Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Datenbanksysteme · Prof. Dr. Grust

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Übungen zur Vorlesung
"Datenbanksysteme II"
SS 2014
Benjamin Dietrich (b.dietrich@uni-tuebingen.de)

2. Übungsblatt

Ausgabe: 16. April 2014 · Abgabe: 22. April 2014

Aufgabe 1: Bélády's Anomalie (4 Punkte)

> "Eine Vergrößerung des Bufferpools (Anzahl der verfügbaren Seiten) verbessert in jedem Fall die Effektivität der Seitenersetzungsstrategie, und minimiert die Anzahl der Seitenfehler."

Im Jahr 1969 zeigte der ungarische Informatiker Lázló Bélády, dass es sehr wohl möglich ist, trotz einer Vergrößerung des Buffers, mehr Seitenfehler auszulösen als bei einem kleineren Buffer. Dieses Phänomen wird in der Literatur als *Bélády's Anomalie* bezeichnet.

1. Zählen Sie die Anzahl der Seitenersetzungsfehler für die Algorithmen `FIFO` (`F`irst `I`n `F`irst `O`ut) und `LRU` jeweils für eine Buffergröße von drei bzw. vier Seiten miteinander, wenn die verfügbaren Seiten in der folgenden Reihenfolge vom Datenbanksystem *gepinnt* werden:

$$p_1, p_2, p_3, p_4, p_1, p_2, p_5, p_1, p_2, p_3, p_4, p_5$$

2. Welche Eigenschaften muss ein Seitenersetzungsalgorithmus haben, damit die Anomalie nicht auftritt?

Aufgabe 2: LRU-k (8 Punkte)

In der Vorlesung haben wir bereits einen Eindruck gewonnen, wie LRU arbeitet. Als Nächstes wollen wir den auf LRU basierenden LRU-k[1] Algorithmus etwas genauer betrachten.

Im Gegensatz zu LRU behält LRU-k für jede Seite die Zeiten der letzten k Zugriffe im Auge. Diese Informationen werden dann zur Auswertung herangezogen, wie häufig die Seite vom Datenbanksystem referenziert wird.

Gegeben sei eine Menge $N = \{p_1, p_2, \ldots, p_n\}$ an *disk pages*. Die *pages* werden vom Datenbanksystem sukzessive referenziert, diese Referenzierung $r_1, \ldots, r_t, \ldots$, mit $r_t = p\,(p \in N)$ werden wir im Folgenden als "Referenzstring" bezeichnen.

Zunächst definieren wir noch eine Metrik, anhand derer LRU-k die zu ersetzende Seite ermittelt.

[1] http://www-2.cs.cmu.edu/~christos/courses/721-resources/p297-o_neil.pdf

Definition 1 *Backward k-Distance* *Gegeben Sei ein Referenzstring $r_1, r_2, \ldots, r_t$. Die* Backward k-Distance $b_t(p,k)$ *ist wie folgt definiert:*

$$b_t(p,k) = \begin{cases} x, & \textit{wenn } r_{t-x} = p \textit{ und es im Referenzstring exakt} \\ & k-1 \textit{ andere Werte } i \textit{ mit } t-x < i \leq t \textit{ gibt,} \\ & \textit{sodass } r_i = p. \\ \infty, & \textit{wenn } p \textit{ nicht mindestens } k \textit{ mal im Referenz-} \\ & \textit{string } r_1, \ldots, r_t \textit{ vorkommt} \end{cases}$$

Die Arbeitsweise des Algorithmus ist wie folgt:
Wenn der Buffer voll ist, dann wird die Seite mit der maximalen Backward-Distanz ausgelagert. Gesetzt den Fall, dass mehrere Seiten die Backward-Distanz ∞ besitzen, so muss eine andere Auslagerungsstrategie zum Einsatz kommen, wie zum Beispiel der klassische LRU Seitenersetzungsalgorithmus.

1. Können Sie einen geeigneten Parameter k finden, sodass der LRU-k Algorithmus äquivalent zu LRU ist? Erklären Sie!

2. Szenario:
   Gegegeben seien Transaktionen $\mathcal{T}_1$ und $\mathcal{T}_2$, und ein Buffer, der 11 Seiten aufnehmen kann.
   Die Transaktionen referenzieren Seiten aus einer Tabelle $\mathcal{R}$ einer Größe von 100 Seiten.
   Die Transaktionen $\mathcal{T}_1$ und $\mathcal{T}_2$ haben folgende Eigenschaften:

Transaktion $\mathcal{T}_1$	$\mathcal{T}_1$ referenziert lediglich eine Teilmenge der Seiten von $\mathcal{R}$, nämlich die Seiten von 1 bis 10. Der Referenzstring ist also $\underbrace{p_1, p_2, \ldots, p_{10}}_{\times 10}$.
Transaktion $\mathcal{T}_2$	$\mathcal{T}_2$ referenziert sequentiell alle Seiten (*scan*). Das heißt, der Referenzstring ist $p_1, \ldots, p_{100}$.

   Beginnend mit Transaktion $\mathcal{T}_1$, referenzieren die Transaktionen alternierend die Seiten der Tabelle $\mathcal{R}$. Der Referenzstring r sieht also folgendermaßen aus:

   $$r = p_1, p_1, p_2, p_2, p_3, p_3, p_4, p_4, \ldots, p_{10}, p_{10}, p_1, p_{11}, p_2, p_{12}, p_3, p_{13}, \ldots, p_{10}, p_{100}$$

   Zählen Sie die Anzahl der Seitenfehler, auf Basis des gegegebenen Szenarios, für den klassischen LRU Algorithmus und für LRU-2! Welche Vor- und Nachteile erkennen Sie?

Aufgabe 3: Page Format (8 Punkte)

Daten werden von Datenbanksystemen in verschiedenster Art und Weise auf dem Sekundärspeicher gespeichert. Tabellen werden normalerweise in Seiten (*data pages*) organisiert.

In der Vorlesung haben Sie zwei verschiedenene Arten solcher *page layouts* kennengelernt. So kann man die Daten einer Tabelle tupel- oder spalten-weise (*row-wise/column-wise*) in Seiten ablegen.

Stellen Sie sich vor, Sie hätten die folgende Tabelle in Ihrem Datenbanksystem angelegt:

professoren

PNr	Name	Rang	Gehalt
2125	Sokrates	C4	20
2126	Russel	C4	40
2127	Kopernikus	C3	12
2133	Popper	C3	23
2134	Augustinus	C3	34
2136	Curie	C4	19
2137	Kant	C4	11
⋮	⋮	⋮	⋮

Diese Tabelle enthält insgesamt 100,000 Tupel. Die verschiedenen Attribute `PNr`, `Name`, `Rang`, `Gehalt` sind jeweils vom Typ `INTEGER` (4 Bytes), `CHAR(30)` (30 Bytes), `CHAR(2)` (2 Bytes), und `INTEGER`. Keines der Atttritube kann `NULL` werden. Ihre Datenbank arbeitet auf einer Seitengröße von 1 KB. Auf jeder Seite nimmt der *page header* jeweils 68 Byte ein. Das dynamisch wachsende Slot Directory benötigt für jeden Record genau ein Bit. Bitte beachten Sie hierbei aber, dass auf jeder Page nicht einzelne Bits, sondern nur Speicher von mindestens einem Byte alloziiert werden kann.

Auf Basis der oben gezeigten Tabelle beantworten Sie bitte folgende Fragen:

1. Berechnen Sie die Breite eines Records. Gehen Sie dabei vereinfachend davon aus, dass ein Record keine Meta-Informationen enthält. Berechnen Sie den unbenutzten Speicher auf Basis der Ihnen zu Verfügung stehenden Informationen, für beide *page layout*-Alternativen!

2. Folgende Queries werden auf Ihrer Datenbank ausgeührt:

   (a) `select * from professoren`

   (b) `select Name from professoren`

   Berechnen Sie jeweils die Anzahl der Seiten, die bei der tupel- bzw. spalten-weisen Organisation vom Datenbanksystem referenziert werden müssen, um die Anfragen Q_1 und Q_2 zu beantworten.

3. Welche Vorteile und Nachteile erwarten Sie bei der tupel- gegenüber der spalten-weisen Organisation der Daten?