

VSDB

Datenbanksynchronisation mit Couchbase und Couchbase Lite

Ausarbeitung

Alexander Rieppel

23. Mai 2014

5AHITT

Inhaltsverzeichnis

1	Einführung	3
1.1	Eigentliche Umsetzung im Diplomprojekt	3
1.2	Datenbanksynchronisation und Ansätze	3
1.2.1	Synchronisationsprobleme	4
1.2.2	Alternativen zur Synchronisation	5
1.3	Möglichkeiten	5
2	Couchbase Lite	6
2.1	Synchronisation mit Couchbase Sync Gateway	7
2.2	Vorteile dieser Umsetzung	7

1 Einführung

1.1 Eigentliche Umsetzung im Diplomprojekt

Ziel des Diplomprojektes war es eine Applikation für ein Tablet zu entwickeln, dass die Mitarbeiter der Lebensmittelversuchsanstalt bei der Probennahme im Supermarkt unterstützen soll. Insgesamt umfasst das Diplomprojekt eine Android Applikation und eine Server-Applikation. Bei der Probennahme werden Lebensmittel eingekauft, die am Tablet entsprechend in eine Datenbank eingetragen und später in der LVA für die weitere Bearbeitung bereitgestellt werden. Die Daten werden anschließend in die interne Datenbank in der LVA synchronisiert und für die Verarbeitung der Daten durch das Laborteam bereitgestellt. Auf weitere Vorgänge innerhalb der Firma hat allerdings die Tablet-Applikation keinen Einfluss mehr. Deshalb wird hier lediglich auf den Teil der Speicherung in die Datenbank und vor allem die Synchronisation mit der LVA-Datenbank eingegangen.

Innerhalb der Tablet Applikation wird eine SQLite Datenbank verwendet, da diese nicht sonderlich viele Ressourcen des Tablets benötigt und so eine einfache und angenehme Verwendung durch den Benutzer ermöglicht. Die Datenbank erlaubt zudem nicht allzu viele Datenbanken und im Vergleich zur klassischen MySQL Lösung nur wenige Nutzer. Auf dem firmeneigenen Server läuft eine klassische Mysql Instanz. Der Auftraggeber hatte zwar die Wahl offen gelassen welche Datenbanklösung bevorzugt wird, allerdings gleichzeitig auch bekanntgegeben, dass bereits eine Lizenz für Mysql vorhanden ist. Weitere in Betracht gezogene Datenbanksysteme waren PostgreSQL und ebenfalls auch für das Serverprogramm SQLite.

Die Umsetzung sah so aus, dass Daten vom Tablet zunächst an die Server-Applikation über Java-Streams versandt und anschließend von der Server-Applikation entsprechend in die Datenbank eingepflegt werden. Während Daten von der Datenbank auf das Tablet synchronisiert sind, dürfen diese auch von niemandem bearbeitet werden. So wird Konsistenz innerhalb des Systems gewährleistet.[RBDT]

1.2 Datenbanksynchronisation und Ansätze

Die sichere Synchronisation von Daten in verteilten Systemen ist ein wichtiges Anliegen, da der Prozess in erster Linie Konsistenz der Daten gewährleisten muss. Zu diesem Thema gibt es verschiedene Ansätze auf die hier näher eingegangen wird.

1.2.1 Synchronisationsprobleme

Der Kernpunkt ist, dass ein zentrales Datenbanksystem und ein oder mehrere kleinere Systeme existieren. Während die zentrale Datenbank sämtliche Daten des Systems beinhaltet, verfügen die kleineren Datenbanken nur über einen Bruchteil dieser Daten. Diese Bruchteile können von den einzelnen Geräten natürlich jederzeit geändert werden. Die Aufgabe besteht nun, dass all diese Bruchteile von Daten, wieder zur Hauptdatenbank synchronisiert werden ohne, dass größere Konflikte entstehen.

Zusätzlich zu den bereits geschilderten Punkten, werden noch wichtige Nebenpunkte betrachtet.

Die Verbindung der verteilten Datenbanken zum Hauptsystem muss nicht immer vorhanden sein. Das zentrale System muss wissen wann eine Verbindung besteht.

Die verteilten Datenbanken besitzen nur einen Teil der in der Hauptdatenbank gespeicherten Daten, wobei allerdings keiner der Datenbestände mit einem anderen der verteilten Datenbanken überlappt. Nur die zentrale Datenbank besitzt den selben Datenbestand wie eine der verteilten Datenbanken.

Ein Problem stellt hierbei ein Fall dar, wenn eine der verteilten Datenbanken vorübergehend offline geht und währenddessen Daten in der Hauptdatenbank geändert werden. Eine Methode um dies zu beheben wäre das simple überschreiben der Daten auf der entsprechenden Datenbank, falls einem der beiden Datenbanken eine höhere Priorität zugesprochen wird. Wenn zum Beispiel die verteilte Datenbank eine höhere Priorität besitzt als die zentrale Datenbank, kann die verteilte Datenbank die Daten der Hauptdatenbank einfach überschreiben.

Die Synchronisation der einzelnen Daten findet stets nur zwischen einer verteilten Datenbank und der zentralen Datenbank statt und keinesfalls unter zwei verteilten Datenbanken. Zusätzlich sind die Systemzeiten der einzelnen verteilten Datenbanken nicht synchronisiert. In einem zusammengefasst kommt man zu folgenden Punkten:

- Die Daten sollten konsistent bleiben, auch wenn die Synchronisation einmal fehlschlägt oder keine Verbindung besteht.
- Konfliktbehandlung hängt von den einzelnen Tabellen ab
- Verteilte Datenbanken werden hier nur mit der zentralen Datenbank synchronisiert und nicht untereinander
- Datenbankzeiten werden nicht synchronisiert

1.2.2 Alternativen zur Synchronisation

1.3 Möglichkeiten

Wie bereits erwähnt ist Datenbanksynchronisation in verschiedensten Bereichen sinnvoll. Anschließend ein Überblick von bestehenden Lösungen um gängige Datenbanken zu Synchronisieren:

- MySQL und SQLite

2 Couchbase Lite

Couchbase Lite ist eine leichtgewichtige, dokumenten-orientierte und leicht synchronisierbare Datenbank welche speziell für den Einsatz in mobilen Anwendungen und Geräten geeignet ist.

Leichtgewichtig bedeutet:

- Die Datenbank Engine ist eine in der Applikation gebundene Bibliothek und kein separater Serverprozess.
- Sie besitzt sehr klein gehaltenen Code damit Apps die auf die Schnittstelle zurückgreifen rasch heruntergeladen werden können.
- Garantiert eine kurze Startzeit da mobile Geräte meist geringere CPU-Leistung haben als PCs.
- Mobile Datensätze sind zwar relativ klein, allerdings können manche Dokumente große multimediale Anhänge haben, weswegen ein geringe Speicherauslastung von Nöten ist.
- Bietet auch eine gute Performance, wobei diese zu einem großen Teil von der implementierten Applikation abhängt.

Dokumenten-orientiert bedeutet:

- Speichert Einträge im flexiblen JSON-Format, weshalb keine vordefinierten Schemata benötigt werden.
- Dokumente können eine frei wählbare Größe von Binary-Anhängen besitzen, wie z.B. multimediale Inhalte.
- Das Datenformat der Applikation kann sich über die Zeit weiterentwickeln, ohne dass die Datenbank geändert werden muss.
- MapReduce Indizierung bietet eine schnelle Datensatzabfrage, ohne dass spezielle Query-Languages verwendet werden müssen.

Leicht synchronisierbar bedeutet:

- Zwei Kopien einer Datenbank können problemlos über einen Replikations-Algorithmus synchronisiert werden.

- Die Synchronisation kann On-Demand oder fortlaufend stattfinden.
- Geräte können auch nur Teilmengen einer riesigen Datenbasis eines Remote-Servers synchronisieren.
- Die Sync-Engine erlaubt auch das Synchronisieren über unbeständige und unzuverlässige Netzwerkverbindungen.
- Konflikte können einfach über einen Merge-Algorithmus, gefunden und behoben werden.
- Revisions-Bäume erlauben auch komplexe Replikations-Topologien, wie z.B. Server-to-Server (für mehrere Datenzentren) und Peer-to-Peer, ohne Datenverlust oder Konflikte befürchten zu müssen.

[Couda]

2.1 Synchronisation mit Couchbase Sync Gateway

[Couda]

2.2 Vorteile dieser Umsetzung

Literaturverzeichnis

- [Coua] COUCHBASE: *Couchbase Lite* - <http://docs.couchbase.com/couchbase-lite/cbl-concepts/>
- [Coub] COUCHBASE: *Couchbase Sync Gateway* - <http://docs.couchbase.com/sync-gateway/>
- [RBDT] RIEPPEL, A. ; BACKHAUSEN, D. ; DIMITRIJEVIC, D. ; TRAXLER, T.: *Diplomarbeit FI@D*