

Diplomprüfung praktische Informatik

Datum: 03.April 2008

Vorlesungen: 1. Einführung in Datenbanksysteme (nach Kemper & Eickler)
2. Indexstrukturen für DB (nach Folien Seidl WS 07/08)
3. Betriebssysteme I + II (nach Tanenbaum)

Prüfer: Prof. Dr. rer. nat. Thomas Seidl
Beisitzer: Dipl. Inf. Christoph Brochhaus

Prüfling: Martin Wermers
Note: 1,0

Einführung in Datenbanksysteme

- Erklärung des relationalen Datenbankschema (Def. Relation, Elemente des Schemas: Attribute & Domänen, sollte man auch allgemein, ohne Beispiel erklären können.

Da ich bei der Definition der Relation sagte, dass die Domänen endlich seien müssten (obwohl die mathematische Definition darüber nichts aussagt), fragte Prof. Seidl nach wieso ich denn der Meinung sei.

Antwort:

1. weil es in Computern keine unendlich große Datentypen gibt, sondern immernur endliche Teilmengen, wie z.B. Integer.
2. weil sichere Ausdrücke in den beiden Relationenkalkülen endliche Domänen benötigen.

Prof. Seidl sagte, dass dies korrekt ist, für die mathematische Definition der Relation allerdings nicht notwendig.

- Schlüsselbegriff (funktionale Abhängigkeiten, volle f.A., Superschlüssel, Kandidatenschlüssel, Primärschlüssel)
- Integritätsbedingungen (referentielle Integrität für Fremdschlüssel)
- Tupelkalkül & Domänenkalkül (Unterschiede erklären, also Tupelvariablen vs. Domänenvariablen; allgemeine Form einer Anfrage aufschreiben; Erklärung freie & gebundene Variablen)
- Der Synthesealgorithmus kam erstaunlicherweise nicht dran, genauso wenig wie das typische Studenten – Vorlesungs Beispiel. Sollte man allerdings auch können, kam schon oft genug dran – vgl. andere Protokolle.

Indexstrukturen für Datenbanken

- Indexstrukturen für 1-dim Daten nennen (Hashing in div. Varianten, Bitmap-Index, B-Baum)
 - Erklären von Bitmap-Index (wurde seltsamerweise nicht nach dem bereichskodierten & intervallkodierten Index gefragt – sollte man aber auch können, genauso wie natürlich Hashing & B-Baum. Alles ähnlich wahrscheinlich, dass es abgefragt wird)
- Frage nach mehrdimensionalen B-Baum (--> R-Baum)
 - Erklärung der Struktur (Objekte werden durch MUR approximiert, die MUR dann im R-Baum gespeichert) & nennen von Eigenschaften (hierarchisch, balanciert, etc.)
 - Erklärung des Verzweigungsgrades mit m und M .

1. Frage: Wieso wählt man typischerweise $m = 40\% M$ und nicht wie beim B-Baum $m = 50\% M$:

Antwort: Weil die Splitalgorithmen nach einer Heuristik arbeiten, und man in der PickNext Iteration ggf. ein Rechteck nicht mehr zu dem Knoten hinzufügen kann, zu dem man das nach Heuristik tun würde, da ansonsten im anderen Knoten ein Unterlauf („Verletzung von m “) entstehen würde. Wählt man m aber kleiner, so können mehr Rechtecke zum heuristisch besseren Knoten hinzugefügt werden.

2. Frage: Und wieso muss $m \leq 50\% M$ sein, könnte das nicht auch größer sein?

Antwort: Nein, dann würde es bei einem Split (Aufteilung von $M+1$ Rechtecken) definitiv in einem Knoten einen Unterlauf geben!

- Eigenständige Erwähnung des Problems der Überlappungen beim R-Baum, damit Beantwortung der häufig gestellten Frage, ob R-Baum entarten kann (Baumstruktur nein, Anfrageeffizienz ja bei vielen Überlappungen).

Dazu eine **Komplexitätsfrage** von Prof. Seidl:

Falls es Überlappungen gibt haben sie schon gesagt kann es ggf. nötig sein den kompletten Baum zu lesen und in der Praxis ist das auch ohne Frage langsamer als ein sequentieller Scan (wegen dem Overhead der Directoryknoten) Wie ist denn dann die Komplexität? Noch in $O(n)$ oder schlechter?

Antwort: Weiß ich leider nicht mehr genau. Man muss allerdings noch die Komplexität berücksichtigen, die für das Lesen aller Directoryknoten benötigt wird. Er wollte nur eine ungefähre Antwort haben, wieviele Directoryknoten man denn hat, wenn man N viele Rechtecke in den Blättern hat. Konnte ich nicht direkt beantworten, er fragte mich daher (zur Vereinfachung), wie das denn im Binärbaum aussieht. Dort hat man $2^h - 1$ viele innere Knoten.

- Was kann man denn in hohen Dimensionen tun?
Antwort: 3 Möglichkeiten:

1. Verbesserung der Splitheuristiken zur Minimierung von Überlappungen (z.B.: X-Baum)
2. Verbesserung des seq. Scans (Bsp.: VA-File)
3. Dimensionsreduktion als Filter

- Daraufhin ausführliche Erklärung des VA-Files:

- Erklärung des Vector File & Vector Approximation File (haben gleiche Reihenfolge, Einfügen immer hinten). Die Idee ist letzteres nicht nach der Raumpartitionierung zu strukturieren, sondern darin die Raumapproximation (d.h. die Zelle in dem sich ein Element befindet) zu speichern.

- Prof. Seidl wollte die quantilbasierte Erklärung des Gitters recht genau erklärt haben:

Frage: Kann es auch vorkommen, dass Zellen trotz quantilbasiertem Gitter leer sind?

Antwort: Ja (Erklärung anhand eines Beispiels im 2D mit 4 Zellen)

- Zudem wollte er auch wissen, wieviele Bits man denn für die Approximation so typischerweise wählt (konnt mich nicht entsinnen, das auf einer Folie explizit gelesen zu haben, gab allerdings auf Folien wohl ein Beispiel mit einer bestimmten Bitzahl, die hatte ich aber dummerweise nicht im Kopf). Da ich nicht direkt eine Antwort geben konnte, hat er mir den Tipp gegeben, dass man ja sicherlich weniger Bits nehmen sollte, als man für die exakte Repräsentation braucht. Er fragte dann, wieviele Bits denn z.B. ein Float hat. Der hat typischerweise 32 Bits. Würde man also auch 32 Bit zur Kodierung der Zellen wählen, hätte man überhaupt keinen Vorteil. Prof. Seidl hat mir dann verraten, dass typischerweise so 3 – 6 Bits verwendet werden, je nach Daten ggf. auch etwas mehr. Auf jeden Fall wählt man die Anzahl Bits aber so, dass sich die Approximation auch lohnt.

Betriebssysteme I + II

Eigentlich nur die Standardfragen, die auch in den anderen Protokollen vorkommen. Deadlocks musste ich seltsamerweise gar nicht erklären, aber die sind auch ein häufig abgefragtes Thema. Bei mir wurde folgendes abgefragt:

- Prozessbegriff (Befehlszähler, Variablenbelegung, Registerinhalte und die 3 möglichen Zustände)
- Über die Frage, ob Prozesse denn vom OS wissen müssen zu Systemaufrufe gekommen. Musste nur den Zweck erklären, nicht mal den genauen Ablauf eines Aufrufs. Es kann aber sicher auch nicht schaden den Ablauf zu kennen (Bibliotheksfunktion, TRAP-Aufruf, Wechseln in Kernmodus etc.)
- Multimedia OS (harte & weiche Echtzeitanforderung, RMS & EDF erklären können)
- Dateisystemrechte. Hier wollte Prof. Seidl 3 Antworten hören:
 1. Wer (Benutzer/Gruppe) darf
 2. was (rwx Bits) und
 3. wo sind diese Rechte vergeben (also auf welche Ressourcen)

Zudem: Was für weitere Rechte kann es geben?

Antwort: In Windows gibt es z.B. das Recht jemandem Rechte zu geben bzw. zu entziehen.

- Abschließende Frage war, wie man den Drucker jemandem entziehen kann, bzw. wenn das möglich wäre, wo man ansetzen müsste. Bzw. er war sich da selbst nicht so sicher. Mir kam es so vor, als hätte er sich das Beispiel auf die Schnelle ausgedacht und dabei festgestellt, dass das zu dem, was er gerne als Antwort hätte, gar nicht so richtig passt.

Er wollte jedenfalls darauf hinaus, dass in Unix Geräte z.B. als virtuelle Dateien (in /dev) realisiert werden, und es dort auch Zugriffsrechte gibt. Bin mir aber nicht sicher, ob es für den Drucker überhaupt so eine Datei gibt. Meines Wissens nach hat in Unix der Drucker-Daemon alleinig die Zugriffsrechte auf den Drucker, daher bin ich nicht darauf gekommen, sondern habe den Drucker-Daemon und Spooling erklärt. Das war wie gesagt nicht das, was er eigentlich hören wollte, war aber auch nicht weiter schlimm.

Resümee

Professor Seidl ist ein sehr angenehmer Prüfer, es herrscht eine lockere Atmosphäre. Wenn man mal nicht direkt weiter kommt, dauert es keine 10 Sekunden, bis man eine kleine Hilfestellung bekommt. Wenn man mal direkt nicht die richtige Antwort (oder die richtige Ausdrucksweise, die er erwartet) bringt, dann ist das nicht schlimm, sofern er merkt, dass man ansich weiß, was gemeint ist (bei mir z.B. bei der differentiellen Integrität). Das ist zwar ein kleines Minus, kann man aber z.B. durch sehr gute Antworten (wie z.B. meine Erklärungen bzgl. m und M beim R-Baum, die haben ihm sehr gut gefallen) wieder ausgleichen. Beim VA-File wurde mir (minimal) angelastet, dass ich kein typisches Beispiel für die Anzahl der Bits wusste, aber das war auch nicht schlimm, weil mir ja der Sinn mit dem Denkanstoss wieviele Bits denn z.B. ein Float benötigt durchaus klar wurde.

Alles in allem kann ich Prof. Seidl als Prüfer sehr empfehlen.

Vorbereitet habe ich mich „eine ganze Zeit“, indem ich das DB-Buch von Eickler über mehrere Wochen während der Vorlesungszeit gelesen habe. Zudem war ich auch in Indexstrukturen – das Bearbeiten der Übungsaufgaben kann ich nur empfehlen. So richtig gelernt habe ich allerdings nur die letzten 2 Monate. Da ich im ersten davon auch noch arbeiten musste, eigentlich so richtig stark nur im letzten (inkl. Lernen der letzten Themen bis 5h morgens vor der Prüfung... *gähnh*, daher hab ich mir jetzt gleich echt mal das erste Bier des Tages verdient! Sorry Micha, dass ich mich wegen dem Protokoll verspäte, aber in ein paar Tagen würd ich es sonst eh nicht mehr schreiben, ich kenn mich :D)

Ich danke meiner netten Kommilitonin Seran für die Lernunterstützung und das gemeinsame Durchkauen des zähen Synthesealgorithmus und der ersten drei Normalformen am abend vor der Prüfung, meiner Freundin für ihre leckere Lasagne, sowie allen Leuten, die ein Prüfungsprotokoll geschrieben haben! Die sind sowas von nützlich! Vor allem bei einem so großen Thema wie Betriebssysteme (der Tanenbaum hat 1000 Seiten, die kann und muss man in der Zeit aber gar nicht alle lesen). In Betriebssysteme wird aber nur ein eher kleiner Teil des Buches (ca. 250 Seiten) abgefragt, wie man anhand der Prüfungsprotokolle erkennen kann.

Ach ja: Ich habe wegen dem Drang nach einem leckeren Getränk und einem guten Mittagessen nur die eher schwierigen Fragen hier beantwortet. Die einfachen Sachen müsst ihr eh können, daher hier nur die Übersicht was ich gerade gefragt wurde, und grob, was ich geantwortet habe. Natürlich ist das in der Prüfung dann mehr, aber ihr packt das schon ;-)

Ich wünsche Euch viel Erfolg beim Lernen und der Prüfung!