



3. Übung - Systempuffer

Architektur von Datenbanksystemen I

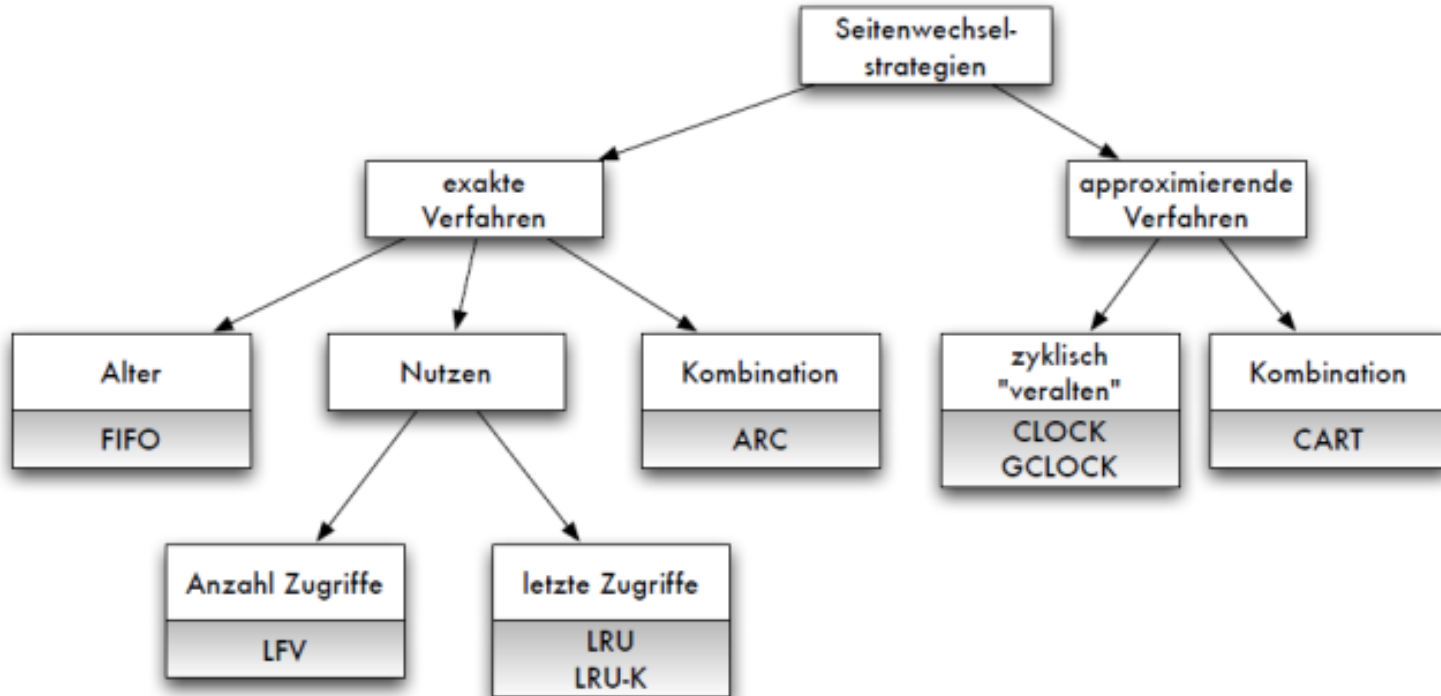
Pufferverwaltung

1. Gegeben sei ein DBMS mit einer Puffergröße von 5 Seiten. Die Seiten a, b, c, d, e, f, g werden während einer Transaktion in folgender Reihenfolge eingelesen:

$a, b, c, a, b, c, d, e, c, d, f, a, d, b, g, a, g, e, a, c$

geben Sie die jeweiligen Seitenersetzungen für die FIFO-, LFU-, CLOCK- und LRU- sowie LRU-2-Strategie an. Bestimmen Sie außerdem die jeweilige Trefferrate!

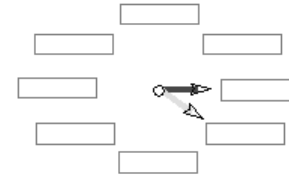
Gängige Strategien)



FIFO (First In First Out)

ERSETZUNG DER ÄLTESTEN SEITE IM DB-PUFFER

- Veranschaulichung durch kreisförmig umlaufenden Uhrzeiger
 - Zeiger zeigt auf älteste Seite
 - Bei Fehlseitenbedingung wird diese Seite ersetzt und der Zeiger auf die nächste Seite fortgeschaltet



MERKMALE

- Unabhängigkeit vom Referenzverhalten
(nur das Alter seit der Einlagerung ist entscheidend)

BEWERTUNG

- + bei strikt sequenziellem Zugriffsverhalten
- bei Direktzugriff
(häufig benutzte Seiten sollen ja gerade im Puffer bleiben und dort "alt" werden)

ERWEITERUNG: CLOCK, GCLOCK UND DGCLOCK

Ablauf - FIFO

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
F1																				
F2																				
F3																				
F4																				
F5																				

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
C1																				
C2																				
C3																				
C4																				
C5																				

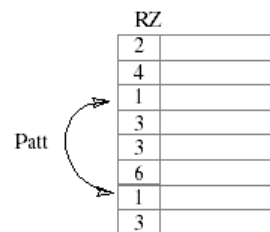
LFU (Least-Frequently-Used)

ERSETZUNG DER SEITE MIT NIEDRIGSTER REFERENZHÄUFIGKEIT

- Führen eines Referenzzählers pro Seite im DB-Puffer

MERKMALE

- nur Referenzverhalten geht ein, nicht das Alter!
- mögliche Pattsituationen müssen durch eine Sekundärstrategie aufgelöst werden
- beim sequentiellen Lesen wird jede Seite einmal referenziert Strategie nicht anwendbar



BEWERTUNG

- + häufig benutzte Seiten werden im Puffer gehalten
- Seiten, die punktuell sehr intensiv benutzt werden und dann nicht mehr, sind praktisch nicht zu verdrängen

Ablauf - LFU

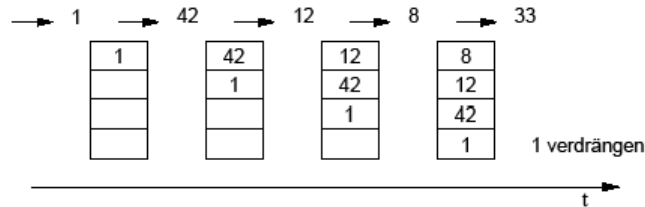
	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
F1																				
F2																				
F3																				
F4																				
F5																				

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
C1																				
C2																				
C3																				
C4																				
C5																				

LRU (Least-Recently-Used)

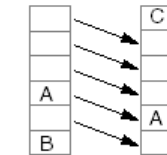
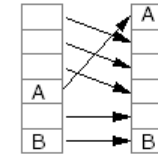
ERSETZUNG DER SEITE, DIE AM LÄNGSTEN NICHT MEHR REFERENZIERT WURDE

- Seiten werden als LRU-Stack verwaltet
- bei jeder Referenz kommt die Seite in die oberste Position



MERKMALE

- bewertet das Alter seit der letzten Referenz, nicht seit dem Einlagern
- geht bei sequentielltem Zugriff in FIFO über



Ablauf - LRU

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
F1																				
F2																				
F3																				
F4																				
F5																				

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
C1																				
C2																				
C3																				
C4																				
C5																				

IDEE

- Verbesserung durch Berücksichtigung der letzten K Referenzierungszeitpunkte
- Bestimmung des mittleren Zeitabstands zwischen den letzten K Referenzen
- K-Distanz $b_t(p, K)$: "LRU-K-Alter"
 - Zeit t , Referenzierungsfolge r_1, r_2, \dots, r_t
 - $b_t(p, K)$ ist Rückwärtsdistanz von t zur K-ten Referenz

$$\bullet \quad b_t(p, K) = \begin{cases} g & \text{wenn } r_{t-g} \text{ die Seite } p \text{ zum } K\text{-ten Mal referenziert} \\ \infty & \text{wenn } p \text{ nicht mind. } K \text{ mal in } r_1, \dots, r_t \text{ vorkommt} \end{cases}$$

- Ersetzung der Seite p mit $b_t(p, K)$ ist maximal

BEWERTUNG

- berücksichtigt aktuelle Referenzierungen häufiger als ältere
- LRU-1 entspricht LRU
- typisch: LRU-2

Ablauf – LRU-2



	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
F1																				
F2																				
F3																				
F4																				
F5																				

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
C1																				
C2																				
C3																				
C4																				
C5																				

IDEA

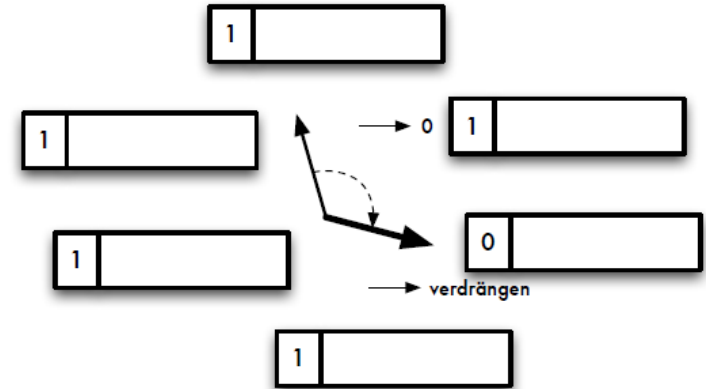
- Vereinfachung der benötigten Datenstrukturen durch Approximation
- Effektivität (Trefferquote) vs. Skalierbarkeit (Anzahl der Threads)

CLOCK: APPROXIMATION DER HISTORIE DURCH BIT-SCHIEBEREGISTER DER LÄNGE K

- $K = 0 \rightarrow$ FIFO
- $K = \text{unendlich} \rightarrow$ LRU
- Typisch: $K = 1 \rightarrow$ Second Chance

CLOCK

- Seite mit Benutzt-Bit; bei Referenzierung auf „1“ setzen
- Bei Seitenfehler:
 - Zyklische Suche
 - Seite mit „0“ verdrängen
 - sonst Setzen auf „0“ (zweite Chance)



Ablauf - CLOCK

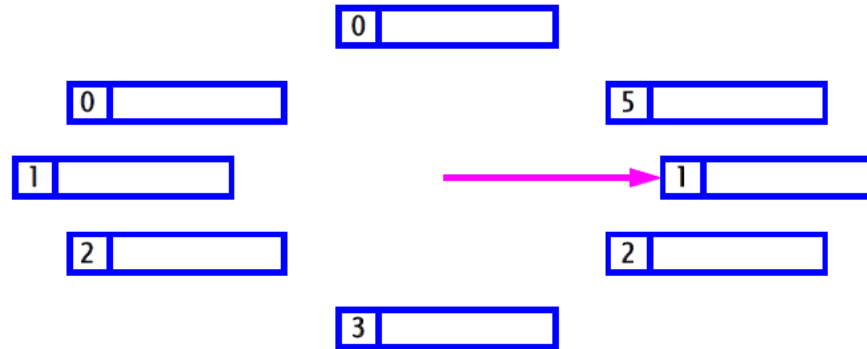
	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
F1																				
F2																				
F3																				
F4																				
F5																				

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
C1																				
C2																				
C3																				
C4																				
C5																				

GCLOCK (Generalized Clock)

BESCHREIBUNG

- Pro Seite wird Referenzzähler geführt (statt Bit)
- Ersetzung nur von Seiten mit Zählwert 0 (sonst Dekrementierung des Zählers und Betrachtung der nächsten Seite)



VERFAHRENSPARAMETER

- Initialwerte für Referenzzähler
- Wahl des Dekrementes
- Zählerinkrement bei erneuter Referenz

Ablauf - GCLOCK

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
F1																				
F2																				
F3																				
F4																				
F5																				

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
C1																				
C2																				
C3																				
C4																				
C5																				

Ablauf



	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
F1																				
F2																				
F3																				
F4																				
F5																				

	A	B	C	A	B	C	D	E	C	D	F	A	D	B	G	A	G	E	A	C
C1																				
C2																				
C3																				
C4																				
C5																				