Prüfung: Datenbanken in Rechnernetzen (Kurs 1666)
Prüfer: Prof. Dr. Schlageter
Datum: 9. Jan 2012
Note: 1,7

Warum verteilt man Datenbanken überhaupt?
- Verfügbarkeit & Durchsatz steigern, Skalierbarkeit verbessern
(vom Thema Verfügbarkeit ging es zur Replikation)

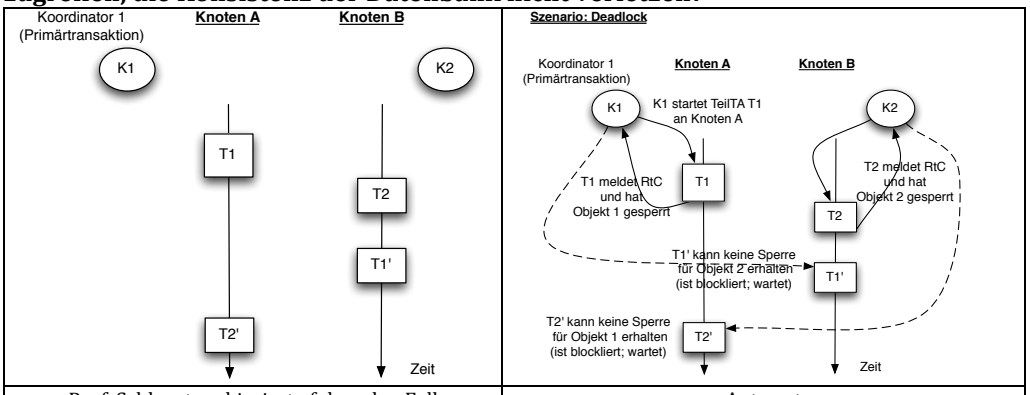
Wieso kann man sich trotz geringerer Übertragungskosten, dazu entscheiden eine Anfrage durch einen Knoten <u>nicht</u> ausführen zu lassen?
- wenn der Knoten überlastet ist → Stichwort: Load Balancing

Wie funktioniert das Primary Copy Verfahren?
- beim Nachteil „fehlendes konsistentes Lesen", wurde gefragt, ob es denn immer nötig ist, Daten konsistent lesen zu müssen → Nein (stark vom Anwendungsfall abhängig; Bsp.: Suchmaschine Google bzw. Wetterdienste)
(als ich ROWA erwähnte, wollte er wissen, wieso es praktisch nicht einsetzbar ist: schlechte Verfügbarkeit)

Welche Informationen benötigt der Query Prozessor, um entscheiden zu können, welche Anfrage gegen welchen Knoten gestellt wird?
- Informationen aus dem globalen Katalog (welche Partitionen sind an welchen Knoten allokiert)
(dann wurde zum 2PC übergeleitet)

Was bedeutet globale Serialisierbarkeit mit dem 2PC?
- lokale Serialisierbarkeit erklärt (Konfliktserialisierbarkeit) und auf globale Serialisierbarkeit eingegangen (es muss eine kompatible Schedule geben, in der sich alle lokalen äquivalenten Schedules einfügen lassen)
- erklärt, dass überlappende Ausführung von Aktionen den Durchsatz steigert
(bei der Frage wollte Prof. Schlageter zu den Sperrverfahren und Deadlocks überleiten; 2PC und simultane Updates sollten ich explizit nicht erläutern; das hat mich etwas aus dem Konzept gebracht und war wahrscheinlich auch der Grund für die folgende Skizze)

Wie wird mit 2PC im verteilten Fall sichergestellt, das Transaktionen die auf gleiche Objekte zugreifen, die Konsistenz der Datenbank nicht verletzen?



Prof. Schlageter skizzierte folgenden Fall	Antwort

- die Schedule ist nicht serialisierbar (Knoten A: T1, T2 und Knoten B: T2, T1) ➔ Folge: Deadlock

Wie können Deadlocks erkannt werden?
- Timeout, Deadlockerkennung (zentral, dezentral), Protokolle zur Vermeidung (kaum relevant)

Wie funktioniert die DDD (Distributed Deadlock Detection) grob?

Wieso nutzt man Semi-Joins?
- Fokus auf verteilten Fall und die Übertragungskosten (Join Optimierung)

Die Prüfung und die Note waren sehr fair. Wenn man die Antwort nicht parat hat, versucht Prof. Schlageter einen durch weitere Erläuterungen (oder eben auch eine Skizze) zum Ziel und zur richtigen Antwort zu führen. Die Verknüpfung 2PC und Sperrverfahren sollte sicher sitzen.

Prüfungsprotokoll Datenbanken in Rechnernetzen

Jahr: 2011
Prüfer: Prof. Schlageter
Note: 2.0

Bei meiner Prüfungsvorbereitung habe ich mich sehr stark auf die Prüfungsprotokolle verlassen. Da diese aber doch ein wenig „betagt“ sind (2003 ist das aktuellste), hatte ich ein recht unsicheres Gefühl. Um euch dieses Gefühl zu ersparen, schreibe ich dieses Prüfungsprotokoll.

Vorneweg: Die „alten“ Prüfungsprotokolle sind weiterhin zur Vorbereitung uneingeschränkt zu empfehlen.
Prof. Schlageter legt (immer noch) viel Wert darauf, dass die Zusammenhänge verstanden werden. Auswendiggelerntes interessiert ihn nicht in dem Maße, wie tiefes, vernetztes Wissen.

Trotzdem gibt es einige Unterschiede zu vorherigen Protokollen:

- Prof. Schlageter erschien pünktlich ;-)
- Ich hatte den Eindruck, dass viele Standardfragen nicht direkt, sondern verpackt in Form von Beispielen gestellt wurden – hier ist es also sehr wichtig logisch und gelassen zu denken und Zusammenhänge herauszuarbeiten (ist mir nicht immer gelungen)
- beim Thema 2PC-Protokoll kamen sehr zielgerichtete Fragen, den allgemeinen Ablauf durfte ich gar nicht erst erklären
- bei der Deadlock-Erkennung musste ich ihm den DDD-Algorithmus grob umreißen
- leider durfte ich auch nichts zu den Joins sagen…

Nach 15 Minuten war die Prüfung zu Ende.

Prof. Schlageter ist meiner Meinung nach als Prüfer weiterhin uneingeschränkt zu empfehlen.

Gedächtnisprotokoll: Master of Computer Science (wobei für den Prüfer egal ist in welchem Studiengang man eingeschrieben ist)

Kurs: 01666 Datenbanken im Rechnernetze (SS2005)
Prüfer: Prof. Dr. Schlageter
Termin: Oktober 2006
Dauer: ca. 15 Min
Note: 1.0

Allgemeines:
1. Die existierenden Gedächtnisprotokolle sind für die Prüfungsvorbereitung gut geeignet. Vor allem sind die Schwerpunkte eindeutig bestimmt.
2. Bei der Vorbereitung zur Prüfung habe ich die vorhandenen Prüfungsprotokolle alle genau angesehen und mich nach denen gerichtet. Viele Autoren der Protokolle schreiben, dass die Zusammenhänge gut verstanden werden sollten. Ich habe mich oft gefragt, was sie damit genau meinen. Ich habe mich deshalb entschieden, nach der Prüfung diese Erfahrung weiterzugeben. Ich hoffe, dass es mir einigermaßen gelingt, die Art wie Prof. Schlageter prüft und was unter Zusammenhänge zu verstehen ist, darzustellen.

Also: Während der Prüfung hatte ich nicht die Chance gehabt, kleine Vorträge, die ich zu Strategien, Algorithmen o.ä. vorbereitet habe, vorzutragen. Nur die Teile davon konnte ich zum Teil ins Gespräch einbringen. Prof. Schlageter stellt nicht Fragen der Art: Was ist das oder jedes? Sondern, nennt ein Themengebiet wie z.B. Prozesskommunikation und dann fragt quer durch das Thema. Er stellt viele Fragen und als Antwort erwartet er kurze, prägnante Antworten. War die Antwort gut, so hört man von ihm ‚Ja, genau so..' oder ‚Exakt', ‚genau das ist es' usw. Lange Reden sind bei ihm bestimmt nicht gefragt.

Prof. Schlageter ist sehr lebhaft und aufgeschlossen. Er hat nicht das Ansehen eines ernsten und im Gespräch trockenen Professors. Man füllt sich während der Unterhaltung mit ihm frei.

Es ist schwer den Ablauf und die gestellten Fragen nachträglich zu rekonstruieren. Ich habe die gestellten Fragen in etwa so gestellt bekommen. Sie sind auch nicht vollständig, da ich mir sie nicht alle merken konnte. Meine Prüfung ist 10 Min zu früh angefangen!!! ☺

Für die, die noch nicht daran gedacht haben, einen Gedächtnisprotokoll für die anderen zu schreiben:
Vorab möchte ich noch dich darauf aufmerksam machen, dass auch du dein Protokoll veröffentlichen solltest. Einige Studenten, die ich kenne, geben ihre Erfahrungen nicht weiter. Sie studieren nach dem Motto: Hauptsache ich habe die Prüfung selbst bestanden. Wer so denkt, hat nicht das Recht die Protokolle anderer Studierenden, die sich die Arbeit mit der Protokollerstellung machen, zu lesen und zur Vorbereitung von Prüfungen zu nehmen.
Kooperation ist alles! Denk nicht nur an dich, sondern auch an die anderen, die sich nach dir auf die Prüfung vorbereiten werden und für deine Bemühungen dankbar sind, wie du jetzt für diese Protokolle bist!!! Gemeint sind vor allem die Studenten, die bei der Prüfung gut abgeschnitten haben.

Viel Glück bei der Prüfung!

1. **Warum verteilt man Datenbanken?**
   ⇒ Je nachdem wie verteilte Datenbank entstanden ist können es verschiedene Gründe geben. Wenn eine vert. DB von Anfang an als verteilt geplant wurde, dann sind es Ziele: Minimierung der Antworzeiten, Erhöhung der Verfügbarkeit des Gesamtsystems, Skallierbarkeit (leichte Erweiterbarkeit durch Hinzufügen zusätzlicher Knoten), Möglichkeit der günstigen Hardwarelösungen. Dann wurde ich unterbrochen. Zur Dezentralisierung und Integration habe ich bei der Frage nichts sagen können.
2. **Welche Probleme entstehen bei der Integration?**
   ⇒ Bei der Integration gehen es darum, Produktions- oder andere Kosten zu senken. Deshalb bringt man autonome Systeme zur Kooperation. Die einzelnen Datenbanken können sich in der Datenbankmodel, Datenbankhersteller und lokalen Schemata unterscheiden. Bei lokalen Schematas können Namens- und Datenstrukturkonflikte auftreten. Bei Datenstrukturkonflikten können es z.B. Typ-, Beziehungs- und Wertekonflikte auftreten.
3. **Was löst man sie?**
   ⇒ Die Heterogenität der einzelnen Datenbanken und Datenverteilung wird mit Hilfe der globalen Schemaarchitektur verborgen. Bei prä-integrierten Datenbanken wird die Integration von oben nach unter durchgeführt. Bei post-integrierten Datenbanken dagegen von unten nach oben. Die Heterogenität der lokalen Schemata und die Vereinheitlichung zu einem Datenmodell wird bei den post-integrierten Datenbanken durch die zusätzliche die so genannte lokale Repräsentationsschicht verborgen.
4. **Was macht genau die lok. Repräsentationsschicht?**
   ⇒ Verbirgt die Heterogenität der lokalen Schemata bei Namens- und Strukturkonflikten und bringt zu einem einheitlichen relationalen Datenbankmodell.
5. **Wozu Redundanz?**
   ⇒ Erhöht die Verfügbarkeit der verteilten Datenbank. Das andere (Steigerung der Anfrageneffizienz) habe nicht nicht geschafft zu sagen, da gleich die nächste Frage gestellt wurde.
6. **Was kann man redundanz Spechern?**
   ⇒ Globale Relationen oder Teilrelationen
7. **Welches Problem entsteht dabei?**
   ⇒ Problematisch ist die Konsistenzhaltung von redundant gespeicherten Daten. Die Abhilfe stellen Replikationsverfahren dar. In der Praxis werden Kopien-Update-Strategien eingesetzt. Sie kann man in zwei Arte unterscheiden: Ein Verfahren mit vorbestimmter Kopie und mehrere Abstimmungsverfahren. Wobei die Abstimmungsverfahren sich in Wesentlichen dadurch unterscheiden, wie viele Kopien bei der Durchführung von Transaktionen benötigt werden und ob alle Kopien gleichbehandelt oder unterschiedlich behandelt werden. Es gibt also Abstimmungsverfahren mit statischem und dynamischen Quorum als auch mit gewichteten und ungewichteten Stimmen.
   ⇒ Das Verfahren mit vorbestimmten Kopie bräuchte ich nicht zu erläutern
8. **Wie werden bei Replikation Updates durchgeführt?**
   ⇒ Beim Verfahren mit vorbestimmten Kopie werden zuerst Änderungen an der Hauptkopie durchgeführt, erst dann werden die restliche Kopien aktualisiert. Bei Abstimmungsverfahren wird nur ein Teil Kopien während der Transaktion aktualisiert. Die restlichen Kopien werden asynchron aktualisiert. (Bei dieser Frage bin ich mir nicht ganz sicher)

9. Wie führt man bei verteilten Datenbanken Transaktionen aus?

⇒ Mit Hilfe des 2-PC-Protokolls

10. Man führt auf den einzelnen Knoten die Teiltransaktionen aus, die jeweils serialisierbar sind. Warum reicht es nicht aus einfach Commits abzusetzen?

⇒ Die globalen Transaktionen müssen serialisierbar sein.

11. Was kann dann passieren?

⇒ Es können globale Deadlocks auftreten.

12. Was passiert mit den glob. Transaktionen wenn so ein Fall aufgetreten ist?

⇒ Sie bleiben hängen. Die Ausführung kann nicht fortgesetzt werden.

13. Was ist dann zu tun?

⇒ Es wird eine Suche nach Verklemmungen durchgeführt. Es gibt zentrale und dezentrale Suche nach Verklemmungen.

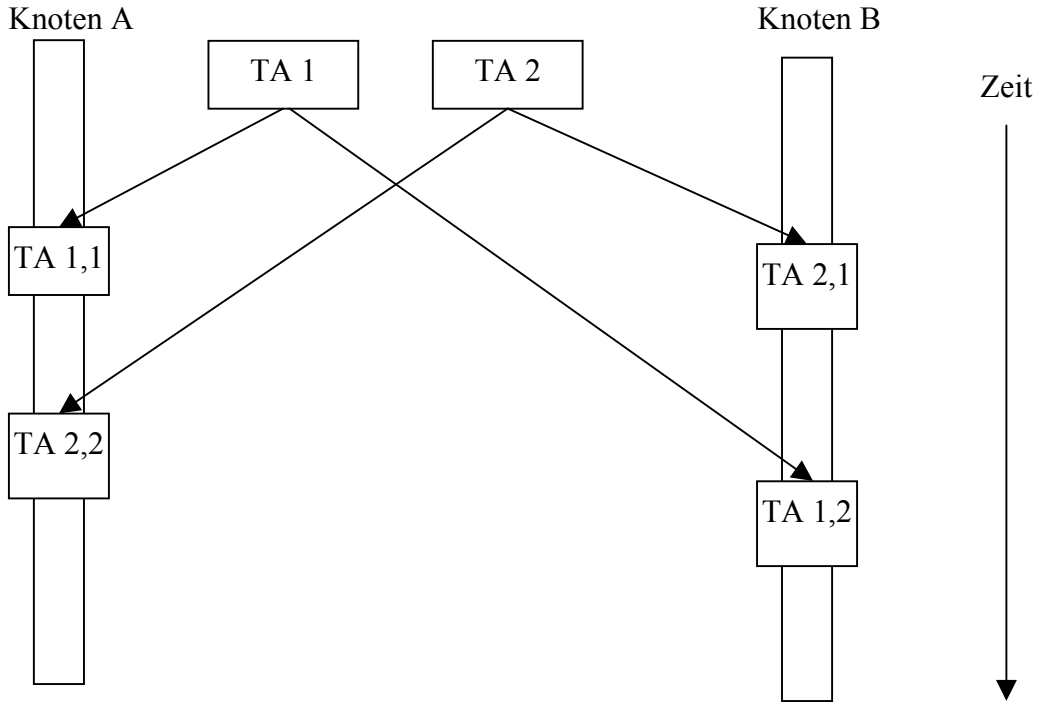
⇒ Ich bräuchte die Einzelheiten nicht nennen**.**

Gedächnisprotokoll Diplomprüfung Kurs 1666 „Datenbanken in Rechernetzen“
Prüfer: Prof. Schlageter
Termin: November 2003
Dauer: ca. 15 Minuten
Note: 1,3

Ich habe nur zwei Fragen aufgeführt, da alle anderen Fragen schon in den älteren Protokollen vorkommen. Besonders die 1.Frage stellt Prof. Schlageter – nach eigener Aussage - in letzter Zeit häufiger. Zum Thema Client-Server (KE 7) wurden mir keine Fragen gestellt.
Bei der Prüfungsvorbereitung empfehle ich darauf Gewicht zu legen, Fragen möglichst schnell und flüssig zu beantworten. Prof. Schlageter ist zwar ein fairer Prüfer, er möchte jedoch für die guten Noten - die er vergibt – etwas sehen.

Welche Entscheidungskriterien gibt es für die folgende Situation?
In einem vDBMS stehen für eine Anfrage zwei Knoten zur Verfügung.
Knoten A ist schneller als Knoten B. Die Verbindung zu beiden
Knoten ist gleich schnell und stabil.
1.Frage: Wieso könnte man sich doch für den langsameren Knoten entscheiden?
Antwort: Wenn die Last auf Knoten A zu groß ist (Stichwort: Load Balancing).

Professor Schlageter skizzierte die folgende Zeichnung, in der zwei Transaktion TA1 und TA2 die zwei Knoten mit den Teiltransaktionen TA1,1 und TA2,1 sperren. Die Blöcke mit den Bezeichnungen der Teiltransaktionen kennzeichnen nur den Start der Sperre, nicht deren Aufhebung!



2.Frage: An welchen Stellen gibt es in der Zeichnung Probleme (Stellen waren einzuzeichnen)?
Antwort: Ein Knoten kann nicht durch zwei verschiedene Transaktionen gleichzeitig gesperrt werden. Einzuzeichnende Stellen sind also der Punkt wo die Pfeilspitze von TA2 auf TA2,2 stößt und die von TA1 auf TA1,2 stößt.

Gedächtnisprotokoll: Diplomprüfung Vertiefende Kurse

Kurs : 1666 Datenbanken in Rechnernetzen (Version SS 99)
Prüfer : Prof. Dr. Schlageter
Termin : Juni 2001
Dauer : 15 Minuten

Themen:

1. Warum verteilt man Datenbanken ?

Die einzelnen Punkte mit kurzer Erklärung genannt: Dezentralisierung, Integration, günstigere Hardware, Redundanz.

Bei Integration gab es noch Rückfragen zum Thema Schema. Kurze Beschreibung der Schemaarchitektur im verteilten Fall. Danach Aufzeigen der Problematik mit unterschiedlichen Schemata.

2. Welches Problem entsteht durch redundante Speicherung ?

Engpaß kann durch Update entstehen.

Einzelne Updateverfahren genannt: ROWA und Primary Copy mit Erläuterung angegeben. Beim Primary Copy Verfahren kam noch die Nachfrage, ob man sicher sein kann beim Lesen immer eine aktuelle Version zu haben.

Ich nannte dann noch den Begriff der Abstimmverfahren. Hier wurde das Thema dann aber abgebrochen.

3. Wie ist beim verteilten Update die Serialisierbarkeit gewährleistet ?

Hinweis auf 2 PC-Protokoll und 2 PL-Protokoll gegeben. 2 PC-Protokoll erklärt. Wichtig war, daß der globale Commitzeitpunkt die Rheinfolge bei der Serialisierbarkeit bestimmt.

Allgemeiner Eindruck:

Professor Schlageter ist ein sehr netter und lockerer Prüfer. Auch bei mir begann die Prüfung etwas später. War dafür aber recht kurz. Die einzelnen Themen wurden meist gar nicht weiter vertieft. Sobald Professor Schlageter das Gefühl hatte, daß der Themenkomplex beherrscht wird, kam das nächste Thema dran.

Mündliche Diplomprüfung – Praktische Informatik

Prüfer : Prof. Dr. Schlageter
Datum : August 2000
Kurs : 01666 Datenbanken in Rechnernetzen

- Warum gibt es verteilte Datenbanken, wozu setzt man diese ein?
  ⇒ *Dezentralisierung*
  (Verfügbarkeit, Gesamtdurchsatz, Lastverteilung, Kostenoptimierung)
  ⇒ *Integration*
  (Single-System-Image, Konsistenzverbesserung)

- Was ist bei der Integration von Datenbanksystemen zu beachten?
  ⇒ *Namenskonflikte*
  ⇒ *Strukturkonflikte*
  ⇒ *Wertbereichskonflikte*
  (Thema wurde nicht weiter fortgeführt)

- Wie erreicht man eine höhere Verfügbarkeit?
  ⇒ *Redundanz*

- Welches Problem ergibt sich daraus?
  ⇒ *Konsistenzerhaltung*
  ⇒ *Abhilfe durch Kopien-Update-Strategien*

- Nennen von Kopien-Update-Strategien
  ⇒ *ROWA*
  (konsistentes Lesen, Updateproblematik erklärt)
  Zwischenfrage: Welche praktische Relevanz hat dieses Verfahren? Keine!
  ⇒ *Primary Copy*
  (Update nur auf eine Kopie, Lesen von jeder Kopie, aber inkonsistent)
  Zwischenfrage: Welche einzige Lesemöglichkeit bleibt also über? Das konsistente Lesen von der Primary Copy.
  (Thema wurde nicht weiter fortgeführt)

- Welche Besonderheit gibt es bei der Anfragebearbeitung in vDBMSen?
  ⇒ *neben der lokalen auch die globale Optimierung*
  ⇒ *Beachtung von Parallelität und Lastverteilung*
  ⇒ *Optimierungsaspekt: Antwortzeit (und damit Übertragungsdauer)*
  ⇒ *Ziel: Reduzierung der zu übertragenden Datenmengen*
  ⇒ *besonders kritische und kostenintensive Operation: Join*
  (Fragestellung diente als Überleitung zu den Join-Verfahren)

- Wie führt man den verteilten Join durch?
  ⇒ *Nested-Loop-Join*
  (nicht erklärt, nur erwähnt, daß er praktisch keine Relevanz hat)
  ⇒ *Sort-Merge-Join*
  (Join-Attribute sortiert oder indexiert)
  ⇒ *Semi-Join*
  (Reduzierung der zu übertragenden Relation; Skizzierung der Vorgehensweise; Hinweis auf die Problematik, daß der Aufwand gegebenenfalls größer als der Nutzen ist; besonders bei kleinen Relationen)

- Was ist eigentlich die " naive" verteilte Join-Verarbeitung?
  - ⇒ *Übertragung der kleineren Relation an die größere Relation und lokale Ausführung*
  - ⇒ *Bewertung: Verfahren ist nicht unüblich und praxisrelevant*

- Wie sind zum Beispiel bei redundanter Speicherung von Partitionen Updates durchzuführen? Wie sieht es mit Serialisierbarkeit aus?
  - ⇒ *2-Phasen-Commit-Protokoll*
  - ⇒ *Serialisierbarkeit definiert durch die globale Commit-Zeitpunkte*
  - ⇒ *Sperren nach dem globalen Commit freigeben, nicht nach dem RtC*
  - ⇒ *Teil-Transaktionen befolgen das 2-Phasen-Sperr-Protokoll (siehe Kurs 1665)*
  - ⇒ *der Zeitpunkt t_{UNLOCK} der Teiltransaktionen wird durch das globale Commit der Primärtransaktion bestimmt*
  - ⇒ *man kann zeigen, daß die globale Transaktion den gleichen Sperrverlauf (Graphik Kurs 1665) des 2-Phasen-Sperr-Protokolls befolgt wie die Teiltransaktionen*

- Was bedeutet RtC für die Teiltransaktion? Was ist in die Logfiles zu schreiben?
  - ⇒ *inhaltliche Bedeutung: Teiltransaktion kann Abort (Undo) sowie Commit (Redo) der globalen Transaktion begleiten*
  - ⇒ ***vor** dem RtC-Zeitstempel sind die Objektsperren sowie die Before- und / oder After-Images (je nach Recovery-Strategie) physisch ins Logfile zu schreiben*
  - ⇒ *die obigen Einträge wichtig für den Recoverymanager nach Anlauf eines abgestürzten Knotens*
  - ⇒ *zum Beispiel ist sicherzustellen, daß eine Teiltransaktion, die RtC erreicht hatte, auch nach Anlauf des Knotens wieder den RtC-Zustand erreicht*

- Client/Server - Was ist darunter zu verstehen?
  - ⇒ *Software-Architekturmodell*
  - ⇒ *Aufteilung einer Anwendung auf Client und Server*
  - ⇒ *Grundprinzip "Client - Server" erklärt*
  - ⇒ *typische Anwendungen im Zusammenhang mit Datenbanken: Client mit dem Anwendungsprogramm, Server mit der Datenbank*
  - ⇒ *Ziel hier: Reduzierung entfernter Datenbankzugriffe*

- Was ist der " dünnste" Client?
  - ⇒ *die verteilte Präsentation (zum Beispiel das X-Windows-System)*
  - ⇒ *den "Rest" der Verarbeitung übernimmt der Server*

<u>Anmerkung:</u>
Die Prüfung lief in lockerer Atmosphäre in Form eines Prüfungsgespräches ab. Hier entwickelten sich auch die meisten der oben angesprochenen Fragen. Herr Prof. Dr. Schlageter ist als Prüfer uneingeschränkt zu empfehlen.

Gedächtnisprotokoll mdl. Diplomprüfung Kurs 1666

am 22.9.99
Prüfer: Prof. Schlageter
Dauer: <15 min
Note: 1.0

- Welche Ziele verfolgt man mit verteilten Datenbanken?
  Dezentralisierung /Integration (jeweils Ziele erklärt)
- Was ist das Problem bei Replikation?
  Update-Stragtegien erklärt, Synchronisation der Updates für Primary Copy erklärt.
- Welche Aufgabe hat der query-processor?
  Globale Transaktionen in lokal ausführbare Teilanfragen umzusetzen. (Informationen über Partitionierung und Allokation nötig)
- Was kann hierbei optimiert werden?
  Ich wollte zunächst allgemeine Query-Optimierung erklären (Eliminierung überflüssiger Teilanfragen durch qualifizierte Relationen, etc.), Prof. Schlageter wollte aber hauptsächlich auf den verteilten Join hinaus:
- Was ist in diesem Zusammenhang die teuerste Operation?
  Der Join.
- Wie kann man ihn optimieren
  Zunächst naive Vorgehensweise erklärt (komplette Relationen verschicken/Nested Loop Join), dann Sort Merge und Semi Join erklärt.
- wie garantiert 2PC Serialisierbarkeit?
  Sperren werden nicht bei RtC, sondern erst beim globalen Commit freigegeben.
- Was bedeutet dies?
  Die äquivalente serielle Schedule wird genau durch den globalen Commit-Zeitpunkt bestimmt.

Eigentlich ist den zahlreichen Protokollen über Prüfungen bei Prof. Schlageter nichts hinzuzufügen. Die Prüfung fing wie immer mit ca 15 min Verspätung an ;-), war dann aber sehr locker und fair. Prof. Schlageter fragt nicht weiter in die Tiefe, sobald er erkennt, daß der Prüfling die Zusammenhänge begriffen hat, dies ist aber absolut essentiell!
Ich hatte mich (ausgehend von früheren Prüfungsprotokollen) sehr gut auf das 2PC-Protokoll vorbereitet, worüber wir dann aber nicht mehr als 2 Sätzen verloren, allerdings hätte prof. Schlageter sicherlich noch detaillierter nachgefragt, wenn ich die Frage zum äquivalenten seriellen Schedule nicht hätte beantworten können.
Insgesamt kann man wohl sagen, daß ein guter Überblick über den Kurstext (und auch weitergehende Zusammenhänge, die vielleicht nicht wörtlich im Skript stehen) wichtiger ist, als einzelne Algorithmen o.ä. zu kennen.

Gedächtnisprotokoll der Diplomprüfungen zu den Kursen

1665 Datenbanksysteme und
1666 Datenbanken in Rechnernetzen

bei Professor Schlageter
im Februar 1999

Note 1

Fragen :

1665 Datenbanksysteme
Welche Sperrverfahren gibt es.
Was sind hot -spots
Was passiert, wenn auf hot-spots X- Locks liegen
Sperrprotokoll beim Sperren mit Locks
2PL-Protokoll
Woran erkennt man, daß Transaktionen serialisierbar sind

1666 Datenbanken in Rechnernetzen
Wie funktioniert die Synchronisierung von globalen Transaktionen bei verteilten Datenbanksystemen.
2PC-Protokoll erklären mit Freigabe der Sperren.
Zu welchen Zeitpunkt ist für die globale Transaktion Serialisierbarkeit gewährleistet.
Was ist die teuerste Operation bei Abfragen in vDBMS
Welche joins gibt es bei vDBMS?

Die Prüfung verlief in sehr angenehmer Atmosphäre.
Tip: Für die Prüfung in vDBMs sollte man das 2 PC-Protokoll in allen Einzelheiten flüssig beherrschen, dann ist die Prüfung schon halb gewonnen.

Prüfungsprotokoll

1666 - Datenbanken in Rechnernetzen

Prof. Dr. Schlageter

15. September 1998

Fragen

- Aus welchen Gründen werden verteilte Datenbanken eingesetzt?
- Wie werden globale Transaktionen ausgeführt?
- Wie kann ein globaler Join möglichst effizient ausgeführt werden?
- Auf welche Weise wird die Konsistenz der verteilten Datenbank gewährleistet?
- Warum garantiert das 2PC-Protokoll Serialisierbarkeit?
- Wie können Relationen auf die verschiedenen Knoten verteilt werden?

Eindruck

Wie bei Prof. Schlageter wohl üblich, begann die Prüfung mit ca. einer Viertelstunde Verspätung. Die Atmosphäre war zunächst wie üblich recht locker, was ich diesmal allerdings nicht optimal ausnutzen konnte. Insbesondere bei dem Themenkomplex **2PC-Protokoll** fragte Prof. Schlageter sehr akribisch nach und ließ nicht locker, bis er mir nach und nach alles entlockt hatte, was er dazu wissen wollte. Obwohl die Fragen beim Vergleich mit anderen Prüfungsprotokollen wenig Überraschungen beinhalteten, war der Prüfungsverlauf ziemlich zäh, weil ich zeitweise das sprichwörtliche Brett vor dem Kopf hatte. In der Benotung hat sich das dann mit leichten Abschlägen bemerkbar gemacht; insgesamt ist die Bewertung aber sehr fair.

Die Frage, wie das 2PC-Protokoll Serialisierbarkeit garantiert, scheint obligatorisch zu sein. Es ist also wärmstens zu empfehlen, als Prüfungsvorbereitung die zugehörigen Kapitel im Kurs sehr gründlich durchzulesen.

Diplomprüfung Vertiefende Kurse der Informatik

Kurs : Datenbanken in Rechnernetzen 1666

Datum : 17.12.1997
Zeit : 11 .OO Uhr

Dauer : 12 Minuten

Prüfer : Prof. Dr. Schlageter

Beisitzer : Dipl. Inform. Laskowski

Note : 1.3

- Was will man mit verteilten Datenbanken erreichen? Was sind die Ziele?
- Welche Möglichkeiten der Datenverteilung gibt es und was soll damit erreicht werden?
- Wie werden Updates auf replizierte Datenbanken behandelt?
- Welche Methoden zur Konsistenthaltung von Daten kennen Sie?
- Wann kann man Inkonsistenzen in Kauf nehmen?
- Welche Möglichkeiten der Join-Ausführung gibt es in verteilten Datenbanken?
- Wie garantiert das 2PC-Protokoll Serialisierbarkeit?
- Was ist bei der Query-Optimierung in verteilten Datenbanken zu beachten?
- Was haben 2PL- und 2PC-Protokoll gemeinsam?

Insgesamt war die Prüfung angenehm - wie immer bei Prof. Schlageter. Er hatte jedoch an diesem Tag wenig Zeit, so daß meine Prüfung erst mit einiger Verspätung begann. Dadurch fiel die Prüfung auch ausgesprochen kurz aus.

Nach der Prüfung hatte ich kein allzu gutes Gefühl, da ich mich bei den Antworten auf die - wohl wegen der Zeitknappheit - in schneller Folge gestellten Fragen etwas verhaspelt hatte.

Vor der Notenbekanntgabe fragte Prof. Schlageter denn auch: “Es interessiert mich, was Sie für ein Gefühl haben?!“. Mein Gefühl lag irgendwo bei 2.X; deshalb war ich über die Note, s.o., sehr angenehm überrascht. Zur Erklärung sagte Prof. Schlageter: “Wir waren überzeugt, daß Sie die Zusammenhange erkannt und verstanden haben.”

Viel Glück bei Eurer Prüfung!

Mündl. Diplomprüfung Datenbanken in Rechnernetzen (01666)

Datum: 29.07.97
Prüfer: Prof. Schlageter
Prüfungsdauer: ca. **20** Minuten

1. *Nennen Sie die Ziele, die ein verteiltes DBS erfüllen soll?*
   Siehe 1 KE, aufzählen der dort angegebenen Ziele, mit kurzer Erklärung (ca. 2 Sätze zu jedem Punkt).
2. *Wie werden Kopien verarbeitet?*
   Replikationsverfahren angeben, wichtigsten Kriterien zu deren Einteilung;
   nachdem ich die ersten 3 Verfahren erkärt hatte, brach Prof. Schlageter dieses Thema ab.
3. *Wie wird Synchronisation bei verteilten DBS realisiert?*
   2-PC-Protokoll genau erklären;
   dabei ruhig auch Dinge aus dem Kurs 01665 (Datenbanksysteme) angeben und erklären (z.B. 2PL-Protokoll plus Zusammenhang zum 2-PC-Protokoll exakt angeben, physische Log erklären).
   Während meiner Ausführung wurden vereinzelt Zwischenfragen gestellt, die ich nicht alle wiedergeben kann; eine war:
   Besteht ein zeitlicher Zusammenhang zwischen der Prepare-to-Commit-Nachricht und dem Zeitpunkt, an dem eine Subtransaktion alle von ihr benötigten Sperren erhalten hat? Nein.
4. *Im Kurs wurden Verfahren zur Anfrageoptimierung genannt, wie heißen sie?*
   Die Join-Verfahren angeben.
5. *Was ist die einfachste Art, einen Join durchzuführen?*
   Man schickt die kleinere Relation komplett zur größeren und verarbeitet die Gesamtmenge an dem Knoten der größeren Relation.
6. *Erklären Sie mir die Join- Verfahren, die ein Informatiker durchführen würde.*
   Siehe Kurs.

Viel Erfolg bei Eurer Prüfung.

Gedächtnisprotokoll zur Diplomprüfung: Prakt. Informatik, Teil 1 **November 1996**

Kurs: 1666 Datenbanken in Rechnernetzen **Dauer: 20 min**
Prüfer: Dr. G. Schlageter **Note: 1.0**

1. Welche Ziele verfolgt man durch den Einsatz verteilter Datenbanken ?
2. Wie kann man Relationen in vDBen verwalten 3
3. Wie kann man Serialisierbarkeit in vDBen sichern ?
   (2 Phasen- Commit-Protokoll geschildert)
4. Wie wird im 2PC Serialisierbarkeit gesichert ?
   (Nach RtC werden die Sperren **nicht** freigegeben, erst nach dem Gobal-Commit)
5. Kann es beim 2PC-Protokoll zu Verklemmungen kommen und wenn ja wie kann man das Problem lösen ?
6. Welche Transkation bricht man ab ?
7. Welche Kostenfaktoren gibt es dazu ?
   (Wichtig: die Kommunikationskosten nicht vergessen)
8. Welche Möglichkeiten hat man in vDBen Daten zu verteilen ?
   (Horizontale, vertikale, replizierte Speicherung)
9. Welche Grundaspekte gibt es bei der Replikation zu beachten ?

10. Welche Möglichkeiten hat man bei der Ausfiihrung von Anfragen zusätzlich bei vDBen?
    (Man kann sie auf einem oder mehreren Knoten ausführen)

Wichtig scheint mir vor allem zu sein, daß man einen guten Überblick hat.

Als Prüfer zu empfehlen.

JG

Diplomprüfung Praktische Informatik

Prüfer: **Prof. Schlageter**

Termin: **16.10.96**

Dauer: **20 min**

Note: **1,3**

Prüfungsinhalte: **Kurs 1666 - Datenbanksysteme in Rechnernetzen**

Prüfungseindruck:

Prof. Schlageter ist ein sehr guter Prüfer! Er versucht dem Prüfling die Nervosität zu nehmen, indem er ihn auffordert zu einem vorgegebenen Thema das anzubringen, was der Prüfling weiß. Zudem wechselt er zu einem anderen Thema, wenn man mal etwas nicht genau weiß (nachdem er etwas nachgebohrt hat) und gibt einem dort wieder alle Chancen.

Die bevorzugten Themen sind:

- allg. über Optimierung von Querys (Kommunikationskosten)
- Joins in verteilten Datenbanken
- 2PC-Protokoll
- 2PL-Protokoll und optimistische Verfahren
- Deadlockerkennung und -auflösung

Ein globales Verständnis, insbesondere der Zusammenhänge, ist wesentlich wichtiger als Details der einzelnen Algorithmen zu kennen!

Prüfungsverlauf:

1. Was ist besonders zu beachten bei der Queryoptimierung in DDBMS?

- Kommunikationskosten zu minimieren; allg. Optimierung erläutert

2. Wie werden Transaktionen korrekt ausgeführt/serialisiert?

- Kombination aus 2PC- und 2PL-Protokoll; beides erläutern!
- Logfile Bedeutung erläutert
- Wichtig: nach RtC werden die Sperren auf Objekte der Teiltransaktion nicht freigegeben! Daher kann keine weitere (Teil-)Transaktion auf diese Objekte zugreifen!

3. Thema Verklemmung: Wie kann man vorgehen?

- zentrale Analyse und Auflösung
- verteiltet Lösung: DDD-Algo. erläutert

4. Wie werden Joins in DDBMS verwirklicht?

- Nested-Loop-Join (Bemerkung Schlageter: "Eines Informatikers unwürdig!")
- Sort-Merge Join: grobe Schilderung des paralellen Durchlaufs
- Semi-Join Verfahren: Wichtig zu erwähnen ist die gewünschte Reduzierung der "Relation B"
- Hashfilter Verfahren: Minimierung des Overhead im Gegensatz zum Semi-Join Verfahren; Güte ist abhängig von der Hashfunktion und der Länge des Bitvektors. Grund für "Pseudo-Treffer" nennen können!