

Prof. Dr. T. Härder
 Fachbereich Informatik
 AG Datenbanken und Informationssysteme
 Universität Kaiserslautern

Übungsblatt 2

für die freiwillige Übung

Unterlagen zur Vorlesung: „www.dvs.informatik.uni-kl.de/courses/DBSREAL/“

Aufgabe 1: Indirekte Seitenadressierung, Schattenspeicher- und Zusatzdateikonzept 81

Eine Datenbank bestehe aus zwei Segmenten mit jeweils 8 Seiten, zu deren Abspeicherung insgesamt 32 Blöcke zur Verfügung stehen. Auf dieser DB laufen zeitlich überlappt zwei Transaktionen ab, die Seiten der DB in der nachfolgend aufgeführten zeitlichen Sequenz verändern:

T1: P12 P15 P24 P17 EOT
 T2: P21 P25 P16 P27 EOT

P_{ij} steht für Seite j in Segment i

EOT bezeichnet das Ende der Transaktion

Machen Sie sich an diesem Beispiel die Funktionsweise der indirekten Adressierung, des Schattenspeicher- und des Zusatzdateikonzeptes klar. Vor Beginn von T1 seien beide Segmente geschlossen.

Bei indirekter Adressierung sei folgende Zuordnung der Seiten (P) zu Blöcken (B) gegeben:

P11 P12 P13 P14 P15 P16 P17 P18 P21 P22 P23 P24 P25 P26 P27 P28
 B1 B3 B6 B5 B7 B13 B9 B10 B15 B18 B19 B17 B21 B28 B24 B26

- Bestimmen Sie den Aufbau der Seitenzuordnungstabellen bei indirekter Seitenadressierung.
- Welche zusätzlichen Hilfsstrukturen werden beim Schattenspeicherkonzept benötigt. Bestimmen Sie deren Modifikationen beim Ablauf der Transaktionen.
- Führen Sie für das Schattenspeicherkonzept nach EOT von T1 einen Sicherungspunkt für beide Segmente durch, nach EOT von T2 einen für Segment 2.
- Was muß am Schattenspeicherkonzept geändert werden, um transaktionsbezogene Sicherungspunkte implementieren zu können? Vollziehen Sie einen solchen Sicherungspunkt bei EOT am Ende von T1 am Beispiel nach.

Aufgabe 2: Bloom-Filter 183

Mit Hilfe des Bloom-Filters kann beim Zusatzdatei-Verfahren einfach entschieden werden, ob eine Seite verändert worden sein kann und deshalb möglicherweise in der Zusatzdatei steht, oder ob dieses nicht der Fall sein kann.

Gegeben seien ein Bitvektor der Länge 8 und eine Hashfunktion $h(x)$, welche die Positionen der 1-en liefert, die bei folgender Funktion $g(x)$ gesetzt werden:

$g(x) = (\text{Binärdarstellung von } x) \text{ XOR } (01010101)$.

Die Seiten mit den Schlüsselwerten 31, 53 und 62 werden verändert. Anschließend sollen die Seiten mit den Schlüsselwerten 53, 93 und 124 gelesen werden.

Welche Ergebnisse liefert der Bloom-Filter bei den Leseoperationen?

Warum ist die angegebene Hashfunktion für den Bloom-Filter ungeeignet und welche Eigenschaften muß eine bessere Hashfunktion haben?

Aufgabe 3: Lokalität und Sequentialität 173

Eine Transaktion erzeuge folgenden Referenzstring:

AABBCEADABGHAAGGHIIKKLKLKLMABABKLKLKLFMFAGHI

Beachten Sie hierbei, daß als sequentieller Reihenfolge angenommen wird: ABCDEFGHIKLMNO

Berechnen Sie folgende Größen:

- Die aktuelle Lokalität zu den Zeitpunkten $t = 6, 18, 28$ und der Fenstergröße 6.
- Die durchschnittliche Lokalität.
- Die sequentiellen Zugriffsfolgen sowie das Maß der Sequentialität der Transaktion.
- Die LRU-Stacktiefenverteilung.

Diskutieren Sie die dynamischen Zuordnungsentscheidungen von Pufferseiten für diese Transaktion. Wie sind die vorgestellten Entscheidungsgrößen effizient im DBMS zu ermitteln.

Aufgabe 4: Suche im DB-Puffer

Im Skript, 3 - 15, sind mehrere Verfahren zur Unterstützung der schnellen Suche einer DB-Seite im DB-Puffer skizziert. Analysieren Sie die folgenden Verfahren im Hinblick auf Wartungs- und Suchkosten:

1. unsortierte Tabelle
2. sortierte Tabelle
3. Tabelle mit verketteten Einträgen (LRU-Reihenfolge)
4. AVL-Baum
5. Hash-Tabelle mit Überlaufketten

Annahmen:

Bei jeweils 10 Suchvorgängen wird die Seite neunmal gefunden, einmal wird sie nicht gefunden (Fehlseitenbedingung), so daß eine Ersetzung erfolgen muß.

Das i -te Verfahren habe dabei m_i Kosteneinheiten w an Wartungskosten und pro Suchvorgang s_i Kosteneinheiten c an Suchkosten.

Jede Tabelle oder Datenstruktur habe $n = 2^{10}$ Einträge. Zur Aufrechterhaltung der Tabellensortierung nehmen wir an, daß $n/2$ Einträge verschoben werden müssen zu $n/2$ Kosteneinheiten w .

Bei LRU-Reihenfolge nehmen wir an, daß eine Seite, die sich im Puffer befindet, mit $n/10$ Kosteneinheiten c gefunden wird. Beim Hash-Verfahren sollen im Mittel 1,5 Vergleiche anfallen. Bestimmen Sie die Kosten C_i , die für jeweils 10 Suchvorgänge mit einer Ersetzung anfallen.

Schätzen Sie die anfallenden Kostenanteile in geeigneter Weise ab!

Welches Verfahren gewinnt, wenn $w = 5 * c$ gesetzt wird?

Ändern sich diese prinzipiellen Aussagen, wenn die mittlere Lokalität nicht 90% ($10/11$), sondern $x\%$ ($(x+1)/y$) beträgt? Diskutieren Sie $x = y$.