

U+H  

# Implementierung von Datenbanksystemen

## Kapitel 3: Pufferverwaltung

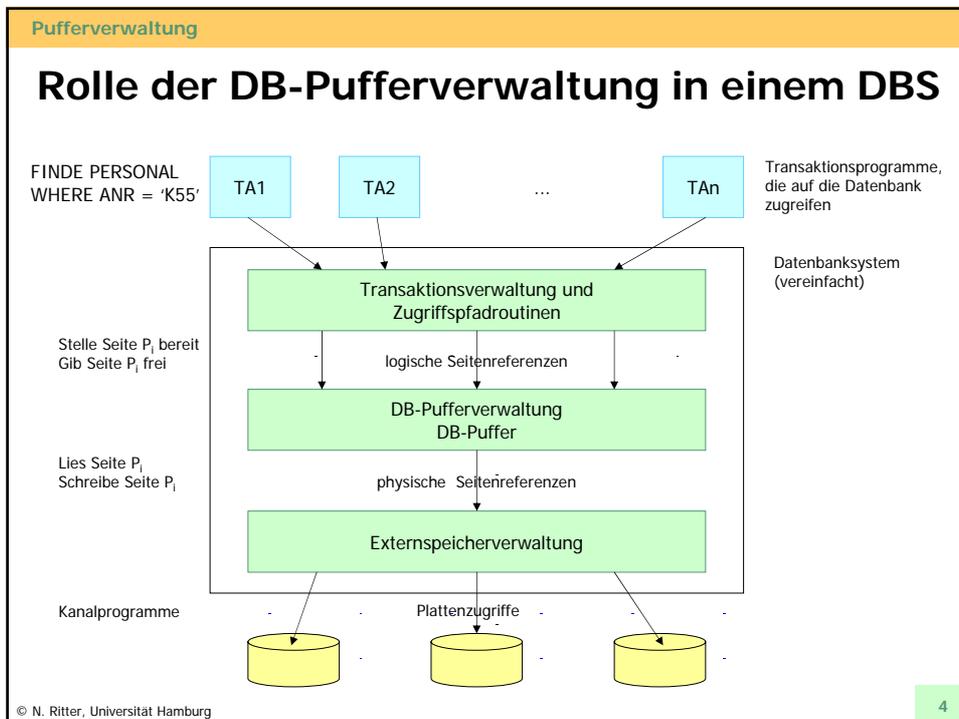
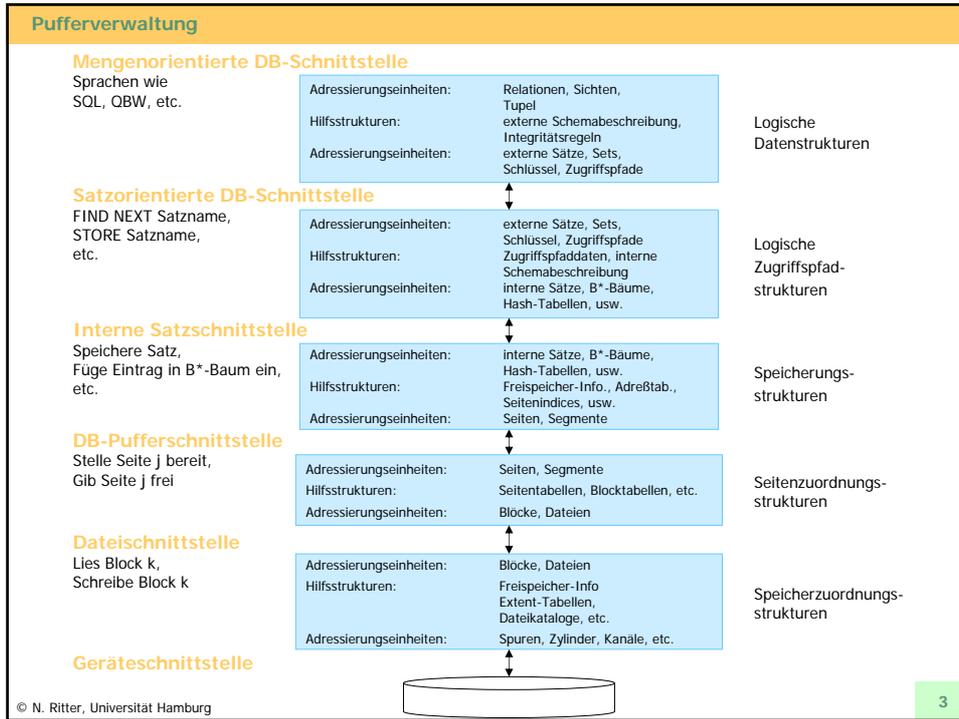
Teile dieses Foliensatzes beruhen auf ähnlichen Vorlesungen, gehalten von Prof. Dr. T. Härder am Fachbereich Informatik der Universität Kaiserlautern und Prof. Dr. B. Mitschang / Dr. Holger Schwarz am Fachbereich Informatik der Universität Stuttgart. Für das vorliegende Material verbleiben alle Rechte (insbesondere für den Nachdruck) bei den Autoren.

Pufferverwaltung

## Kapitel 3: Pufferverwaltung

- Allgemeine Charakteristika
  - Ablauf des Pufferzugriffs
  - logische und physische Seitenreferenzen
- Eigenschaften von DB-Referenzstrings
  - Seitenreferenzstrings
  - Lokalität, Sequentialität
  - LRU-Stacktiefenverteilung
  - Referenzdichte-Kurven
- Speicherzuteilung im Puffer
- Seiteneretzungsverfahren
  - Klassifikation von Ersetzungsverfahren
  - LRU, FIFO, CLOCK, GCLOCK, LRD ...
  - Leistungsanalyse von Ersetzungsverfahren
- Ersetzungsverfahren - Einbezug von Kontextwissen
  - 'Hot Set'-Modell
  - Prioritätsgesteuerte Seiteneretzung
- Pufferverwaltung bei Seiten variabler Größe
  - zusätzliche Anforderungen
  - VAR-PAGE-LRU
- Seiteneretzung bei virtuellem Speicher

© N. Ritter, Universität Hamburg 2



**Pufferverwaltung**

## Seitenreferenzstrings

- Jede Datenanforderung ist eine **logische Seitenreferenz**.

**Seitenreferenzstring (SRS)**  
 $R = \langle r_1, r_2, \dots, r_i, \dots, R_n \rangle$  mit  $r_i = (T_i, D_i, S_i)$

- $T_i$  zugreifende Transaktion
- $D_i$  referenzierte DB-Partition
- $S_i$  referenzierte DB-Seite

- Aufgabe der DB-Pufferverwaltung:
  - Minimierung der **physischen Seitenreferenzen**
- Bestimmung von Ausschnitten aus R bezüglich bestimmter Transaktionen, Transaktions-Typen und DB-Partitionen sinnvoll zur Analyse des Referenzverhaltens
- Wie kann Referenzstring-Information verwendet werden für
  - Charakterisierung des Referenzverhaltens?
  - Bestimmung von Lokalität und Sequentialität?
  - Unterstützung einer effektiven Seitenersetzung?

© N. Ritter, Universität Hamburg 5

**Pufferverwaltung**

## Vergleich mit BS-Funktionen

Wichtige Aspekte für den Datenbankpuffer	Bewertung der verfügbaren BS-Funktionalität
effizienter Pufferzugriff	Zugriff auf Dateipuffer ist teuer (supervisor call)
DB-spezifische Referenzmuster sollen unterstützt werden	BS-Ersetzungsverfahren sind z.B. nicht auf zyklisch sequentielle oder baumartige Zugriffsfolgen abgestimmt
In DBMS ist aufgrund von Seiteninhalten oder Referenzmustern eine Voraussage des Referenzverhaltens (z. B. bei Tabellen-Scans) möglich; Prefetching erzielt in solchen Fällen eine enorme Leistungssteigerung	Normale Dateisysteme bieten keine geeignete Schnittstelle für Prefetching
Selektives Ausschreiben von Seiten zu bestimmten Zeitpunkten (z. B. für Logging)	In existierenden Dateisystemen nicht immer möglich

**DBVS muss eigene Pufferverwaltung realisieren**

- Implementierung**
  - Ersetzungsalgorithmen im DB-Puffer in Software implementiert
  - Seitenersetzung in Adressräumen bei virtuellem Speicher ist Hardware-gestützt
- Seitenreferenz vs. Adressierung**
  - nach einem FIX-Aufruf kann eine DB-Seite mehrfach bis zum UNFIX referenziert werden
  - unterschiedliches Seitenreferenzverhalten
  - andere Ersetzungsverfahren?

© N. Ritter, Universität Hamburg 6

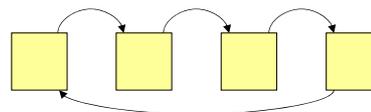
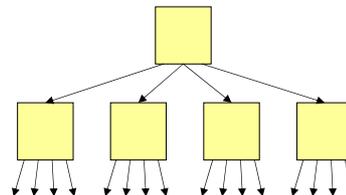
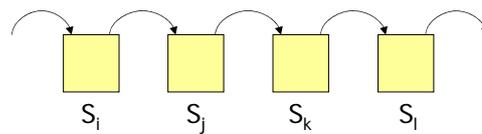
## Kapitel 3: Pufferverwaltung

- Allgemeine Charakteristika
  - Ablauf des Pufferzugriffs
  - logische und physische Seitenreferenzen
- Eigenschaften von DB-Referenzstrings
  - Seitenreferenzstrings
  - Lokalität, Sequentialität
  - LRU-Stacktiefenverteilung
  - Referenzdichte-Kurven
- Speicherzuteilung im Puffer
- Seiteneretzungsverfahren
  - Klassifikation von Eretzungsverfahren
  - LRU, FIFO, CLOCK, GCLOCK, LRD ...
  - Leistungsanalyse von Eretzungsverfahren
- Eretzungsverfahren - Einbezug von Kontextwissen
  - 'Hot Set'-Modell
  - Prioritätsgesteuerte Seiteneretzung
- Pufferverwaltung bei Seiten variabler Größe
  - zusätzliche Anforderungen
  - VAR-PAGE-LRU
- Seiteneretzung bei virtuellem Speicher

## Eigenschaften von Referenzstrings

Typische Referenzmuster in DBS

- Sequentielle Suche:
  - Bsp.: Durchsuchen ganzer Satztypen (Relationen)
- Hierarchische Pfade:
  - Bsp.: Suchen mit Hilfe von B\*-Bäumen
- Zyklische Pfade:
  - Bsp.: Abarbeiten von Sets ((1:n)-Beziehungen), Suchen in DBTT-/Datenseiten



## Sequentialität

- SRS weisen typischerweise Phasen von Sequentialität und Lokalität auf
- **Sequentielle Zugriffsfolge (SZ):**
  - Zwei aufeinander folgende Referenzen  $r_i$  und  $r_{i+1}$  gehören zu einer sequentiellen Zugriffsfolge, falls sie gleiche oder benachbarte DB-Seiten referenzieren
- Algorithmus
  - Seitenreferenzstring wird vollständig durchmustert; alternativ kann die Folge der ankommenden Referenzen analysiert werden
  - Solange obige Bedingung erfüllt ist, gehören alle aufeinander folgenden Referenzen zu einer SZ, sonst beginnt eine neue SZ

## Sequentialität

- Länge einer sequentiellen Zugriffsfolge (LSZ):
  - LSZ ist die Anzahl der verschiedenen in SZ referenzierten Seiten
  - Bsp.:  
Referenzstring **A A B B D E E F F H** enthält drei SZ:  
(**A A B B**)<sub>1</sub> mit LSZ(1) = 2  
(**D E E F F**)<sub>2</sub> mit LSZ(2) = 3  
(**H**)<sub>3</sub> mit LSZ(3) = 1
- Maß für Sequentialität:
  - Die kumulative Verteilung der SZ-Längen LSZ(i) wird berechnet  
 $S(x) = \Pr(\text{SZ-Länge} \leq x)$
  - Für obiges Beispiel gilt:  $S(1) = 0.33$ ,  $S(2) = 0.67$ ,  $S(3) = 1.0$
- Bei Sequentialität Optimierung durch (asynchrones) Prefetching von DB-Seiten möglich

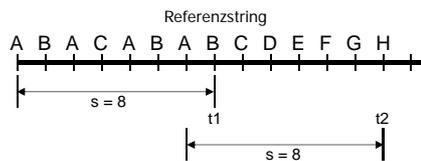
## Lokalität

- Erhöhte Wiederbenutzungswahrscheinlichkeit für zuletzt referenzierte Seiten (gradueller Begriff)
- Grundlegende Voraussetzung für
  - effektive DB-Pufferverwaltung (Seitenersetzung)
  - Einsatz von Speicherhierarchien
- Wie kann man Lokalität messen?
  - Working-Set-Modell
  - LRU-Stacktiefenverteilung

## Working-Set-Modell

- **Working-Set**  $W(t,s)$ : Seiten, die von der betrachteten Transaktion innerhalb der letzten  $s$  Referenzen (Fenstergröße) bezogen auf Zeitpunkt  $t$  angesprochen wurden.

- **Working-Set-Größe**  
 $w(t,s) = |W(t,s)|$



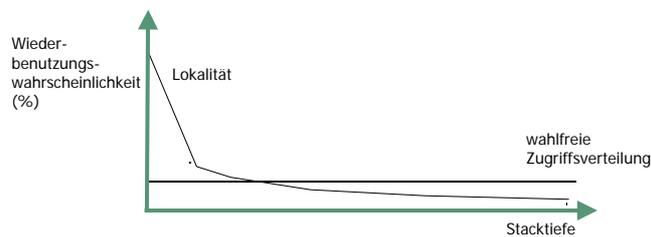
- Working sets
 

$W(t_1, 8) = \{A, B, C\}$	$W(t_2, 8) = \{A, B, C, D, E, F, G, H\}$
$w(t_1, 8) = 3$	$w(t_2, 8) = 8$

- Aktuelle Lokalität:  $AL(t, s) = \frac{w(t, s)}{s}$                       Mittlere Lokalität:  $L(s) = \frac{\sum_{t=1}^n AL(t, s)}{n}$   
 ( $n$  = Länge des Referenzstrings)

## LRU-Stacktiefenverteilung

- Wie lässt sich Lokalität charakterisieren?
  - LRU-Stacktiefenverteilung liefert Maß für die Lokalität (präziser als Working-Set-Ansatz)
  - LRU-Stack enthält alle bereits referenzierten Seiten in der Reihenfolge ihres Zugriffsalters
- Bestimmung der Stacktiefenverteilung:
  - pro Stackposition wird Zähler geführt
  - Rereferenz einer Seite führt zur Zählererhöhung für die jeweilige Stackposition
- Zählerwerte entsprechen der Wiederbenutzungshäufigkeit
  - Für LRU-Seitenersetzung kann aus der Stacktiefenverteilung für eine bestimmte Puffergröße unmittelbar die Trefferrate (bzw. Fehlseitenrate) bestimmt werden



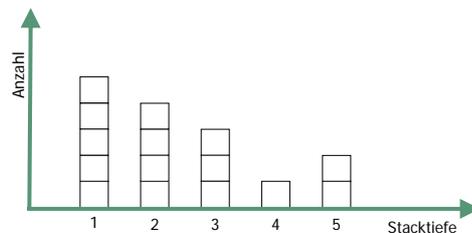
## Beispiel: Ermittlung der Stacktiefenverteilung

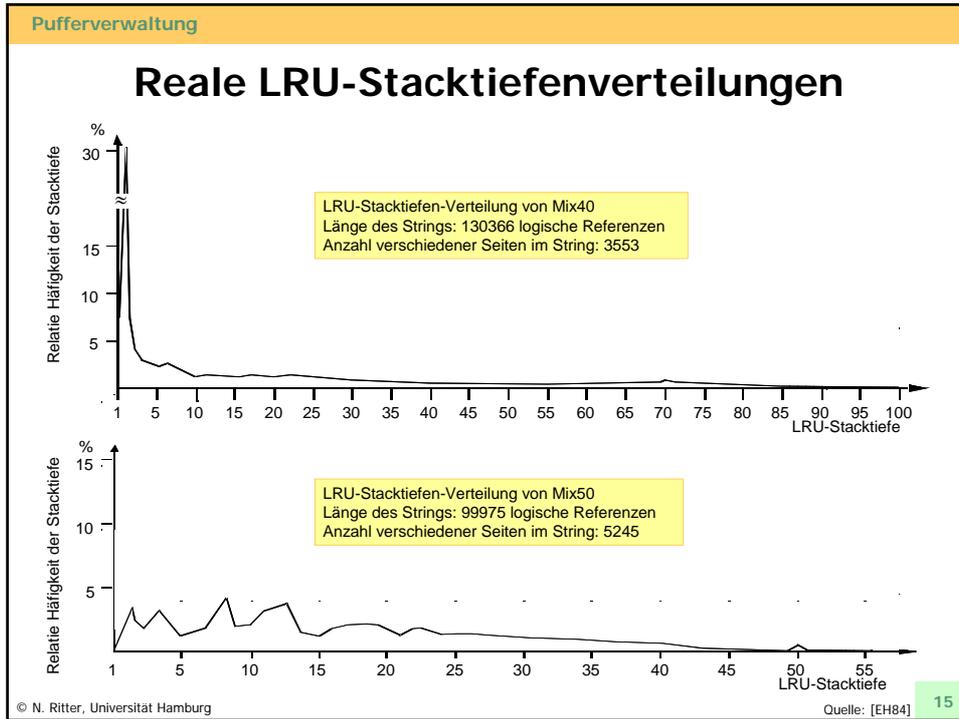
Referenzstring: A B A C A A A B B B C D E A E

LRU-Stack

1	A
2	B
3	C
4	D
5	E

Stacktiefenverteilung





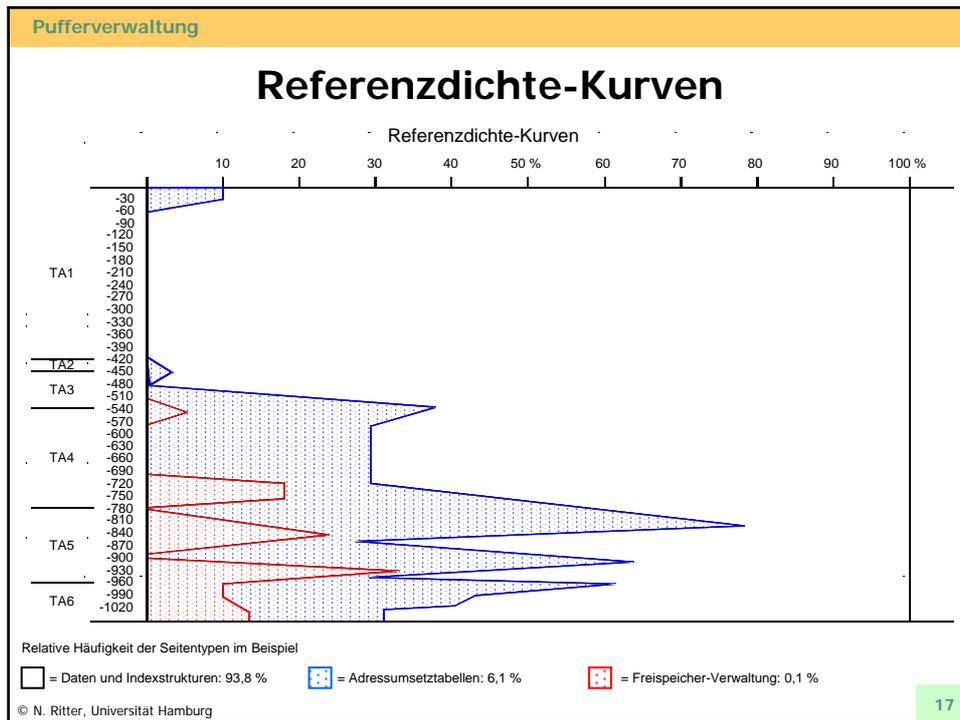
**Pufferverwaltung**

## Relative Referenzmatrix (DOA-Last)

ca. 17 500 Transaktionen, 1 Million Seitenreferenzen auf ca. 66 000 verschiedene Seiten

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	Total
TT1	9.1	3.5	3.3		5.0	0.9	0.4	0.1				0.0		22.3
TT2	7.5	6.9	0.4	2.6	0.0	0.5	0.8	1.0	0.3	0.2	0.0			20.3
TT3	6.4	1.3	2.8	0.0	2.6	0.2	0.7	0.1	1.1	0.4		0.0	0.0	15.6
TT4	0.0	3.4	0.3	6.8			0.6	0.4			0.0			11.6
TT5	3.1	4.1	0.4		0.0		0.5	0.0						8.2
TT6	2.4	2.5	0.6		0.7		0.9	0.3						7.4
TT7	1.3		2.6			2.3	0.1							6.2
TT8	0.3	2.3	0.2		0.0		0.1							2.9
TT9	0.0	1.4	0.0					1.1						2.6
TT10	0.3	0.1	0.3			1.0	0.1					0.0		1.8
TT11		0.9						0.2						1.1
TT12		0.1												0.1
<b>Total</b>	<b>30.3</b>	<b>26.6</b>	<b>11.0</b>	<b>9.4</b>	<b>8.3</b>	<b>4.9</b>	<b>4.1</b>	<b>3.3</b>	<b>1.4</b>	<b>0.6</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>	<b>100.0</b>
partition size (%)	31.3	6.3	8.3	17.8	1.0	20.8	2.6	7.3	2.6	1.3	0.8	0.0	0.0	100.0
% referenced	11.1	16.6	8.0	2.5	18.1	1.5	9.5	4.4	5.2	2.7	0.2	13.5	5.0	6.9

© N. Ritter, Universität Hamburg 16



**Pufferverwaltung**

## Kapitel 3: Pufferverwaltung

- Allgemeine Charakteristika
  - Ablauf des Pufferzugriffs
  - logische und physische Seitenreferenzen
- Eigenschaften von DB-Referenzstrings
  - Seitenreferenzstrings
  - Lokalität, Sequentialität
  - LRU-Stacktiefenverteilung
  - Referenzdichte-Kurven
- **Speicherzuteilung im Puffer**
- Seiteneretzungsverfahren
  - Klassifikation von Ersetzungsverfahren
  - LRU, FIFO, CLOCK, GCLOCK, LRD ...
  - Leistungsanalyse von Ersetzungsverfahren
- Ersetzungsverfahren - Einbezug von Kontextwissen
  - 'Hot Set'-Modell
  - Prioritätsgesteuerte Seiteneretzung
- Pufferverwaltung bei Seiten variabler Größe
  - zusätzliche Anforderungen
  - VAR-PAGE-LRU
- Seiteneretzung bei virtuellem Speicher

© N. Ritter, Universität Hamburg 18

Pufferverwaltung

## Speicherzuteilung im DB-Puffer

Speicherzuteilung im DB-Puffer

```

    graph TD
      Root[Speicherzuteilung im DB-Puffer] --> Global[global  
(ein gemeinsamer Pufferbereich)]
      Root --> Lokal[lokal /  
partitionierte Pufferbereiche]
      Lokal --> Statisch[statisch]
      Lokal --> Dynamisch[dynamisch]
      Statisch --> Gleichf[gleichförmige  
Partitionen]
      Statisch --> Angepasst[angepasste  
Partitionen]
    
```

- Partitionierungsmöglichkeiten:
  - eigener Pufferbereich pro Transaktion
  - TA-Typ-bezogene Pufferbereiche
  - Seitentyp-bezogene Pufferbereiche
  - DB-(Partitions)spezifische Pufferbereiche

19

© N. Ritter, Universität Hamburg

Pufferverwaltung

## Dynamische Pufferallokation: Working-Set-Ansatz (WS)

- Pro Pufferpartition P soll Working-Set im Puffer bleiben; Seiten, die nicht zum Working-Set gehören, können ersetzt werden
- Bei Fehlseitenbedingung muss Working-Set bekannt sein, um Ersetzungskandidat zu bestimmen
  - Fenstergröße pro Partition:  $s(P)$
  - Referenzzähler pro Partition:  $RZ(P)$
  - letzter Referenzzeitpunkt für Seite  $i$ :  $LRZ(P, i)$
  - ersetzbar sind solche Seiten, für die gilt:  $RZ(P) - LRZ(P, i) > s(P)$
- Fenstergröße kritischer Parameter → Thrashing-Gefahr

P1: A A C A A G H A

P2: B D E E F F

Referenz-string:

20

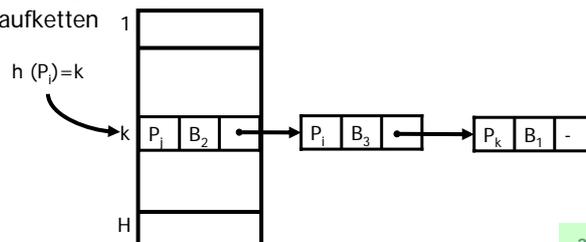
© N. Ritter, Universität Hamburg

## Dynamische Pufferallokation: Page-Fault-Frequency-Ansatz (PFF)

- Vorgabe einer Soll-Fehlseitenrate  $F$
- Bei Fehlseitenbedingung wird aktuelle Fehlseitenrate  $FA(P)$  ausgewertet:
  - gilt  $FA(P) > F$  wird Partition um eine Seite erweitert
  - gilt  $FA(P) < F$  wird eine Seite abgegeben
- Verfahren entspricht einer speziellen Working-Set-Strategie mit Fenstergröße  $w' = 1/F$ .
- Falls für den Abstand  $a$  zwischen zwei Fehlseitenbedingungen
  - $a > w'$  gilt:
    - wird das Working-Set beibehalten
    - Seiten, die vor der letzten Fehlseitenbedingung referenziert wurden, werden freigegeben
  - $a < w'$  gilt:
    - werden zusätzliche Seitenrahmen bereitgestellt

## Suche im DB-Puffer

- Direktes, sequentielles Durchsuchen der Pufferrahmen
  - sehr hoher Suchaufwand
  - Gefahr vieler Paging-Fehler bei virtuellem Speicher
- Nutzung von Hilfsstrukturen (Eintrag pro Pufferrahmen)
  - unsortierte oder sortierte Tabelle
  - Tabelle mit verketteten Einträgen
    - geringere Änderungskosten
    - Anordnung in LRU-Reihenfolge möglich
  - Suchbäume (z.B. AVL-, m-Weg-Bäume)
  - Hash-Tabelle mit Überlaufketten
    - beste Lösung



**Pufferverwaltung**

## Kapitel 3: Pufferverwaltung

- Allgemeine Charakteristika
  - Ablauf des Pufferzugriffs
  - logische und physische Seitenreferenzen
- Eigenschaften von DB-Referenzstrings
  - Seitenreferenzstrings
  - Lokalität, Sequentialität
  - LRU-Stacktiefenverteilung
  - Referenzdichte-Kurven
- Speicherzuteilung im Puffer
- Seiteneretzungsverfahren
  - Klassifikation von Ersetzungsverfahren
  - LRU, FIFO, CLOCK, GCLOCK, LRD ...
  - Leistungsanalyse von Ersetzungsverfahren
- Ersetzungsverfahren - Einbezug von Kontextwissen
  - 'Hot Set'-Modell
  - Prioritätsgesteuerte Seiteneretzung
- Pufferverwaltung bei Seiten variabler Größe
  - zusätzliche Anforderungen
  - VAR-PAGE-LRU
- Seiteneretzung bei virtuellem Speicher

© N. Ritter, Universität Hamburg 23

**Pufferverwaltung**

## Seiteneretzungsverfahren

- Klassifikation:
 

**Verfahrensklassen**

```

graph TD
    A[Verfahrensklassen] --> B[preplanning]
    A --> C[prefetching]
    A --> D[demand fetching]
            
```

<p><b>preplanning</b></p> <p>Programmanalyse, Vorabuntersuchung des Datenbedarfs</p> <p>große Fehlrate, ungenauere Obermengen</p>	<p><b>prefetching</b></p> <p>physische Daten- strukturierung, Clusterbildung, Verarbeitungswissen</p> <p>datenmodellbezogen (hierarchisch), spekulative Entscheidungen</p>	<p><b>demand fetching</b></p> <p>keine Vorausaktionen</p> <p>Lokalitätserhaltung im DB-Puffer</p>
---	--	---
- Grundannahme bei Ersetzungsverfahren:
 

A B A C A B C A B B C D ...

Referenzen

← jüngerste Vergangenheit | nächste Zukunft →

Referenzverhalten ähnlich

© N. Ritter, Universität Hamburg 24

**Pufferverwaltung**

## Referenzverhalten und Ersetzungsverfahren

- Referenzverhalten in DBS
  - typischerweise hohe Lokalität: Optimierung durch Ersetzungsverfahren
  - manchmal Sequentialität oder zufällige Arbeitslast (RANDOM-Referenzen)
  - Grenzfälle des Referenzverhaltens und der Ersetzungsverfahren zeigen Optimierungsmöglichkeiten auf.
- Prinzipielle Zusammenhänge, welche die Fehlseitenrate bestimmen

		Referenz	
		Random	Lokalität
Ersetzung	Random	R/R	L/R
	OPT	R/OPT	L/OPT

D = DB-Größe in Blöcken  
N = # Rahmen im DB-Puffer

© N. Ritter, Universität Hamburg 25

**Pufferverwaltung**

## Behandlung geänderter Seiten im DB-Puffer

- Ersetzung einer geänderten Seite erfordert ihr vorheriges (synchrones) Zurückschreiben in die DB
  - Antwortzeitverschlechterung
- Abhängigkeit von der gewählten Ausschreibstrategie:

FORCE	NOFORCE
Alle Änderungen einer Transaktion werden spätestens beim EOT in die DB zurückgeschrieben („write-through“).	Kein Durchschreiben der Änderungen bei EOT (verzögertes Ausschreiben, „deferred write-back“).
+ i.A. stets ungeänderte Seiten zur Ersetzung vorhanden	+ Seite kann mehrfach geändert werden, bevor ein Ausschreiben erfolgt (geringerer E/A-Overhead, bessere Antwortzeiten)
+ vereinfachte Recovery (nach Rechnerausfall sind alle Änderungen beendeter TA bereits in die DB eingebracht)	+ Vorausschauendes (asynchrones) Ausschreiben geänderter Seiten erlaubt auch bei NOFORCE, vorwiegend ungeänderte Seiten zu ersetzen
- hoher E/A-Overhead	- aufwändigere Recovery nach Rechnerausfall
- starke Antwortzeiterhöhung für Änderungstransaktionen	

Synchrone DB-Schreibvorgänge lassen sich weitgehend vermeiden

© N. Ritter, Universität Hamburg 26

Pufferverwaltung

### Kriterien für die Auswahl der zu ersetzenden Pufferseite

Verfahren	Alter	letzte Referenz	Referenzhäufigkeit	andere Kriterien
OPT	-	-	-	Vorauswissen
RANDOM	-	-	-	---
LFU	-	-	X	---
FIFO	X	-	-	---
LRU	-	X	-	---
CLOCK	X	(x)	-	---
GCLOCK	X	(x)	(x)	---
LRD(V1)	X	(x)	X	---
LRD(V2)	(x)	X	X	---
LRU-K	-	X	X	---

Pufferverwaltung

### Least Frequently Used (LFU)

- Pro Seite wird ein Referenzzähler (RZ) geführt (statt Bit)
  - Initialisierung mit 1 bei der Einlagerung der Seite
  - Erhöhung um 1 bei jeder Referenz
- Fehlseitenbedingung:
  - Ersetzung der Seite mit der geringsten Referenzhäufigkeit
- Alter einer Seite wird nicht berücksichtigt!

RZ	
2	
4	
1	
3	
3	
6	
1	
3	

**Pufferverwaltung**

## First-In First-Out (FIFO)

- Die älteste Seite im DB-Puffer wird ersetzt
- Referenzen während des Pufferaufenthaltes werden nicht berücksichtigt
- Nur für strikt sequentielles Referenzierungsverhalten geeignet

© N. Ritter, Universität Hamburg 29

**Pufferverwaltung**

## Least Recently Used (LRU)

- Beispiel (Puffergröße 4):
  - Referenz der Seite C
- Referenz der Seite E

LRU-Stack

- Unterscheidung zwischen *Least Recently Referenced* und *Least Recently Unfixed*

© N. Ritter, Universität Hamburg 30



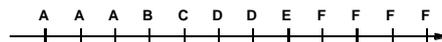
## Least Reference Density (LRD): Variante 1

- Algorithmus
  - Wenn eine Seite ersetzt werden muss, wird die Referenzdichte aller Seiten im DB-Puffer bestimmt
  - Referenzdichte = Referenzhäufigkeit in einem bestimmten Referenzintervall
  - Ersetzungskandidat ist Seite mit geringster Referenzdichte
- Variante 1: Referenzintervall entspricht Alter einer Seite
- Berechnung der Referenzdichte:
  - Globaler Zähler (GZ): Gesamtanzahl aller Referenzen
  - Einlagerungszeitpunkt (EZ): GZ-Wert beim Einlesen der Seite
  - Referenzzähler (RZ)

Referenzdichte:

$$RD(j) = \frac{RZ(j)}{GZ - EZ(j)}$$

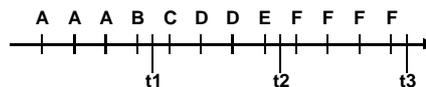
- Beispiel:



	RZ	EZ	RD
A			
B			
C			
D			
E			
F			

## Least Reference Density (LRD): Variante 2

- Variante 2: konstante Intervallgröße
  - Künstliches Altern von Seiten: Ältere Referenzen werden bei der Bestimmung der Referenzdichte geringer bewertet
  - Periodisches Reduzieren der Referenzzähler, um Gewicht früher Referenzen zu reduzieren
  - Reduzierung von RZ durch Division oder Subtraktion:



	t1	t2	t3
RZ(A)			
RZ(B)			
RZ(C)			
RZ(D)			
RZ(E)			
RZ(F)			

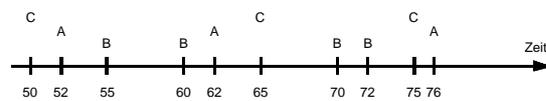
$$RZ(i) = \frac{RZ(i)}{K1} \quad (K1 > 1)$$

oder

$$RZ(i) = \begin{cases} RZ(i) - K2 & \text{falls } RZ(i) - K2 \geq K3 \\ K3 & \text{sonst } (K2 > 0, K3 \geq 0) \end{cases}$$

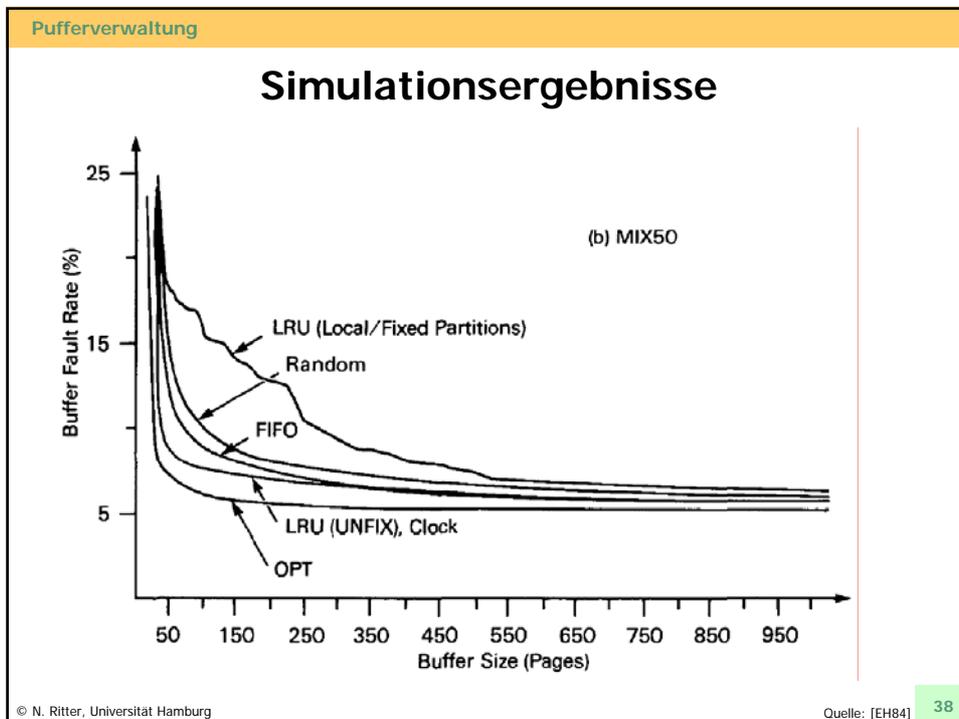
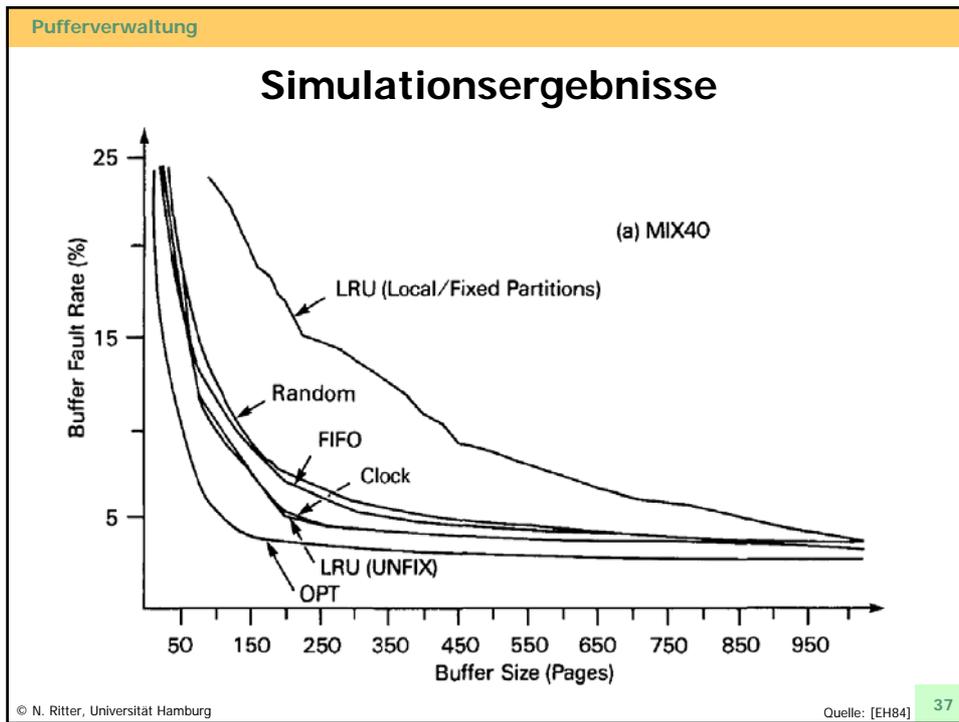
## LRU-K

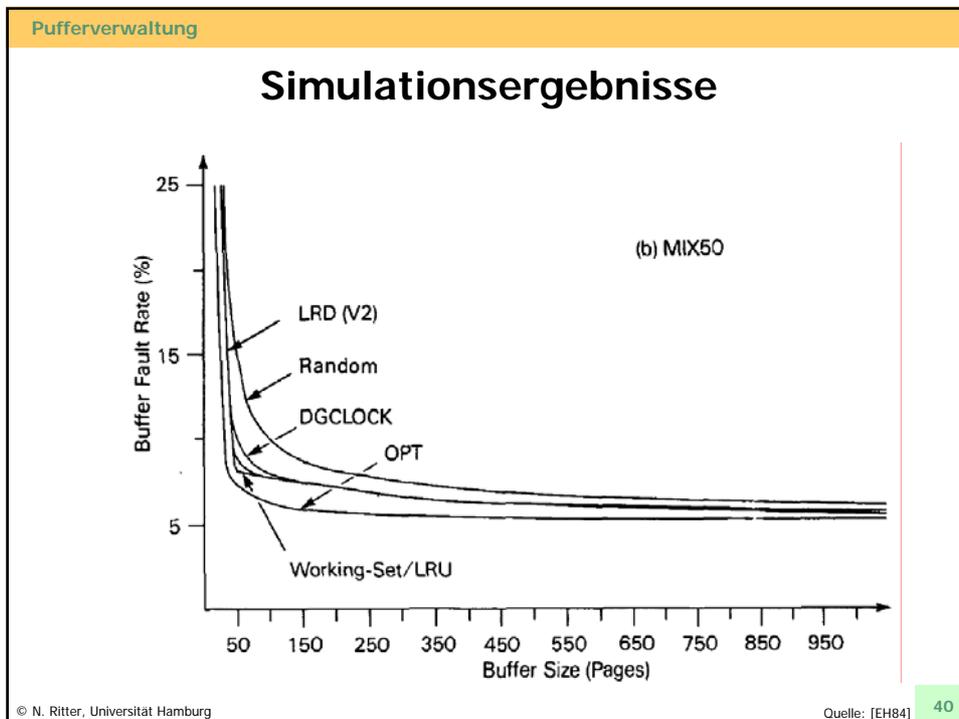
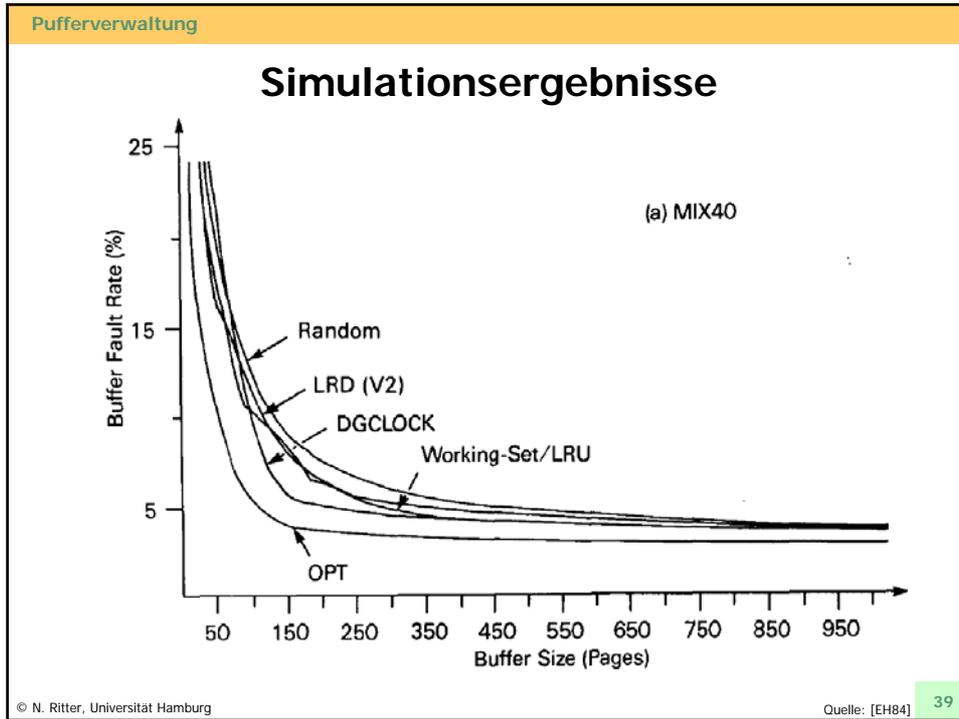
- Aufzeichnung der K letzten Referenzzeitpunkte (pro Seite im DB-Puffer)
  - Gegeben sei bis zum Betrachtungszeitpunkt  $t$  der Referenzstring  $r_1, r_2, \dots, r_t$ .
  - Rückwärtige K-Distanz  $b_t(P, K)$  ist die in Referenzen gemessene Distanz rückwärts bis zur K-jüngsten Referenz auf Seite P:
  - $b_t(P, K) = x$ , wenn  $r_{t-x}$  den Wert P besitzt und es genau K-1 andere Werte  $i$  mit  $t-x < i \leq t$  mit  $r_i = P$  gab.
  - $b_t(P, K) = \infty$ , wenn P nicht wenigstens K mal in  $r_1, r_2, \dots, r_t$  referenziert wurde
- Beispiel



## LRU-K

- Merkmale:
  - Aufwändigere Aufzeichnung
  - Methode benötigt kein explizites „Altern“
  - Bei Ersetzung werden die  $b_t(P_i, K)$  der Pufferseiten benötigt! Sonderbehandlung für Seiten mit  $b_t(P, K) = \infty$
- Wie hängt LRU-K mit LRD zusammen? Approximation der Referenzdichte?
- LRU-2 (d.h.  $K=2$ ) stellt i.A. die beste Lösung dar
  - ähnlich gute Ergebnisse wie für  $K > 2$ , jedoch einfachere Realisierung
  - schnellere Reaktion auf Referenzschwankungen als bei größeren K





**Pufferverwaltung**

## Simulationsergebnisse

		MIX40	MIX50	
Charakteristika der DB, der Transaktionslast und der logischen Seitenreferenzstrings	Total number of pages (school-DB)	30,000		
	Number of different pages in the string	3,553	5,245	
	Number of logical references	130,366	99,975	
	Number of page modifications	9,378	2,865	
	Number of pages being fixed*	maximum	11	10
		average	4.61	6.26
	Percentage of references with FIX duration	= 1	71%	41%
		2-10	22%	41%
		>10	7%	18%
	FIX duration (in logical references)	max.	1,786	—
avg.		4.62	6.26	
Percentage of pages of a given type	FPA	0.1%		
	DBTT	6.1%		
	USER	93.8%		
Percentage of references to page-types	FPA	0.9%	0.1%	
	DBTT	9.4%	21.7%	
	USER	89.7%	78.2%	
Percentage of references to most frequently referenced pages ("hot spot" pages)	1.	12.6%	3.3%	
	2.	9.8%	0.5%	
	3.	4.8%	0.4%	
Relative frequency distribution of references to the other pages	>0.9%	5	1	
	0.9%–0.1%	195	293	
	0.1%–0.03%	1,606	2,741	
	<0.03%	1,747	2,208	
Number of references to shared pages (concurrently fixed)*		1,175	359	
cold start buffer fault rate in %		2.72	5.24	
number of executed transactions	total	262	39	
	parallel: max.	8	8	
	avg.	6.31	6.29	

© N. Ritter, Universität Hamburg \* measured with a buffer size of 128 pages

**Pufferverwaltung**

## Kapitel 3: Pufferverwaltung

- Allgemeine Charakteristika
  - Ablauf des Pufferzugriffs
  - logische und physische Seitenreferenzen
- Eigenschaften von DB-Referenzstrings
  - Seitenreferenzstrings
  - Lokalität, Sequentialität
  - LRU-Stacktiefenverteilung
  - Referenzdichte-Kurven
- Speicherzuteilung im Puffer
- Seiteneretzungsverfahren
  - Klassifikation von Ersetzungsverfahren
  - LRU, FIFO, CLOCK, GCLOCK, LRD ...
  - Leistungsanalyse von Ersetzungsverfahren
- Ersetzungsverfahren - Einbezug von Kontextwissen
  - 'Hot Set'-Modell
  - Prioritätsgesteuerte Seiteneretzung
- Pufferverwaltung bei Seiten variabler Größe
  - zusätzliche Anforderungen
  - VAR-PAGE-LRU
- Seiteneretzung bei virtuellem Speicher

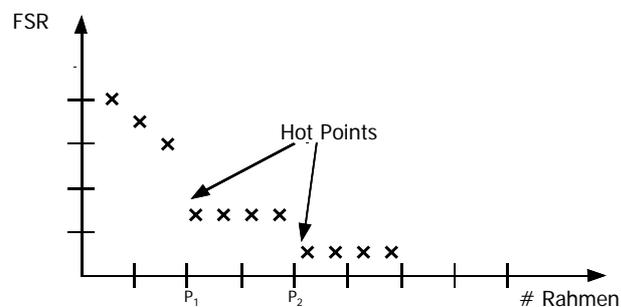
© N. Ritter, Universität Hamburg 42

## Probleme bei LRU-ähnlichen Verfahren

- Internes Thrashing:
  - Zyklisches Referenzieren (Loop) einer Menge von Seiten mit  $\# \text{Seiten} > \# \text{Rahmen}$
  - Auswirkung: Häufigkeit der Seitenersetzung steigt stark an
- Externes Thrashing:
  - $T_1$ : sehr schnelle Seitenanforderungen (wahlfrei oder sequentiell)
  - $T_2$ : langsame Seitenanforderungen für eine Seitenmenge ( $\# \text{Seiten} < \# \text{Rahmen}$ ), die zyklisch referenziert wird
  - Auswirkung: Seiten der  $T_1$  werden auch bei hoher Referenzlokalität verdrängt
- WS-Modell:
  - Es wird versucht,  $w(t,s)$  Rahmen zu allokkieren
  - Bei sequentiellem Scan hätte ein Rahmen genügt

## Hot-Set-Modell

- Ausnutzung von Kontextwissen bei mengenorientierten Anforderungen
  - Verbesserung in relationalen DBS möglich
- Hot Set: Menge der Seiten im Referenzzyklus
- Hot Point: abrupte Veränderung in der FSR, z. B. verursacht durch Schleife beim Verbund
- Prinzipieller Verlauf der Fehlseitenrate (FSR) bei der Verarbeitung von Hot Sets



Pufferverwaltung

## Hot-Set-Modell

- Hot Set Size (HSS):
  - größter Hot Point kleiner als der verfügbare DB-Puffer
- Anfrage-Optimierer berechnet HSS für die verschiedenen Zugriffspläne (Abschätzung der #Rahmen)
- Beispiel:
  - ...
  - ...
  - ...
  - ...
  - ...

# Rahmen	Kosteneinheiten/10 <sup>3</sup> (PERS in äußerer Schleife)	Kosteneinheiten/10 <sup>3</sup> (ABT in äußerer Schleife)
0	50	30
10	5	30
20	5	15

```
SELECT *
FROM   ABT X, PERS Y
WHERE  X.ANR = Y.ANR
AND ...
```

— PERS in äußerer Schleife  
- - - ABT in äußerer Schleife  
 Index-Scan für beide Relationen

- Anwendungscharakteristika:
  - Berücksichtigung der HSS in den Gesamtkosten
  - Auswahl abhängig von verfügbarer DB-Puffergröße
  - Bindung zur Laufzeit möglich

© N. Ritter, Universität Hamburg 45

Pufferverwaltung

## Ersetzungsverfahren – Einbezug von Kontextwissen

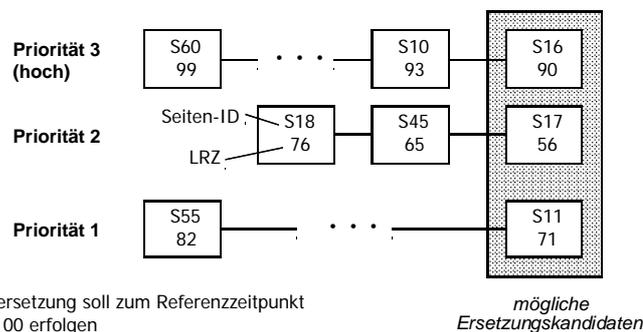
- Zugriffspläne durch Anfrage-Optimierer
  - Zugriffscharakteristik/Menge der referenzierten Seiten kann bei der Erstellung von Plänen vorausgesagt/abgeschätzt werden
  - Zugriffsmuster enthält immer Zyklen/Loops (mindestens Kontrollseite – Datenseite, nested loop join etc.)
  - Kostenvoranschläge für Zugriffspläne können verfügbare Rahmen berücksichtigen
  - Bei Ausführung wird die Mindestrahmenzahl der Pufferverwaltung mitgeteilt

© N. Ritter, Universität Hamburg 46

## Prioritätsgesteuerte Seitenersetzung

- Bevorzugung bestimmter Transaktionstypen/DB-Partitionen vielfach wünschenswert (z.B. um Benachteiligungen durch sehr lange TA oder sequentielle Zugriffe zu vermeiden)
- Berücksichtigung von Prioritäten bei der Pufferverwaltung
- Verfahren PRIORITY-LRU:
  - pro Prioritätsstufe eigene dynamische Pufferpartition
  - LRU-Kette pro Partition
  - Priorität einer Seite bestimmt durch DB-Partition bzw. durch (maximale) Priorität referenzierender Transaktionen
  - ersetzt wird Seite aus der Partition mit der geringsten Priorität. Ausnahme: die  $w$  zuletzt referenzierten Seiten sollen (unabhängig von ihrer Priorität) nicht ersetzt werden
- Kompromiss zwischen Prioritäts- und absolutem LRU-Kriterium möglich

## Prioritätsgesteuerte Seitenersetzung



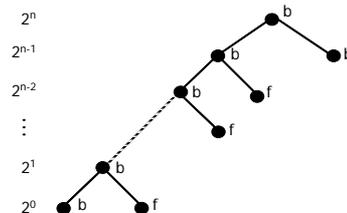
- Seitenersetzung soll zum Referenzzeitpunkt  $RZ = 100$  erfolgen
- $w = 30$  zuletzt referenzierten Seiten dürfen nicht ersetzt werden
- LRZ = Zeitpunkt der letzten Referenz einer Seite

## Kapitel 3: Pufferverwaltung

- Allgemeine Charakteristika
  - Ablauf des Pufferzugriffs
  - logische und physische Seitenreferenzen
- Eigenschaften von DB-Referenzstrings
  - Seitenreferenzstrings
  - Lokalität, Sequentialität
  - LRU-Stacktiefenverteilung
  - Referenzdichte-Kurven
- Speicherzuteilung im Puffer
- Seiteneretzungsverfahren
  - Klassifikation von Ersetzungsverfahren
  - LRU, FIFO, CLOCK, GCLOCK, LRD ...
  - Leistungsanalyse von Ersetzungsverfahren
- Ersetzungsverfahren - Einbezug von Kontextwissen
  - 'Hot Set'-Modell
  - Prioritätsgesteuerte Seiteneretzung
- Pufferverwaltung bei Seiten variabler Größe
  - zusätzliche Anforderungen
  - VAR-PAGE-LRU
- Seiteneretzung bei virtuellem Speicher

## Pufferverwaltung bei Seiten variabler Größe

- Seitengröße:
  - keine beliebigen Seitengrößen (Fragmentierung, Abbildung auf Externspeicher)
  - Seitenlänge (SL) als Vielfaches eines Einheitsrasters (Transporteinheit, Rahmengröße im Puffer)
  - Beispiel:  
 $SL = 1, 2, 4, \dots, 2^n$  • Rahmengröße ( $n \leq 8$ )
- Trennung von Seite und Rahmen (Rastergröße)
- Schnittstellenforderung: Zusammenhängende Speicherung der Seite im Puffer
- Buddy-System (BS-Algorithmus)
  - (Verwaltung variabler Bereiche: frei(f) / belegt(b))
  - festes Raster, hierarchischer Vergabemechanismus
  - Suche nur in freien Bereichen (belegte Bereiche können nicht verschoben werden)
  - Zusammenfassung von freien Neffen aufwendig



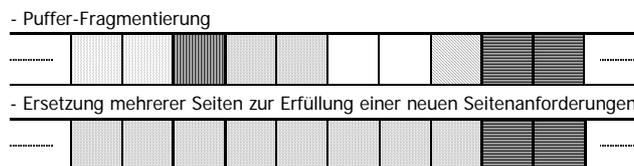
## Seiten variabler Größe - Zusätzliche Anforderungen

- Zustände von Seiten/Rahmen
  - frei: keine Seite vorhanden
  - unfixed: Seite ist ersetzbar/verschiebbar
  - fixed: Seite muss Pufferadresse behalten
- Vermeidung von Seitenersetzungen
  - Umlagern von Seiten mit Unfix-Vermerk und "hoher" Wiederbenutzungswahrscheinlichkeit
- Suche nach "bestem" Ersetzungskandidaten
- Was heißt "bestem" Ersetzungskandidat?
  - Wiederbenutzungs-Wahrscheinlichkeit: Anzahl der Referenzen, Alter, letzte Referenz einer Seite
  - Fragmentierung/Lückenbenutzung: first fit, best fit
  - Rahmeninhalte: Anteil ersetzbarer und freier Rahmen
  - Anzahl der zu ersetzenden Seiten zur Platzbeschaffung für eine neue Seite
- Kombination von Ersetzung und Umlagern von Seiten
  - sehr komplexe Entscheidungssituation
- Alternative: Partitionierung des DB-Puffers für Seiten gleicher Größe (und anwendungsabhängige Ersetzungsverfahren)

## Seiten variabler Größe - Pufferverwaltung



### Probleme:



- Ziele:
  - Maximierung der Pufferbelegung (Speicherplatzoptimierung)
  - Minimierung der E/A durch Berücksichtigung von Lokalität im Referenzverhalten
- Vorschlag für einen Algorithmus: VAR-PAGE-LRU
- Neue Aufgabenstellung: Caching und spekulatives Prefetching von Dokumenten in Speicherhierarchien (Tertiärspeicher)

Pufferverwaltung

## Kapitel 3: Pufferverwaltung

- Allgemeine Charakteristika
  - Ablauf des Pufferzugriffs
  - logische und physische Seitenreferenzen
- Eigenschaften von DB-Referenzstrings
  - Seitenreferenzstrings
  - Lokalität, Sequentialität
  - LRU-Stacktiefenverteilung
  - Referenzdichte-Kurven
- Speicherzuteilung im Puffer
- Seiteneretzungsverfahren
  - Klassifikation von Ersetzungsverfahren
  - LRU, FIFO, CLOCK, GCLOCK, LRD ...
  - Leistungsanalyse von Ersetzungsverfahren
- Ersetzungsverfahren - Einbezug von Kontextwissen
  - 'Hot Set'-Modell
  - Prioritätsgesteuerte Seiteneretzung
- Pufferverwaltung bei Seiten variabler Größe
  - zusätzliche Anforderungen
  - VAR-PAGE-LRU
- Seiteneretzung bei virtuellem Speicher

© N. Ritter, Universität Hamburg 53

Pufferverwaltung

## Seiteneretzung bei virtuellem Speicher

- Page Fault:
  - P<sub>1</sub> in SP virtuell, aber nicht in SP real (HSP)
- Database Fault:
  - P<sub>2</sub> nicht in SP virtuell, Seitenrahmen R<sub>2</sub> für P<sub>2</sub> jedoch in SP real
- Double Page Fault:
  - P<sub>3</sub> nicht in SP virtuell, ausgewählter Seitenrahmen R<sub>3</sub> für P<sub>3</sub> nicht in SP real

© N. Ritter, Universität Hamburg 54

## Zusammenfassung

- Referenzmuster in DBS sind Mischformen
  - sequentielle, zyklische, wahlfreie Zugriffe
  - Lokalität innerhalb und zwischen Transaktionen
  - „bekannte“ Seiten mit hoher Referenzdichte
  - Erkennen von Scan-basierter Verarbeitung
- Ohne Lokalität ist jede Optimierung der Seitenersetzung sinnlos (~ RANDOM)
- Suche im Puffer durch Hash-Verfahren
- Speicherezuteilung:
  - global ⇒ alle Pufferrahmen für alle Transaktionen (Einfachheit, Stabilität, ...)
  - lokal ⇒ Sonderbehandlung bestimmter TAs/Anfragen/DB-Bereiche
- Behandlung geänderter Seiten: NOFORCE, asynchrones Ausschreiben
- Seiteneretzungsverfahren
  - Prefetching und Pipelining mit Wechsellpuffer
  - „zu genaue“ Verfahren sind schwierig einzustellen (⇒ instabil)
  - Nutzung mehrerer Kriterien: Alter, letzte Referenz, Referenzhäufigkeit
  - CLOCK – LRU, aber einfachere Implementierung
  - GCLOCK, LRD, LRU-K relativ komplex
  - LRU-2 guter Kompromiss; vorletzter Referenzzeitpunkt bestimmt „Opfer“
- Erweiterte Ersetzungsverfahren
  - Nutzung von Zugriffsinformationen des Anfrage-Optimierers
  - Hot Set Model
  - Einsatz vieler Puffer zur Separierung von Lasten verschiedenen Typs, optimiert für spezifische Datentypen
- Double-Paging sollte vermieden werden

## Literatur zu diesem Kapitel

- [EH84] W. Effelsberg, T. Härder: Principles of Database Buffer Management. In: ACM TODS Vol. 9, No. 4, 1984.
- [JC+90] R. Jauhari, M.J. Carey, M. Livny: Priority-Hints: An Algorithm for Priority-Based Buffer Management. In: Proc. 16th VLDB Conf., 1990.
- [OO+93] E. J. O’Neil, P. E. O’Neil, G. Weikum: The LRU-K Page Replacement Algorithm for Database Disk Buffering. In: Proc. ACM SIGMOD Conf. Washington. D.C., 1993.
- [Sik88] A. Sikeler: VAR-PAGE-LRU—A Buffer Replacement Algorithm Supporting Different Page Sizes. In: Proc. Extending Database Technology, 1988.
- [KW98] A. Kraiss, G. Weikum: Integrated Document Caching and Prefetching in Storage Hierarchies Based on Markov-Chain Predictions. VLDB Journal, 1998.