

Datenbanken & Informationssysteme Übungen Teil 3

Verteilte Datenbanken

1. Unterschied horizontaler und vertikaler Zerlegung

Beschreiben Sie den Unterschied zwischen *horizontaler* und *vertikaler Zerlegung* in einer verteilten Datenbank. Durch welche relationalen Operatoren werden die beiden Typen der Zerlegung in der Regel realisiert: beschreiben Sie die Zerlegung und das Zusammenführen der Daten als relationale Operationen.

2. Horizontale Zerlegung einer Tabelle

Ein Unternehmen verwendet in drei Werken Teile. Die Teile mit den Nummern 1 bis 300 werden in Werk I verwendet, in Werk II werden die Teile mit den Nummern 301 bis 500, aber mit Ausnahme von 399 eingesetzt und in Werk III werden alle übrigen Teile verwendet.

Konzipieren Sie die horizontale Zerlegung der globalen Tabelle

Teile(TeileNr, Bez, LieferNr, Preis)

unter der Voraussetzung, dass in der Regel in der verteilten Datenbank im jeweiligen Werk auf die Teile zugegriffen wird, die dort verwendet werden.

Zeigen Sie außerdem, wie die globale Tabelle aus den Fragmenten entsteht.

3. Beispiel für Zerlegung

Die Relation

Angest(PersNr, Name, Gehalt, MgrNr, AbtNr, Anschrift)

soll so zerlegt werden, dass ein Fragment Namen und Anschrift, eines die Personalnummer des Vorgesetzten und die Abteilungsnummer enthält, schließlich soll das dritte Fragment Namen und Gehalt umfassen.

Wie kann man die globale Relation aus den Fragmenten definieren?

4. Schema einer verteilten Datenbank

Ein Softwarehaus möchte eine verteilte Datenbank für das Projektmanagement in ihren Geschäftsstellen Hamburg (HH), Darmstadt (DA) und München (M) einrichten. Die im Moment in der zentralisierten Datenbank verwendete Datenstruktur ist so aufgebaut:

Angest(PersNr, Name, Gehalt, AbtNr, Anschrift)

Abt(AbtNr, Name, MgrNr, BereichNr, RegionNr)

Projekt(ProjNr, Name, Kontraktvolumen, ProjMgrNr, AbtNr)

Leistung(PersNr, ProjNr, Stunden)

Bereich(BereichNr, Name)

Region(RegionNr, Name)

dabei sind

Bereiche: Consulting (SC), Entwicklung (SE) und Rechenzentrumsbetrieb (RZ)
Regionen: Hamburg (HH) mit den Bereichen SC, SE; Darmstadt (DA) mit dem Bereich SE und München mit den Bereichen SE und RZ.

Die Zentrale der Firma ist in München, wo auch die Personalabteilung (HR) ist. Üblicherweise werden folgende Informationen benötigt:

- Informationen über die Projekte in den Bereichen, wobei jedes Projekt mit Mitarbeiter aus der Region durchgeführt wird.
- Informationen der zentralen Personalabteilung über alle Mitarbeiter.
- Informationen der Abteilung über die Arbeit der Mitarbeiter an Projekten für die Rechnungsstellung.

Erarbeiten Sie einen Vorschlag für die Zerlegung und Replikation von Daten, die der geschilderten Situation gerecht wird. Begründen Sie Ihre Vorgehensweise:

- Erstellen Sie ein Entity-Relationship-Diagramm für die Datenstruktur.
- Erstellen Sie das Design für Verteilung und Replikation.
- Zeigen Sie wie die globalen Relationen aus den Fragmenten rekonstruiert werden können.
- Überlegen Sie Anfragestrategien für die oben skizzierten typischen Anfragen in der verteilten Datenbank

5. Semijoin

Betrachten Sie folgende Situation in einer verteilten Datenbank:

An Site 1 ist die Tabelle T gespeichert, die Zeilen mit den ids 1, 3, 5, 8, 14 enthält. Jede Zeile ist 1024 Bytes lang, 4 Bytes für die id und 1020 Bytes für die Nutzdaten.

An Site 2 ist die Tabelle S gespeichert, deren Zeilen analog aufgebaut sind, und die ids 1, 2, 4, 7, 14, 20 haben.

An Site 1 soll der Join durchgeführt werden:

```
select * from T, S where T.id = S.id.
```

Spiele Sie die Schritte durch, die erforderlich sind, wenn man den Join mittels des *Semijoins* ermittelt. Zeigen Sie, welche (Teile von) Tabellen von Site 1 an Site 2 und umgekehrt übertragen werden müssen.

Wieviele Bytes müssen bei diesem Vorgehen über das Netz übertragen werden? Vergleichen Sie den Wert mit der Vorgehensweise, dass die komplette Tabelle S an Site 1 übertragen wird und dort der Join durchgeführt wird.

6. Bloomjoin aka Bitvektor-Join

Gehen Sie wieder von der Situation der vorherigen Aufgabe aus, spielen Sie nun den *Bloomjoin* durch, bei dem Sie die Hashfunktion $h(x) = x \bmod 7$ verwenden. Berechnen Sie wieder, wieviele Bytes über das Netz übertragen werden müssen.

7. Vorgehen in einer verteilten Datenbank I

Betrachten Sie folgende Datenbank:

```
Mitarbeiter( MId, MName, WId, Gehalt)
Werk( WId, Sitz)
```

Setzen Sie voraus, dass die Tabelle *Mitarbeiter* horizontal fragmentiert ist nach der *WId* am jeweiligen Sitz des Werks.

Beschreiben Sie Strategien für folgende Statements:

- `update Mitarbeiter set Gehalt = Gehalt * 1.05`
- `update Mitarbeiter set WId = 12 where MId = 2314`
- `insert into Mitarbeiter values(4567, 'Schneider', 12, 5000)`

8. Vorgehen in einer verteilten Datenbank II

Verwenden Sie die Datenbank und ihre Verteilung aus der vorherigen Aufgabe. Setzen Sie zusätzlich voraus, dass folgende Replikationsstrategie eingeführt ist: Jedes Fragment hat 2 Replika: eines in der Zentrale in Frankfurt, das andere am Sitz des jeweiligen Werkes.

Beschreiben Sie gute Suchstrategien für folgende Abfragen, wenn Sie sich in Hamburg befinden:

- Alle Angestellten in Bremen
- Durchschnittsgehalt aller Beschäftigten

9. Vorgehen in einer verteilten Datenbank III

Wir wandeln unsere Datenbank etwas ab:

```
Mitarbeiter( MId, MName, AId, Gehalt, ...)
Abteilung( AId, MgrId, Sitz, Status, ...)
(MgrId ist die MId des Managers der Abteilung.)
```

Die Abfrage, die uns interessiert, lautet:

```
select * from Mitarbeiter M, Abteilung A
where M.MId = A.MgrId and A.Status > 50
```

und wir wissen, dass etwa 1 Prozent der Mitarbeiter Abteilungsleiter sind und etwa die Hälfte der Abteilungen den gewünschten Status hat.

Setzen Sie voraus, dass die Tabelle der Mitarbeiter in Frankfurt, die der Abteilungen in München gespeichert ist. Sie selbst sind in Hamburg.

Die Tabelle *Mitarbeiter* hat 100.000 Zeilen à 4 KBytes, die Tabelle *Abteilung* 1.000 Zeilen à 4 KBytes. Die Schlüssel sind jeweils 4 Bytes lang. Für die Übertragungsrate im Netz nehmen wir 100 KBytes pro Sekunde an.

Diskutieren Sie folgende Strategien und machen Sie Aussagen über die Übertragungskosten:

- Transferieren Sie beide Tabellen nach Hamburg und berechnen den Join dort.
- Transferieren Sie *Abteilung* nach Frankfurt und berechnen den Join dort, übermitteln Sie das Ergebnis nach Hamburg.
- Berechnen Sie den Join mit der Strategie des *Semijoin* in München und übermitteln Sie das Ergebnis nach Hamburg.

10. Arten der Replikation

Erläutern Sie den Unterschied zwischen *synchroner* und *asynchroner* Replikation in verteilten Datenbanken. Nennen Sie Gründe, weshalb bei den heute verfügbaren Datenbank-Systemen in der Regel die asynchrone Replikation verwendet wird?

11. 2PC Absturz des Teilnehmers

Im 2PC sieht der Ablauf (bei globalem Commit) für Teilnehmer und Koordinator (zusammengefasst) so aus:

Koordinator:

```
schreibe [T,prepare] in Log
sende <prepare T> an alle Teilnehmer
nach Erhalt der (positiven) Antwort: schreibe [T,commit] in Log
sende <commit T> an alle Teilnehmer
nach Bestätigung: schreibe [T,eot] in Log
```

Teilnehmer:

```
nach <prepare T>: schreibe [T,ready] in Log
sende <ready T> an Koordinator
nach <commit T>: schreibe [T,commit] in Log
sende <ack T> an Koordinator
```

Erläutern Sie, wie das 2PC-Protokoll verläuft, wenn der Teilnehmer in folgender Situation abstürzt:

- vor Erhalt der Nachricht <prepare T>
- nach Erhalt von <prepare T>, aber vor der Antwort an den Koordinator
- nach Antwort <ready T> an Koordinator
- nach der Nachricht <ack T> an Koordinator

12. 2PC Absturz des Koordinators

Erläutern Sie, wie das 2PC-Protokoll verläuft, wenn der Koordinator in folgender Situation abstürzt:

- vor dem Schreiben von [T,prepare]
- nach dem Senden von <prepare T> an alle Teilnehmer
- nach Erhalt der Antwort <ready T> von einem Teil der Teilnehmer
- nach dem Schreiben von [T,commit]
- nach dem Senden von <commit T> an einen Teil der Teilnehmer
- nach dem Senden von <commit T> an alle Teilnehmer