GIS-Technologie-News

SOGI Informationsblatt 1/2011

NoSQL-Datenbanksysteme

Unter dem Begriff NoSQL werden gegenwärtig zahlreiche Kategorien und Produkte angeboten, so dass ein gewisser Klärungsbedarf besteht. Dieser Beitrag gibt eine grobe Übersicht zum Thema.

Was ist NoSQL?
Unter NoSQL wird eine Bewegung verstanden, die relativ jung ist. Viele entsprechende Datenbanksystem-Produkte und -Techniken gibt es schon seit einigen Jahren – auch in relationalen Datenbankmanagementsystemen (RDBMS). Neu ist vor allem der Anwendungsfall.

NoSQL wird von der Bewegung meist mit ‚not only SQL' übersetzt, obschon vielleicht ‚NoREL' passender wäre. Der Zusammenhang von NoSQL und RDBMS kann wie folgt veranschaulicht werden: In RDBMS sind die Daten in Tabellen organisiert. Einzelne Datensätze müssen eindeutig sein (Primärschlüssel) und Tabellen können über Beziehungen (Fremdschlüssel) miteinander verbunden werden. RDBMS sind jedoch schwer skalierbar und können nur umständlich über Cluster in der Cloud betrieben werden.

Gemäss Edlich ([1]) besteht die wichtigste philosophische Idee hinter NoSQL darin, Vorreiter zu sein für eine freie Datenbanksystem-Auswahl ohne langfristige Bindungen an bestimmte Hersteller. Ebenfalls an die Adresse der Benutzer wird empfohlen, noch mehr auf die Analyse der Daten und der Anforderungen zu fokussieren.

Beispiel
NoSQL-Datenbanksysteme der Kategorie „Schlüssel-Wert-Speicherung" (Key/Value Store) können zum Beispiel einen Schlüssel und dessen Wert effizient verwalten wie in einem Array. Ein anschauliches Beispiel für eine Schlüssel-Wert Speicherung ist die Windows Registry. Ein wichtiges Merkmal der NoSQL-Bewegung ist u.a., dass die Datenspeicher verteilt sind und sich daher gut für den Betrieb in einer Cloud eignen.

Generell sind die Anfragesprachen in NoSQL nicht genormt. Deklarative Abfragen und Views, wie sie mit SQL möglich sind, gibt es in NoSQL nicht. Eine SQL-Abfrage wird typischerweise über das REST-Prinzip formuliert, also: `http://<Server>:<Port>/<DBName>/<Ressource>` (die spitzigen Klammern sind Platzhalter). Folgende SQL-Anfrage auf eine „Kontaktadresse":

```
SELECT * FROM Kontaktadresse WHERE Name='Meier'
```

muss zum Beispiel in Microsofts NoSQL-Datenbanksystem „Azure Table Storage" wie folgt formuliert werden:

```
GET http://mydomain.ch/Kontaktadresse()?$filter=(Name='Meier')
```

Entstehung
NoSQL wurde durch die Anwendungsfälle der Web 2.0-Technologien vorangetrieben. Dort entstand das Bedürfnis, grosse Datenmengen in hinreichend schneller Zeit zu verarbeiten. Dazu kam, dass auch Schreibzugriffe zugelassen werden mussten. Google nahm ab ca. 2004 eine Vorreiterrolle ein mit seinen proprietären Eigenentwicklungen. Dabei spielte das von Google entwickelte Map/Reduce-Verfahren eine zentrale Rolle. Damit lassen sich grosse Datenmengen, die in verteilten Datenspeichern lagern, parallel durchsuchen. NoSQL Produkte wie CouchDB und HBase nutzen dieses Verfahren. Firmen wie Yahoo, Amazon und später auch Sozialnetzwerke wie MySpace und Facebook zogen nach.

Kategorien
Es werden folgende Kategorien von NoSQL-Datenbanksystemen unterschieden:

1. Key/Value Stores: verwalten Schlüssel-Werte-Paare (z.B. Azure Table Storage, Berkeley DB, Membase)
2. Document Stores: verwalten Dokumente, wobei der Dokumentbegriff mehr als nur Office-Dokumente umfasst, sondern auch Schlüssel-Werte-Paare, die zudem geschachtelt sein können (z.B. CouchDB, MongoDB)
3. Graph Databases: verwalten Graphen-Strukturen (z.B. Neo4J, FlockDB)

4. Column Oriented Databases, Wide Column Stores: verwalten Tabelleninhalte in Form von Spalten (z.B. HBase, Cassandra)
5. Andere: Objektdatenbanken (z.B. db4o) und reine XML-Datenbanken (z.B. Tamino, eXist).

Charakteristika
Die unter dem Begriff NoSQL zusammengefassten Systeme berücksichtigen meistens einige der nachfolgenden Punkte:

- "Eventuell konsistent" (siehe BASE unten)
- Horizontal skalierbar auf herkömmlicher Hardware
- Kaum bis keine Schema-Restriktionen
- Nicht-relatonal
- Verteilt
- Einfache Datenreplikation
- Einfache Anfragesprache (API)
- Open Source

Prinzipien
Bisher hielten die RDBMS am Prinzip fest, stets konsistent zu sein. Damit stiessen Anwendungen, die auf Web 2.0 basierten und grosse Datenmengen mit vielen Transaktionen zu verarbeiten hatten, an ihre Grenzen. Eric Brewer zeigte an einem Symposium im Jahre 2000 in verständlicher Weise auf, dass es nur möglich ist, zwei der drei folgenden Prinzipien aufrecht zu halten:

1. Consistency: Die Daten sind in jeder Kopie enthalten und auf jedem Server gleich.
2. Availability: Die Daten müssen immer verfügbar sein.
3. Partition Tolerance: Eine Datenbank mit auf mehreren Servern verteilten Daten funktioniert auch wenn Teile des Netzwerks oder ein Server ausfallen.

Brewer bezeichnete dies das „CAP-Theorem". Da für Internetfirmen die Verfügbarkeit oberste Priorität hatte, blieb nur die Option, Zugeständnisse bei der Konsistenz zu machen. "Eventuell konsistent" zu sein wurde damit ein eigenes, neues Paradigma der NoSQL-Datenbanken. Dieses Credo der permanenten Verfügbarkeit mit bewusst in Kauf genommener vorübergehender Inkonsistenz wird auch mit der Abkürzung BASE, bezeichnet, d.h.

- Basically Available: permanente Verfügbarkeit.
- Soft State: Konsistenz ist nicht ein Dauerzustand.
- Eventual Consistency: Daten sind manchmal konsistent.

Dies im Gegensatz zum ACID-Prinzip, das von RDBMS her bekannt ist:

- Atomicity: Transaktionen werden ganz durchgeführt – oder gar nicht.
- Consistency: Die Daten sind immer konsistent.
- Isolation: Jeder Datenbankbenutzer sieht nur "seine" eigenen Daten.
- Durability: Daten gehen auch bei Systemcrashes nicht verloren.

Implementierungen
Heute bekannte Implementierungen sind im proprietären Bereich Google BigTable, Microsoft „Azure Table Storage" und Amazon Dynamo. Alle anderen oben erwähnten Produkte sind Open Source Software. Eine Einführung gibt [1] und eine gute Übersicht ist in [2] zu finden.

Ausblick
NoSQL ist noch relativ jung, so dass die Kategorisierung noch nicht gefestigt ist. Je nachdem eignet sich die eine oder andere Kategorie mehr oder weniger für ein bestimmtes Anwendungsgebiet oder für eine Datenstrukturart. Viele RDBMS-Hersteller arbeiten derzeit daran, ebenfalls Map/Reduce-Methoden zu integrieren. Geodaten kommen oft in grossen Mengen vor. Für die CouchDB und die effiziente Abfrage von Geodaten gibt es z.B. bereits eine Erweiterung namens GeoCouch. Einige NoSQL-Datenbanksysteme scheinen sich darum auch im Geo-Bereich als Alternative anzubieten.

Quellenangaben

[1] S. Edlich, A. Friedland, J. Hampe, B. Brauer, "NoSQL; Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken", 2010, Carl Hanser Verlag München, ISBN 978-3-446- 42355-8.
[2] http://nosql-database.org (aufgerufen am 15.3.2011)

Weiterführende Quellen:

- www.pgcon.org/2010/schedule/events/219.en.html (aufgerufen am 15.3.2011)
- www.odbms.org/experts.aspx (aufgerufen am 15.3.2011)
- www.guido-muehlwitz.de/2010/03/brave-new-world-was-ist-eigentlich-nosql/ (aufgerufen am 15.3.2011)
- www.heise.de/developer/artikel/CouchDB-angesagter-Vertreter-der-NoSQL-Datenbanken-929070.html (aufgerufen am 15.3.2011)

Fachgruppe GIS-Technologie
technologie@sogi.ch
Stefan Keller und Hans-Jörg Stark