

Prof. Dr.-Ing. Dr. h. c. T. Härder
Fachbereich Informatik
AG Datenbanken und Informationssysteme
Universität Kaiserslautern

Übungsblatt 1

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

Aufgabe 1: Der Transaktionsbegriff

Was ist eine Transaktion? Was sind ihre wesentlichen Eigenschaften?

Beurteilen Sie anschließend folgende Aussagen oder Fragen unter ACID-Gesichtspunkten:

1. „Mein Transaktionsprogramm wurde abgebrochen, und nun ist meine Datenbank zerschossen.“
2. „Leider wurde ihre erfolgreich abgeschlossene Transaktion zurückgesetzt, da das Datenbanksystem abgestürzt ist.“
3. „Seit dem Abort meiner Transaktion sind deren Änderungen überhaupt nicht mehr vorhanden!“
4. „Eine andere Transaktion hat Änderungen meiner Transaktion überschrieben. Darf ich jetzt meine Transaktion überhaupt noch beenden, oder muß ich sie abbrechen?“
5. „Nachdem diese mysteriöse Transaktion *M* gelaufen ist, kann keine weitere Transaktion mehr laufen, da sie – trotz erfolgreichem Abschluß – die Datenbank zerstört hat.“
6. „Die Bank mußte leider feststellen, daß Geld ausgezahlt wurde, obwohl die Überweisungstransaktion abgebrochen wurde.“

Aufgabe 2: Function Request Shipping

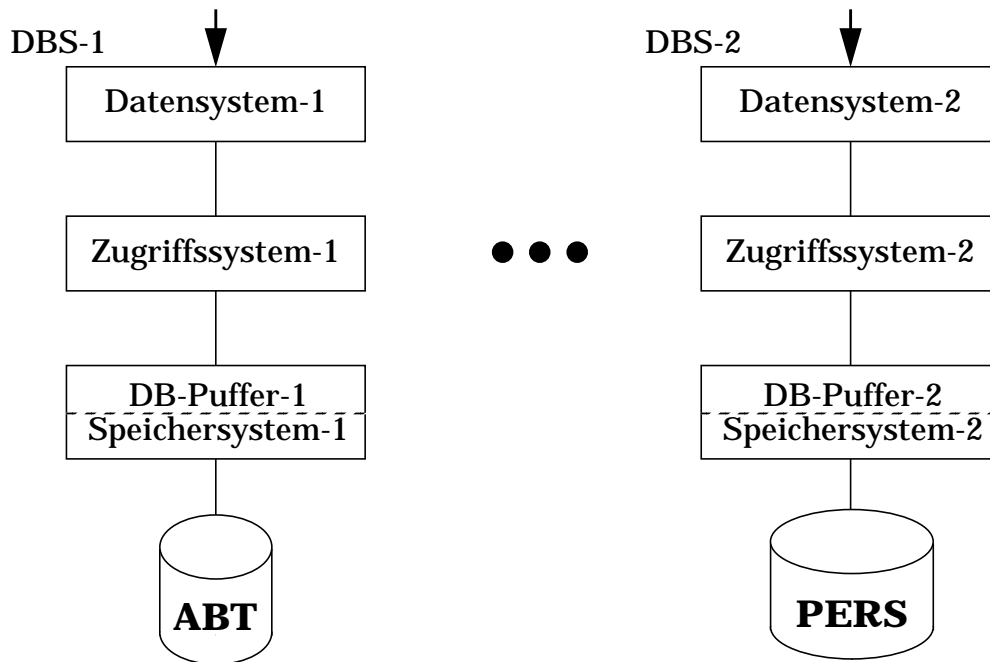
141m

Gehen Sie von der Verteilung einer „PERS – ABT“- Datenbank auf zwei Rechner aus. Welche Kommunikationshäufigkeit und welches Kommunikationsvolumen (Anfragen und Antworten) ergibt sich bei *Function Request Shipping* auf den Ebenen Datensystem, Zugriffssystem und Speichersystem (DB-Puffer), wenn folgende Zuordnung von Anfragen zu Rechnern erfolgt?

Welche Anforderungen werden auf den einzelnen Ebenen verschickt?

```
SELECT COUNT ( * )
FROM PERS
WHERE ANR = 'K55' AND
      BERUF = 'SYSPROG'
```

```
SELECT *
FROM PERS P, ABT A
WHERE P.ANR = A.ANR
```



- Annahmen:** PERS-Datei habe 10^6 Seiten
 ABT-Datei habe 10^4 Seiten
 PERS-Relation habe 10^7 Tupel
 ABT-Relation habe $5 \cdot 10^4$ Tupel
 P.ANR = 'K55' AND P.BERUF = 'SYSPROG': 25 Sätze in PERS

Aufgabe 3: Blockadressierung auf externen Speicher

184

Zur Unterstützung der sequentiellen Verarbeitung werden die Blöcke auf einer Spur nicht sequentiell, sondern versetzt angeordnet, so daß während einer Umdrehung der Platte mehrere Blöcke sequentiell gelesen werden können. Eine Spur auf der Platte bestehe aus $m = 25$ Slots mit je 4 KBytes. Jeder Slot nimmt einen Block auf.

- a) Wie werden Blöcke bei einem Versetzungsmaß I von 1, 2, 3, 4, 6, 7, 8 oder 9 auf einer Spur abgelegt?
- b) Wieviele Umdrehungen sind erforderlich, um die m Blöcke in ihrer logischen Reihenfolge (sequentiell) zu lesen?
- c) Welche Vor-/Nachteile ergeben sich beim zufälligen Lesen einzelner Blöcke?

Aufgabe 4: Zugriffsaufwand bei Dateien

182

Gegeben sei folgende SQL-Anfrage:

```
SELECT *
FROM PERS
WHERE PNR = '123456'
```

Folgende Annahmen sollen gelten:

Die Relation `PERS` sei in Datei `D1` abgelegt. Diese Datei bestehe aus 10^5 Seiten.

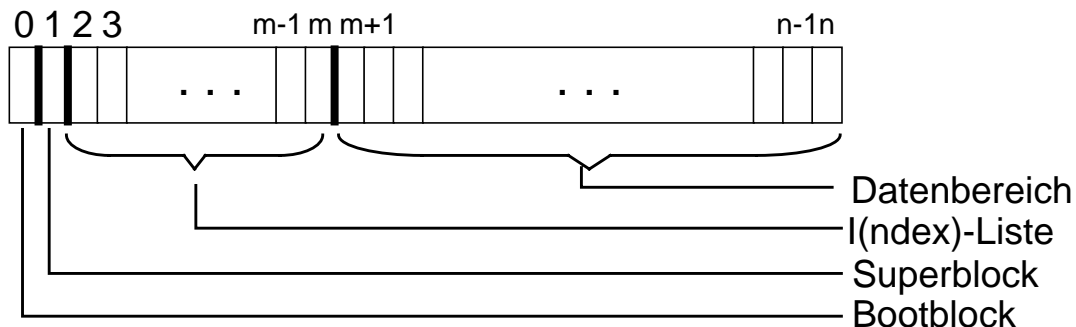
Für das Attribut `PNR` in `PERS` sei ein Index `IPERS(PNR)` als B^* -Baum definiert. Dieser B^* -Baum mit $h^*=3$ sei in der Datei `D2` abgelegt, die aus 10^6 Seiten bestehe. Die Blätter des B^* -Baumes enthalten die Seitennummern P_i in der Datei `D1`, in der sich die zugehörigen `PERS`-Sätze befinden.

Wieviele (externe) Seitenzugriffe sind für `D1` und `D2` zur Beantwortung der SQL-Anfrage notwendig bei:

- dynamischer Extent-Zuordnung
- dynamischer Blockzuordnung (Vektor)
- dynamischer Blockzuordnung (UNIX-Dateisystem):

Kurzinfo: Das Dateisystem in UNIX

- Prinzipieller Aufbau (zur Organisation von Plattenspeicher)

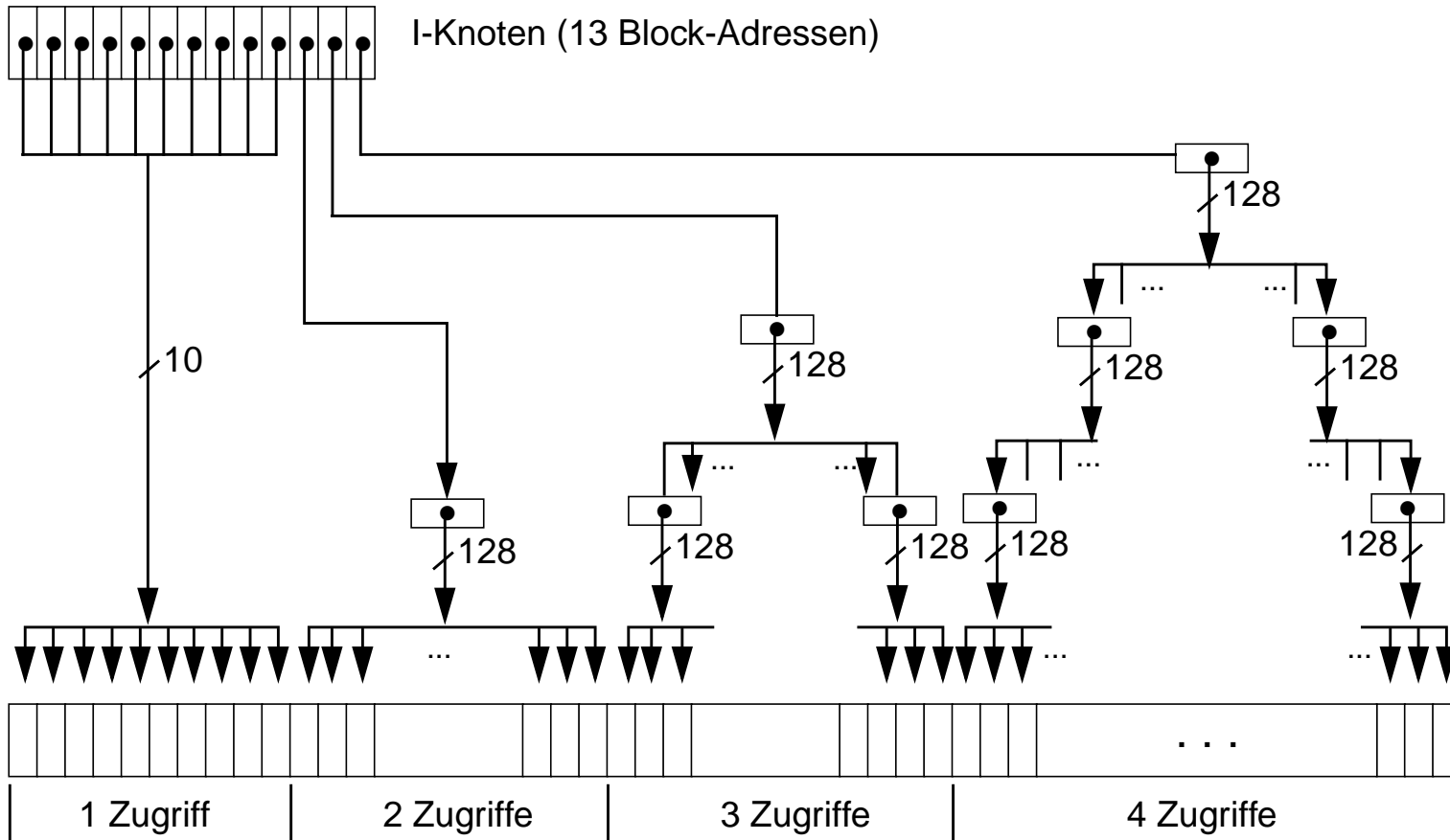


- Wichtige Beschreibungsaspekte

Directory-Einträge ordnen dem Datei-Namen einen Index I ($1 \leq I \leq m-1$) zu. Jeder Knoten der I -Liste enthält folgende Informationen:

- Identifikation des Datei-Eigentümers
- Schutzbits
- Adressen von 13 physischen Blöcken
- Größe der Datei in Bytes
- Zeitpunkt der Erstellung, letzten Referenz, letzten Modifikation
- Anzahl der Verweise auf die Datei
- Art der Datei

Speicherzuordnung im UNIX-Dateisystem



Die Dateiblöcke der untersten Ebene sind nur logisch zusammenhängend