



**Hasso
Plattner
Institut**

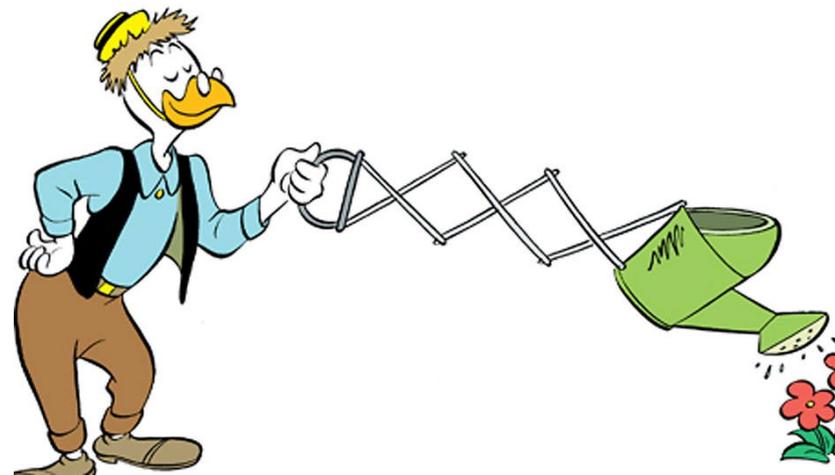
IT Systems Engineering | Universität Potsdam

Übung Datenbanksysteme II

Anfrageoptimierung I

Leon Bornemann

Folien basierend auf
Maximilian Jenders,
Thorsten Papenbrock



2

1. Nachbesprechung Hausaufgabe 3
2. Hybrid-Hashjoin
3. Postgresql Einführung
4. Algebraische Transformation
5. Kardinalitätsschätzung

3

1. Nachbesprechung Hausaufgabe 3
2. Hybrid-Hashjoin
3. Postgresql Einführung
4. Algebraische Transformation
5. Kardinalitätsschätzung

Hausaufgabe 4: Aufgabe 1

Strategie	R	R, a<50	R, a=50	R, 50<a<100
Full Scan	$10^6/B = 10^5$	$10^6/B = 10^5$	$10^6/B = 10^5$	$10^6/B = 10^5$
B ⁺ -Baum	$(h-1)+10^6/M+10^6$	$(h-1)+50/M+50$	$(h-1)+1+1$	$(h-1)+50/M+49$
Hash-Index	$10^6/(N*c)+10^6$	50+50	1+1	49+49

- B: #Tupel/Block (=10)
- M: #Schlüssel-Pointer-Pare/Blatt-Block (>10)
- h: Höhe des B⁺-Baums
- N: #Daten-Pointer/Hash-Block (>10)
- c: Füllgrad eines Hash-Blocks

▪ Warum?

Hausaufgabe 3: Bemerkungen

5

- Tupel und Blockkosten nicht vermischen!
- Bei Annahmen auf Sinnhaftigkeit achten
 - Man braucht nur zwei (wie gezeigt)
 - Bei mehr Annahmen jede Begründen!

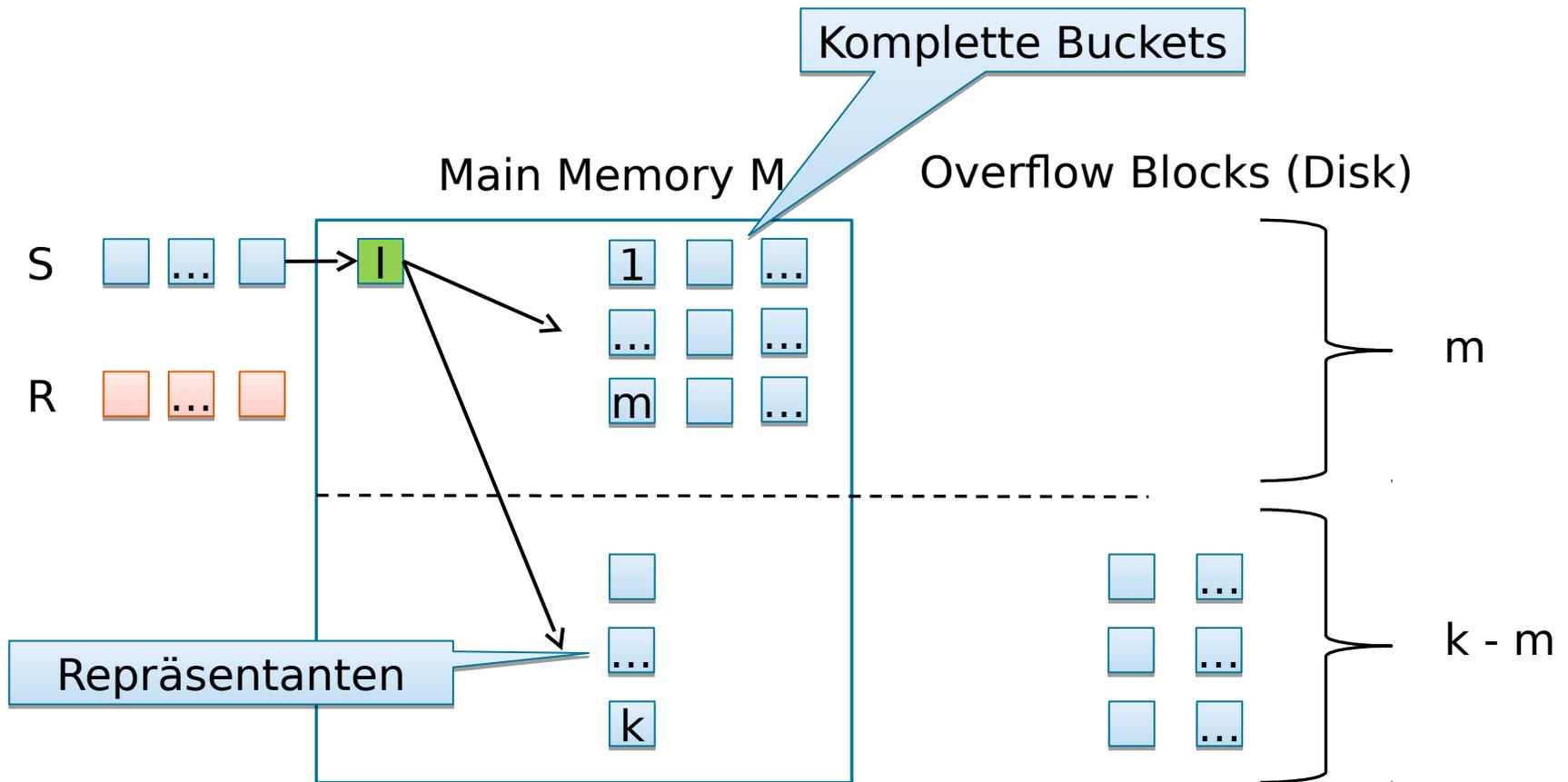
6

1. Nachbesprechung Hausaufgabe 3
- 2. Hybrid-Hashjoin**
3. Postgresql Einführung
4. Algebraische Transformation
5. Kardinalitätsschätzung

Wiederholung: Hybrid Hashjoin

7

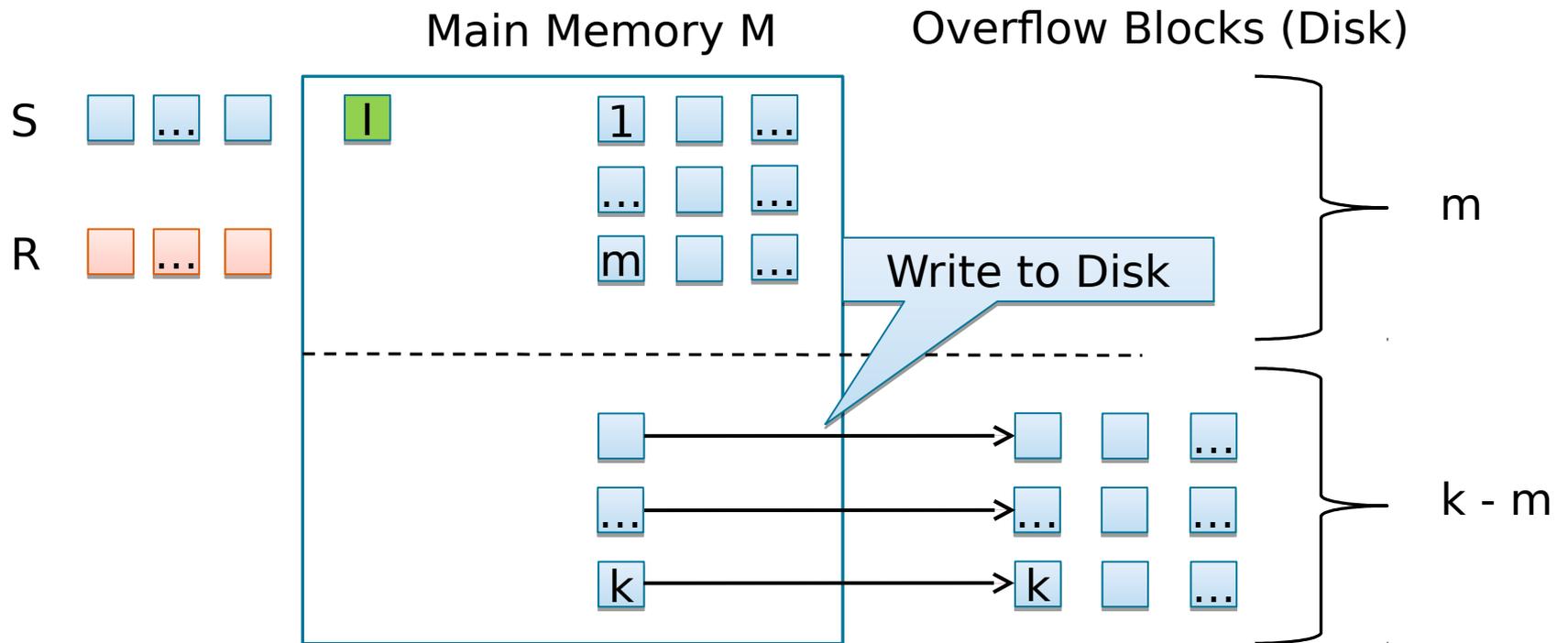
- Hashe S in k Buckets aber behalte m Buckets komplett in M



Wiederholung: Hybrid Hashjoin

8

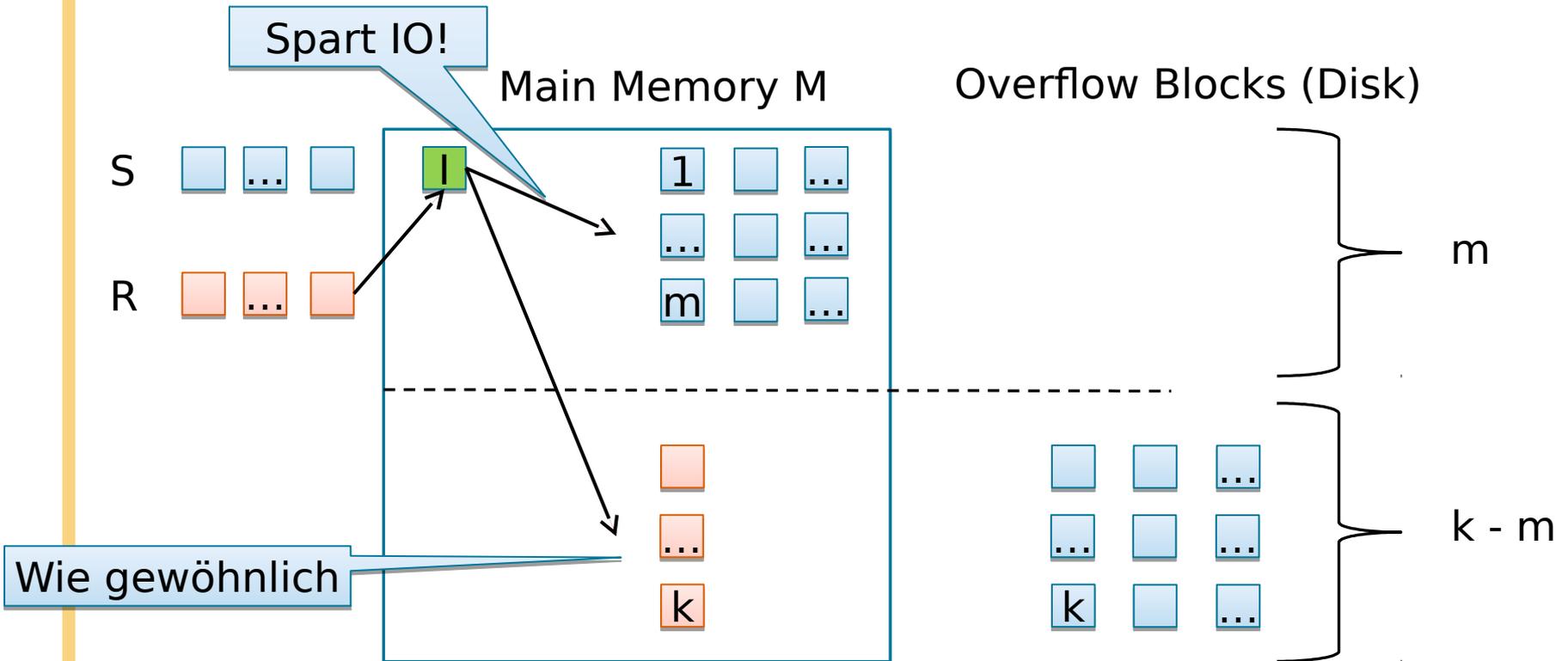
- Hashe S in k Buckets aber behalte m Buckets komplett in M



Wiederholung: Hybrid Hashjoin

9

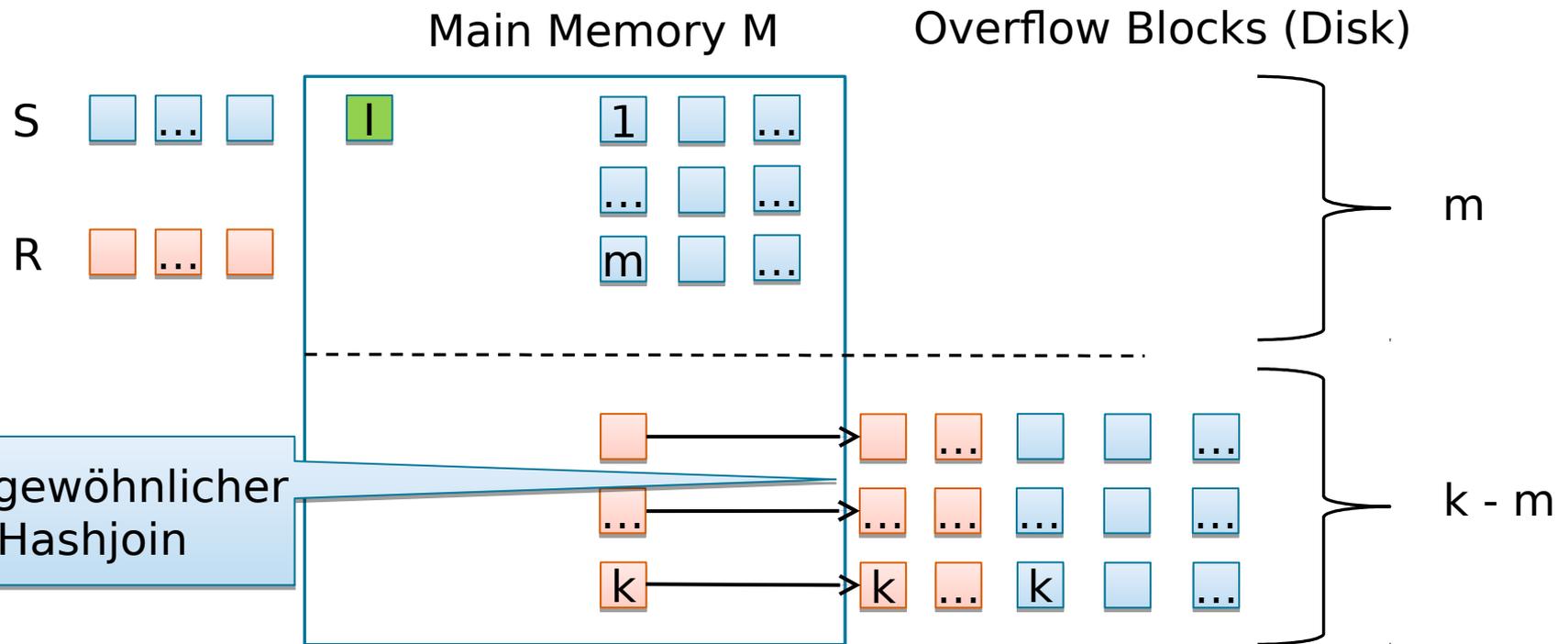
- Hashe S in k Buckets aber behalte m Buckets komplett in M
- Joinpartner aus R für die ersten m Blöcke brauchen keine zusätzliche IO



Wiederholung: Hybrid Hashjoin

10

- Hashe S in k Buckets aber behalte m Buckets komplett in M
- Joinpartner aus R für die ersten m Blöcke brauchen keine zusätzliche IO



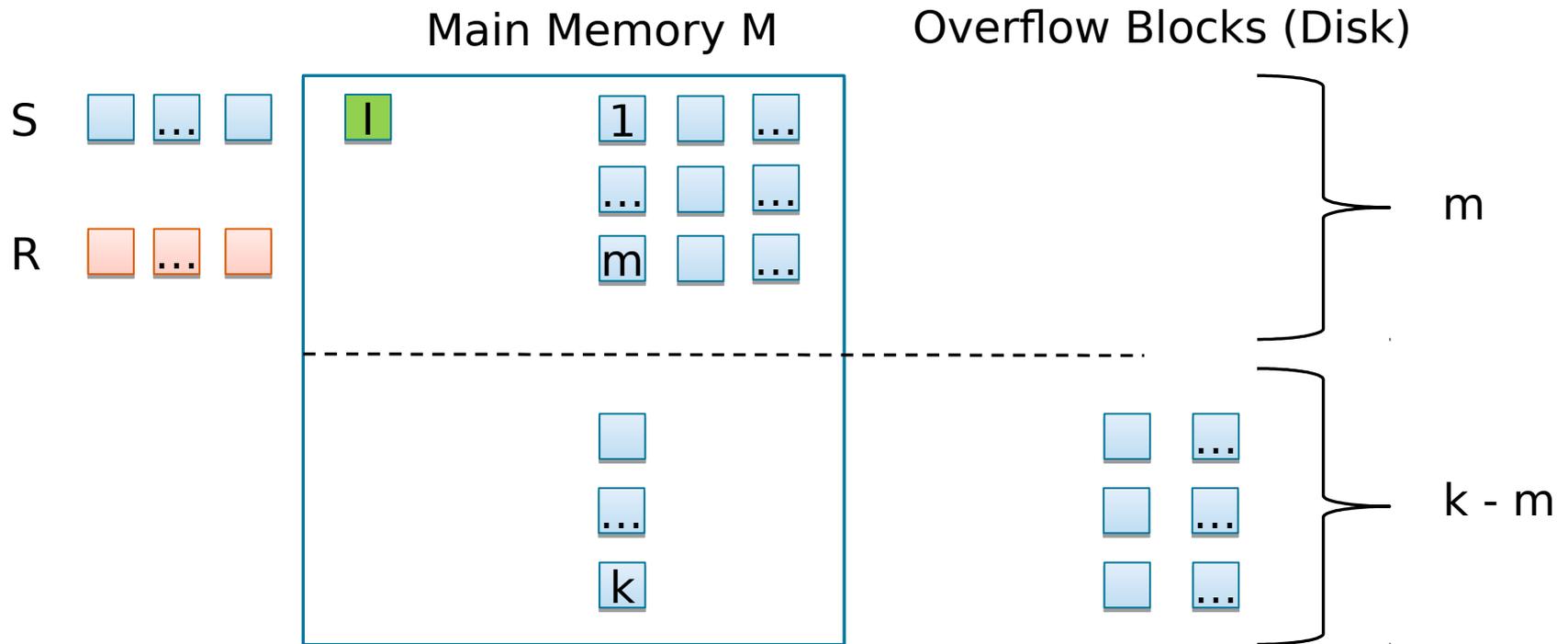
Wiederholung: Hybrid Hashjoin

11

Vollst. Buckets

- Hashe S in k Buckets ab und behalte m Buckets komplett in M
- Es muss gelten $1 + (m \cdot \frac{B(S)}{k}) + 1 \cdot (k - m) \leq M$
- \rightarrow Einsparung: $2 \cdot \frac{m}{k} \cdot (B(S) + B(R)) \rightarrow$ maximiere $\frac{m}{k}$

Repräsentanten



Wiederholung: Hybrid Hashjoin

12

- Hashe S in k Buckets aber behalte m Buckets komplett in M
- Es muss gelten $1 + (m \cdot \frac{B(S)}{k}) + 1 \cdot (k - m) \leq M$
- \rightarrow Einsparung: $2 \cdot \frac{m}{k} \cdot (B(S) + B(R)) \rightarrow$ maximiere $\frac{m}{k}$
- \rightarrow Setze m auf 1, minimiere k \rightarrow wenige, große Buckets
- Sei BG die Bucketgröße $\rightarrow BG = \frac{B(S)}{k}$, es gilt:
- *Einsparung* = $2 \cdot \frac{m}{k} \cdot (B(S) + B(R)) = 2 \cdot \frac{BG}{B(S)} \cdot (B(S) + B(R))$
- $IO = (3 - 2 \cdot \frac{BG}{B(S)}) \cdot (B(S) + B(R))$

1. Nachbesprechung Hausaufgabe 3
2. Hybrid-Hashjoin
- 3. Postgresql Einführung**
4. Algebraische Transformation
5. Kardinalitätsschätzung

Hausaufgabe 4: Vorbemerkungen

14

- Code bitte nicht als PDF abgeben :)
 - Bemerkung auf dem Zettel wird angepasst
- Bekannte Schnittstelle einhalten
- Wenn ihr Änderungen an bestehenden Klassen macht, bitte genau überlegen, ob ihr diese wirklich braucht.
- Die Postgresql Resultate können je nach Version deutlich unterschiedliche sein. Das ist nicht weiter dramatisch. Schreibt einfach was in eurer Version das Ergebnis war.

PostgreSQL Einführung

15

- Wer hat Postgresql bereits installiert?
 - Tutorials sind verlinkt
 - Tech support über die Feiertage ist sehr unwahrscheinlich :)
 - Erstellen von Login, Roles und anderem Administrativem:
 - Diverse Tutorials findet ihr Online

PostgreSQL Einführung

16

- Eure Quellen
 - Offizielle Dokumentation: <https://www.postgresql.org/docs/9.6/static/index.html>
 - Stackoverflow
- Bemerkungen zur Dokumentation:
 - Beginnt mit allgemeiner Syntax, endet mit Beispielen
 - Create Table: <https://www.postgresql.org/docs/8.2/static/sql-createtable.html>
 - Beginnt mit heranzuführender Erklärung
 - Explain: <https://www.postgresql.org/docs/9.6/static/using-explain.html>

```
testdb=# select * from actors where name LIKE '%Bale, Christian' LIMIT 100;
```

- Ausgabe im Terminal, beenden mit 'q'
- Explain gibt QEP aus:

```
testdb=# explain select name,description from actors, plots where plots.title = actors.title
testdb=# AND name = 'Bale, Christian';
                                QUERY PLAN
-----
Nested Loop (cost=0.98..418.93 rows=31 width=583)
->  Index Only Scan using actors_pkey1 on actors (cost=0.56..106.28 rows=37 width=42)
    Index Cond: (name = 'Bale, Christian'::text)
->  Index Scan using title_author on plots (cost=0.42..8.42 rows=1 width=593)
    Index Cond: ((title)::text = (actors.title)::text)
(1 rows)
```

Geschätzte Startup Kosten

Geschätzte Größe eines Tupels in Bytes

Innerstes wird zuerst ausgeführt (von oben nach unten)

Geschätzte Kosten (willkürliche Einheit)

Geschätzte Größe

PostgreSQL Einführung

18

- Ausführung der Query, Ausgabe der tatsächlichen Zeiten/Größen
- $\text{Loops} * \text{rows} = \text{Gesamtgröße}$

```
testdb=# explain analyze select name,description from actors,plots where plots.title = actors.title
AND name = 'Bale, Christian';
                                QUERY PLAN
-----
Nested Loop (cost=0.98..418.93 rows=31 width=583) (actual time=0.257..2.949 rows=103 loops=1)
->  Index Only Scan using actors_pkey1 on actors (cost=0.56..106.28 rows=37 width=42) (actual time=0.181..0.256 rows=83 loops=1)
    Index Cond: (name = 'Bale, Christian'::text)
    Heap Fetches: 83
->  Index Scan using title_author on plots (cost=0.42..8.44 rows=1 width=593) (actual time=0.029..0.031 rows=1 loops=83)
    Index Cond: ((title)::text = (actors.title)::text)
Planning time: 1.975 ms
Execution time: 3.019 ms
(8 rows)
```

Tatsächliche Größe

Wie oft wurde die Operation ausgeführt

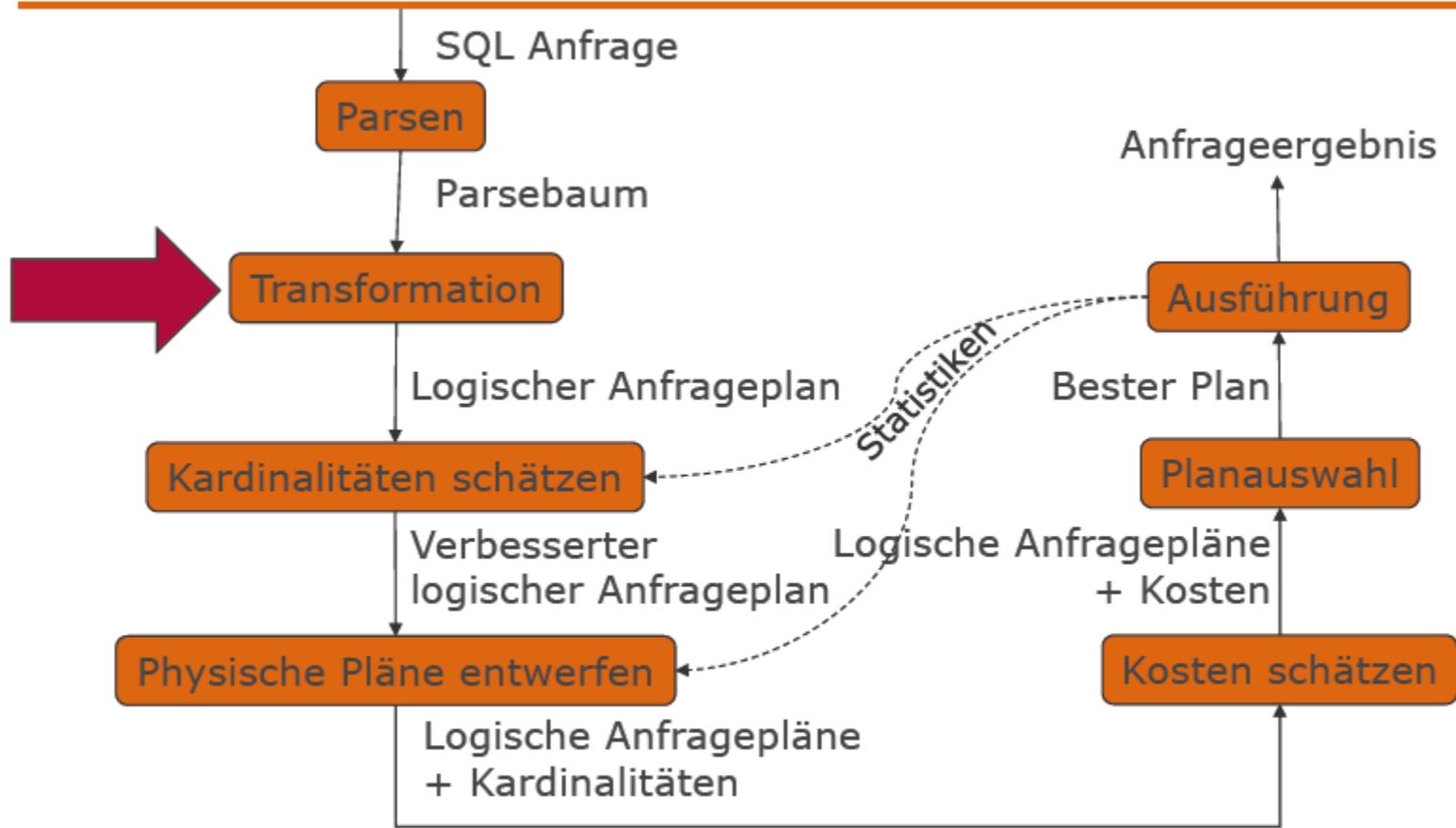
Time und Cost sind nicht direkt vergleichbar!

1. Nachbesprechung Hausaufgabe 3
2. Hybrid-Hashjoin
3. Postgresql Einführung
- 4. Algebraische Transformation**
5. Kardinalitätsschätzung

Recap: Algebraische Transformation

Ablauf der Anfragebearbeitung

20



Recap: Algebraische Transformation

21

Anfragebearbeitung – Transformationsregeln

- Transformation der internen Darstellung
 - Ohne Semantik zu verändern
 - Zur effizienteren Ausführung
 - Insbesondere: Kleine Zwischenergebnisse
- Äquivalente Ausdrücke
 - Zwei Ausdrücke der relationalen Algebra heißen äquivalent, falls
 - Gleiche Operanden (= Relationen)
 - Stets gleiche Antwortrelation
 - Stets?

Stets = Für jede mögliche
Instanz der Datenbank

Recap: Algebraische Transformation

22

- \times ist kommutativ und assoziativ
 - $R \times S = S \times R$
 - $(R \times S) \times T = R \times (S \times T)$
- \cup ist kommutativ und assoziativ
 - $R \cup S = S \cup R$
 - $(R \cup S) \cup T = R \cup (S \cup T)$
- \cap ist kommutativ und assoziativ
 - $R \cap S = S \cap R$
 - $(R \cap S) \cap T = R \cap (S \cap T)$
- \bowtie ist kommutativ und assoziativ
 - $R \bowtie S = S \bowtie R$
 - $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$

Gilt jeweils für Mengen
und Multimengen

Ausdrücke können in beide
Richtungen verwendet werden.

Weitere Regeln

Selektion

- $\sigma_{c_1 \text{ AND } c_2}(R) = \sigma_{c_1}(\sigma_{c_2}(R))$
- $\sigma_{c_1 \text{ OR } c_2}(R) = \sigma_{c_1}(R) \cup \sigma_{c_2}(R)$
 - Nicht bei Multimengen
- $\sigma_{c_1}(\sigma_{c_2}(R)) = \sigma_{c_2}(\sigma_{c_1}(R))$
- $\sigma_c(R \Phi S) \equiv (\sigma_c(R)) \Phi (\sigma_c(S))$
 - $\Phi \in \{\cup, \cap, -, \bowtie\}$
- $\sigma_c(R \Phi S) \equiv (\sigma_c(R)) \Phi S$
 - $\Phi \in \{\cup, \cap, -, \bowtie\}$
 - Falls sich c nur auf Attribute in R bezieht.

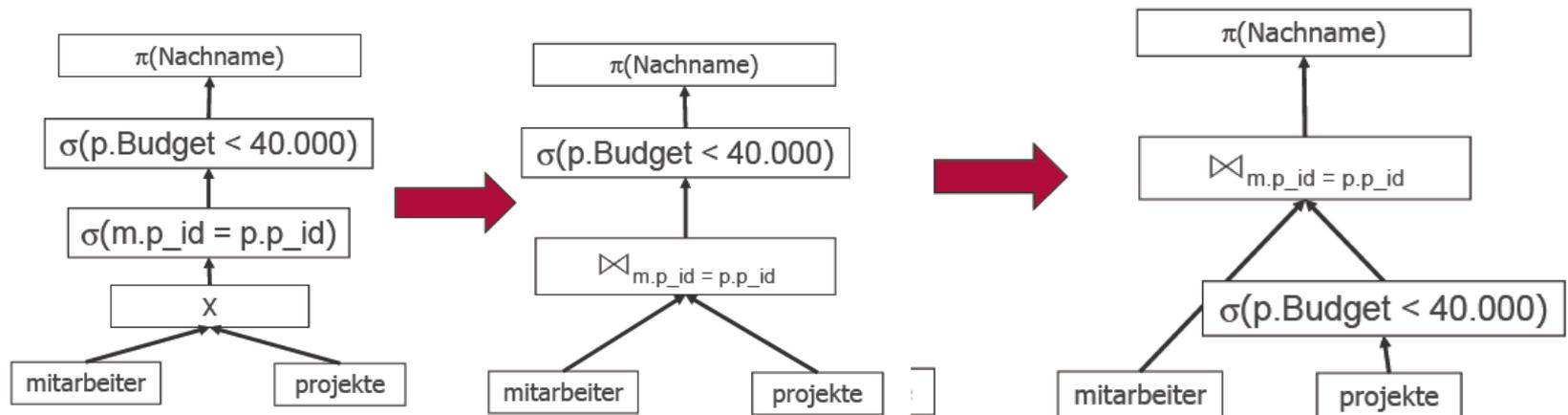
Projektion

- $\pi_L(R \bowtie S) = \pi_L(\pi_M(R) \bowtie \pi_N(S))$
- $\pi_L(R \bowtie_C S) = \pi_L(\pi_M(R) \bowtie_C \pi_N(S))$
- $\pi_L(R \times S) = \pi_L(\pi_M(R) \times \pi_N(S))$
- $\pi_L(\sigma_C(R)) = \pi_L(\sigma_C(\pi_M(R)))$

Recap: Algebraische Transformation

24

Anfragebearbeitung – Beispiel



- Gegeben: $R(a,b,c)$ und $S(c,d,e)$
- Gesucht: Kostengünstigste Anfragepläne für folgende Anfragen

d.h. möglichst kleine Zwischenergebnisse, also Selektionen und Projektionen so früh wie möglich

$$\text{a. } \sigma_{b=3 \wedge e=4 \wedge c>10} (R \bowtie S) \quad \rightarrow \quad \sigma_{c>10 \wedge b=3} (R) \bowtie \sigma_{c>10 \wedge e=4} (S)$$

$$\text{b. } \pi_{a,d} (R \bowtie S)$$

$$\rightarrow \pi_{a,d} (\pi_{a,c} (R) \bowtie \pi_{c,d} (S))$$

1. Nachbesprechung Hausaufgabe 3
2. Hybrid-Hashjoin
3. Postgresql Einführung
4. Algebraische Transformation
- 5. Kardinalitätsschätzung**

Kosten von Operationen - Zwischenergebnisse

- Wesentliches Kostenmerkmal: Anzahl der Tupel im Input
 - Insbesondere: Passt die Relation in den Hauptspeicher?
 - Selektion, Projektion, Sortierung, Join

- Output ist Input des nächsten Operators.

- Deshalb: Ein Kostenmodell schätzt u.a. für jede Operation die Anzahl der Ausgabebetupel.
 - „Selektivität“ in Bezug auf Inputgröße
 - $\# \text{Ausgabebetupel} = \# \text{Eingabetupel} \times \text{Selektivität}$
 - Auch „Selektivitätsfaktor“ (*selectivity factor, sf*)

28 Kostenschätzung – Selektion

- Anzahl Tupel sinkt, Tupelgröße bleibt
- $S = \sigma_{A=c}(R)$
 - $T(S) = T(R)/V(R,A)$
 - Reminder: $V(R,A)$ = Anzahl der *distinct* Werte in Spalte A
 - D.h. selectivity factor ist $1/V(R,A)$
 - Annahme: Werte sind gleichverteilt
 - Annahme: c ist einer dieser Werte
 - Bessere Abschätzung mittels Histogramme
- $S = \sigma_{A<c}(R)$
 - Erste Abschätzung: $T(S) = T(R) / 2$
 - Typischer: $T(S) = T(R) / 3$
- $S = \sigma_{A \neq c}(R)$
 - Erste Abschätzung: $T(S) = T(R)$
 - Etwas genauer: $T(S) = T(R) \cdot (V(R,A) - 1) / V(R,A)$
- Bei Konjunktionen mehrerer Selektionsbedingungen: Multiplikation der Selektivitätsfaktoren
 - Annahme: Unabhängigkeit der Bedingungen

Recap: Kardinalitätsschätzung

Kostenschätzung – Selektion mit Disjunktion

- $S = \sigma_{C1 \text{ OR } C2}(R)$
- Idee 1: Summe der Ergebniskardinalitäten
 - Annahme: Kein Tupel erfüllt beide Bedingungen
 - Problem: $T(S) > T(R)$
- Idee 2: $\text{MIN}[T(R), \text{Summe der Ergebniskardinalitäten}]$
- Idee 3: Wahrscheinlichkeitstheorie
 - Annahme: C1 und C2 sind unabhängig
 - $T(R) = n$ und $T(\sigma_{C1}(R)) = m_1$ und $T(\sigma_{C2}(R)) = m_2$

$$\square \Rightarrow T(S) = n \left(1 - \left(1 - \frac{m_1}{n}\right) \left(1 - \frac{m_2}{n}\right)\right)$$

Anteil Tupel, die nicht
C1 entsprechen

Anteil Tupel, die nicht
C2 entsprechen

Einfacher:

Sei $T(R,X)$ die (geschätzte) Anzahl Tupel, die X erfüllen

$$\rightarrow T(S) = T(R,C1) + T(R,C2) - T(R,C1 \text{ AND } C2)$$

- Annahme: Kein Tupel erfüllt beide Bedingungen
- Problem: $T(S) > T(R)$
- Idee 2: $\text{MIN}[T(R), \text{Summe der Ergebnisse}]$
- Idee 3: Wahrscheinlichkeitstheorie
- Annahme: $C1$ und $C2$ sind unabhängig
- $T(R) = n$ und $T(\sigma_{C1}(R)) = m_1$ und $T(\sigma_{C2}(R)) = m_2$

$$\square \Rightarrow T(S) = n \left(1 - \left(1 - \frac{m_1}{n}\right) \left(1 - \frac{m_2}{n}\right)\right)$$

Anteil Tupel, die nicht
C1 entsprechen

Anteil Tupel, die nicht
C2 entsprechen

Kardinalitätsschätzung

31

- Gegeben: $R(a,b,c,d)$ und $S(d,e)$
 $T(R)=100$; $V(R,a)=100$; $V(R,b)=10$; $V(R,c)=1$; $V(R,d)=50$
 $T(S)=500$; $V(S,d)=30$; $V(S,e)=100$

- Gesucht: Geschätzte Ergebniskardinalität für folgende Anfragen

a. $\sigma_{b=25}(R)$

$$T(R)/V(R,b) = 10$$

b. $\sigma_{c=30}(R)$

$$T(R)/V(R,c) = 100$$

c. $\sigma_{b=25 \wedge c=30}(R)$

$$T(R)/(V(R,b) \cdot V(R,c)) = 10$$

d. $\sigma_{b>25}(R)$

$$T(R)/3 \approx 33$$

e. $\sigma_{a>30 \wedge b=10}(R)$

$$T(R)/(3 \cdot V(R,b)) \approx 3$$

f. $\sigma_{b>25 \wedge b=11}(R)$

0 (widersprüchliche Selektion)

g. $\sigma_{b=25 \vee d=13}(R)$

$$T(R)/V(R,b) + T(R)/V(R,d) - T(R)/(V(R,b) \cdot V(R,d)) = 11,8 \approx 12$$

h. $R \bowtie S$

$$T(R) \cdot T(S) / \max[V(R,d), V(S,d)] = 1000$$