sich nutzen können. Datenbanken, die in der Cloud als Service angeboten werden, werden als Database-as-a-Service bezeichnet. Es gibt zwei mögliche Szenarien, unter denen ein Nutzer eine Cloud Datenbank anmieten möchte: Eine Anwendung läuft bereits in der Cloud. Es macht Sinn, in diesem Fall ebenfalls eine Datenbank in der Cloud zu verwenden, da die Anwendung sonst aus der Cloud auf eine lokale Datenbank zugreifen müsste. Oder aber eine Anwendung läuft zwar lokal, verwendet aber eine Datenbank in der Cloud, um deren Vorteile gegenüber einer lokalen Datenbank zu nutzen.

Weitere Informationen zum Thema Database-as-a-Service: `http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Cloud-Datenbanken`

3 Architekturen

Das Ziel des Anbieters von Cloud-Datenbanken ist es, seine Ressourcen optimal zu nutzen, das heißt, möglichst viele Nutzer durch möglichst wenig Server unterhalten zu können. Dazu muss die Architektur der Datenbank angepasst werden. Es gibt zwei verschiedenen Ansätze: Multi-Instance und Multi-Tenancy. Bei einer Multi-Instance-Architektur erhält jeder Teilnehmer eine eigene Datenbankinstanz auf einem Server(cluster). Bei einer Multi-Tenancy-Architektur teilen sich mehrere Mandanten eine Datenbankinstanz. In jedem Fall scheint es für den Anwenden immer, als wenn er eine eigene Datenbankinstanz zur Verfügung gestellt bekommt. Es gibt dabei drei verschiedene Implementierungen dieser grundlegenden Ideen:

Shared Maschiene: Jeder Tenant erhält seine eigene Datenbankinstanz auf einem Server oder Servercluster. Alle Mandanten sind vollkommen voneinander isoliert und teilen sich nur die Hardwareressourcen. Der Nachteil daran ist, dass der Cloudanbieter eine schlechtere Ressourcennutzung als bei den anderen Implementierungsansätzen hat, da jede Datenbankinstanz unabhängig von der Menge der abgelegten Daten einen initialen Ressourcenverbrauch hat.

Shared Process: Mehrere Teilnehmer teilen sich eine Datenbankinstanz, erhalten aber verschiedene Datenbankenschemen. Die einzelnen Mandanten sind logisch voneinander getrennt. Bei diesem Ansatz gewinnt der Anbieter eine bessere Ressourcennutzung, da das Problem des initialen Ressourcenverbrauches getrennter Daten-

bankinstanzen entfällt.

Shared Table: Die Mandanten teilen sich einen Server(cluster), Datenbankinstanz und ihre Tabellen werden in einem Datenbankschema konsolidiert. Dabei hat jeder Teilnehmer ein eigenes, logisches Datenbankschema, die Datenbankinstanz selber hat jedoch nur ein einziges physisches Datenbankschema. Eine sogenannte Query-Transformation-Schicht mapt die logischen Schemen auf das physische Schema. So wird sichergestellt, dass jeder Teilnehmer nur auf seine eigenen Daten Zugriff hat, obwohl die Daten mehrerer Mandanten in einem Schema abgelegt werden. Es gibt verschiedene Implementierungstechniken dieser Idee, die unterschiedliche physische Schemen zur Konsolidierung der logischen Schemen vorschlagen.

Weiterführende Quellen, insbesondere hinsichtlich der Implementierungstechniken für den Shared-Table-Ansatz: `http://dbs.uni-leipzig.de/file/seminar_0910_kerkhoff_ausarbeitung.pdf`
`http://www.informatik.uni-jena.de/dbis/lehre/ss2010/saas/material/Ausarbeitung03-Kobold.pdf`
Ritter: Cloud-Datenbanken, in: T. Kudraß (Hrsg): Taschenbuch Datenbanken, 2.Auflage, Carl Hanser, 2015

4 Allgemeine Probleme von Cloud Data Management

4.1 Verteilte Datenbanksysteme

Bei verteilten Datenbanksystemen werden die Daten verteilt auf mehreren Datenbankknoten gespeichert. Sie werden eingesetzt, sobald Ein-Server-Datenbanken an Ihre grenzen stoßen, weil beispielsweise das Volumen der Daten zu groß ist. Bei Cloud-Datenbanken handelt es sich praktisch immer um verteilte Datenbanksysteme.

Weitere Informationen zu verteilten Datenbanksystemen: `http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Verteilte-Datenbank`

4.2 Skalierbarkeit

Skalierung meint das Vermögen einer Datenbank, mit steigenden Anforderungen mitzuwachsen. Bei der sogenannten "vertikalen Skalierung"wird ein Server durch Aufrüstung der Hardware und Softwareupdates leistungsfähiger gemacht. Bei Cloud-Datenbanken hat sich jedoch die horizontale Skalierung durchgesetzt (SScale out"). Dabei werden

dynamisch mehr Server in den Verbund geschalten. Grundlage für die Möglichkeit der horizontalen Skalierung ist die Virtualisierung.

Weitere Informationen zu Skalierbarkeit: `http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Skalierbarkeit`

4.3 Partitionierung

Partitionierung wird verwendet, um Skalierbarkeit zu erreichen. In der Praxis gibt es Datenbanken, in denen eine einzige Tabelle eine Millionen Einträge hat. Diese haben dementsprechend hohe Lese- und Schreibzeiten. Zur Verbesserung dieser Situation wird Partitioniert, also ein Tabelle horizontal oder vertikal in Untertabellen aufgeteilt und die Partitionen verschiedenen Speicherbereichen zugeordnet. Diese Partitionierung ist vollständig und disjunkt. Im Zusammenhang mit horizontaler Skalierung werden die verschiedenen Partitionen verschiedenen Datenbankknoten zugeordnet.

Weitere Informationen zum Thema Partitionierung: `https://homepages.thm.de/~hg10013/Lehre/MMS/SS02/Brot/text.htm` `http://ceur-ws.org/Vol-850/paper_mohammad.pdf`

4.4 Replikation

Replikation meint das redundante Speichern von Daten in verschiedenen Datenbankknoten zur Erhöhung der Ausfallsicherheit und Verfügbarkeit der Daten. Es gibt einige Designentscheidungen bezüglich der Replikation zu fällen: Sind Daten nach einem Write schon verfügbar, bevor alle Duplikate geupdatet wurden? Wer kann Änderungen an die Duplikate propagieren? Wann werden Konflikte zwischen den Duplikaten aufgelöst?

Weitere Informationen zum Thema Replikation: `http://ceur-ws.org/Vol-850/paper_mohammad.pdf`
Ritter: Cloud-Datenbanken, in: T. Kudraß (Hrsg): Taschenbuch Datenbanken, 2.Auflage, Carl Hanser, 2015.

4.5 CAP-Theorem

Das CAP-Theorem spielt bei verteilen Datenbanksystemen eine Rolle. Es besagt, dass von den drei Eigenschaften Consistency, Availability und Partition Tolerance von einem verteilten Datenkbanksystem höchstens zwei vollkommen erfüllt werden können.

Consistency meint die Konsistenz der gespeicherten Daten. Es muss sichergestellt werden, dass nach einem Schreibzugriff alle Replikate des betroffenen Datensatzes aktualisiert werden. **Availability** meint Verfügbarkeit im Sinne akzeptabler Antwortzeiten auf jedwede Anfrage. **Partition Tolerance** meint die Ausfalltoleranz der Rechner-/Servernetze. Je nachdem, welche der beiden Eigenschaften erfüllt werden, wird in AP, AC und CP-Systeme unterschieden.

Weitere Informationen zum Thema CAP-Theorem: `http://ceur-ws.org/Vol-850/paper_mohammad.pdf`

5 Speicherkategorien in der Cloud

Man unterscheidet prinzipiell drei Arten der Datenspeicherung in Clouds: Blob, Table, und Datenbank.

5.1 Datenbankserver

Es wird ein virtueller Datenbankserver für jeden Kunden zur Verfügung gestellt. Dieser bietet einen ähnlichen Funktionsumfang wie klassische relationale Datenbank. Es handelt sich also um eine fremdverwaltete Instanz eines Datenbankservers. Ein Datenbankserver muss dabei aufgrund der Virtualisierung nicht einem physischen Rechner zugeteilt sein. Diese Datenbanken werden vor allem von Software genutzt, welche in der Cloud laufen und keine riesigen Datenmengen produzieren.

5.2 Table-Storage

Bei Table Storages handelt es sich um NoSQL-Datenbanken.Er dient der Speicherung strukturierter Daten in einer großen, unstrukturierten Tabelle, genannt Big Table. Die Tabellenstruktur ergibt sich dynamisch aus den gespeicherten Inhalten. In Folge dessen entfallen Joins. Der Zugriff erfolgt über REST und SOAP-Requests.

5.3 BLOB-Storage

BLOBS-Storages sind gedacht für die Speicherung großer Binär- und Textdaten wie Bilder, Musik, Software und XML-Dokumente. Grundkonzept sind Container, die einen eindeutigen Namen haben und die BLOBs enthalten. Ein Container kann keine weiteren

Container enthalten. Ein BLOB enthält die eigentlichen Objektdaten und Metadaten. BLOBS werden über REST- und SOAP-Requests angesprochen.

Weitere Informationen zu den Speicherkategorien: M.C. Jaeger, U. Hohenstein: "Cloud Storage. Wie viel Cloud Computing steckt dahinter?Ïn: 14. Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW), Kaiserslautern, 2011

6 Anbieter von Cloud-Datenbanken

Es gibt drei "Big Player": Amazon, Google, und Microsoft. Alle haben sowohl ein allgemeines Angebot zu Cloud-Computing, als auch ein auf Datenbanken spezialisiertes Cloud-Angebot: Alle bieten eine SQL-Datenbank, BLOBs-Storages als auch NoSQL-Datenbanken an. Bei den NoSQL-Datenbanken kann jeweils zwischen einem AP und einem CP-Profil hinsichtlich des CAP-Theorems entschieden werden. Amazon und Google bieten darüber hinaus Infrastructure-as-a-Service-Angebote, welche vom Nutzer wahlweise mit einer Datenbank ausgestattet werden können.

Weitere Informationen zu den Anbietern von Cloud-Datenbanken: `http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Anbieter-Cloud-DB`