VSDB

Replikation

Protokoll

Alexander Rieppel Dominik Backhausen

30. Januar 2014

5AHITT

Inhaltsverzeichnis

1 Aufgabenstellung

Eine Handelsgesellschaft, die mehrere Filialen hat, betreibt eine Online-Plattform für den Verkauf der Produkte. Der Webshop wird mit Hilfe einer Datenbank betrieben. Bei dem Verkauf der Produkte werden Rechnungen in Form von PDF-Dokumenten erzeugt.

Aufgabenstellung:
Die Daten (Datenbank, Rechnungen) sollen stets auf die Filialrechner repliziert werden, damit die Sachbearbeiter vor Ort diese einsehen und bearbeiten können.

- Entwickle ein vereinfachtes Datenbankmodell für den Webshop
- Wähle ein Konsistenzprotokoll Deiner Wahl (siehe Theorie bzw. Tanenbaum)
- Implementiere einen Replikationsmanager in Java (JDBC, Sockets, o.ä. ...) für Datenbank und Rechnungen
- alle Transaktionen im Zuge der Replikation sollen protokolliert werden (zum Beispiel mit Log4J)

Beispiel fuer Log-Eintrag:
Replikation Rechnungen München -> Berlin OKAY
Replikation DB MÜnchen -> Berlin FEHLGESCHLAGEN

Problemstellungen:

- Wie oft wird repliziert?
- Wie erfolgt der Aufruf des Replikationsmanager bzw. läuft der Replikationsmanager stets im Hintergrund?
- Was passiert im Fehlerfall?
- Welche Probleme können auftreten? (Dateien mit gleichen Namen, Dateien mit gleichen Namen und unterschiedlicher Größe, Datensatz mit gleichem Schlüssel)

Meilensteine (16Pkt):

- Erstelle ein Replikationskonzept für diese Handelsgesellschaft (4 Punkte)

- Implementiere dieses Konzept für zwei Rechner (6 Punkte) mind. 10 Datensätze pro Tabelle, mind. 10 Rechnungen
- Implementierung Logging (2 Punkte)
- Dokumentiere drei Fehler-/Problemfälle und entsprechende Lösungvorschläge (4 Punkte)

2 Designüberlegung

2.1 Allgemein

Es soll einen oder mehrere Clients geben die sich zuerst mit dem Server verbinden müssen um die aktuellen Daten zu erhalten. Als nächstes können die Clients sich mit dem Server synchronisieren, wobei das Verfahren im Kapitel Konsistenzmodell näher erläutert ist. Wenn ein Client ein File in seinem Dateisystem aktualisiert, kann er dieses File synchronisieren. Es wird danach ein Abgleich der Änderungsdaten gemacht und die entsprechenden Files aktualisiert. Bei der Datenbank soll dies ähnlich funktionieren.

2.2 Konsistenzmodell

Als Konsistenzmodell wird das Monotone Schreiben gewählt, da es nach unserer Ansicht für diese Aufgabenstellung sinnvoll ist, erst alle Schreibvorgänge des Servers beim Client auszuführen bevor dieser Client wieder etwas zum Server hochladen kann.

3 Arbeitsaufteilung

Name	Arbeitssegment	Time Estimated	Time Spent
Alexander Rieppel	Datenbankverbindung	1h	2h
Alexander Rieppel	Replikationsmanager	4h	3h
Dominik Backhausen	Verbindung der Clients	1h	0.5h
Dominik Backhausen	Replikationsmanager	4h	3.5h
Gesamt		10h	9h

4 Arbeitsdurchführung

Die tatsächliche Implementierung ist nach oben beschriebener Designüberlegung erfolgt mit der Ausnahme, dass die Synchronisation der Datenbank noch nicht funktioniert. Der Austausch der Files funktioniert problemlos. Das einzige Problem ist,dass der Erstkontakt noch nicht reibungslos funktioniert, da die Files nicht heruntergeladen werden vom Server.

5 Testbericht

Zum Start des Programms gibt es folgende Möglichkeiten:

- ds (Starten des Servers mit Default Einstellungen)
- dc (Starten des Clients mit Default Einstellungen)
- s <port> <pfad> (Starten des Servers mit angegebenen Argumenten)
- c <ip> <port> <pfad> (Starten des Clients mit angegebenen Argumenten)

Die Argumente für die Verbindung der Datenbank sind noch nicht vorhanden, da die Datenbankreplikation bislang nicht implementiert ist. Derzeit wird auch noch nicht automatisch synchronisiert, was ebenfalls in Planung war.

5.1 Fehlerszenarien

- Dateien mit gleichen Namen

  Wird insofern gehandhabt, dass die Datei simpel als gleiche Datei behandelt und überschrieben wird, wenn eine neuere Datei bei Client oder Server ankommt.

- Dateien mit gleichen Namen und unterschiedlicher Größe

  Siehe Punkt 1

- Datensatz mit gleichem Schlüssel

  Derzeit keine Datenbank Implementierung, allerdings darf immer nur ein eindeutiger Schlüssel vorhanden sein, demnach wird der ältere ebenfalls überschrieben.