



LUDWIG-
MAXIMILIANS-
UNIVERSITY
MUNICH


DEPARTMENT
INSTITUTE FOR
INFORMATICS


DATABASE
SYSTEMS
GROUP

Skript zur Vorlesung:

Datenbanksysteme II

Sommersemester 2014

Kapitel 6

Anfragebearbeitung

Vorlesung: PD Dr. Peer Kröger

http://www.dbs.ifi.lmu.de/cms/Datenbanksysteme_II

© Peer Kröger 2015

Dieses Skript basiert im Wesentlichen auf den Skripten zur Vorlesung Datenbanksysteme II an der LMU München von

Prof. Dr. Christian Böhm (SoSe 2007),
PD Dr. Peer Kröger (SoSe 2008, 2013, 2014) und
PD Dr. Matthias Schubert (SoSe 2009)



6. Anfragebearbeitung

6.1 Einleitung

6.2 Indexstrukturen

6.3 Grundlagen der Anfrageoptimierung

6.4 Logische Anfrageoptimierung

6.5 Kostenmodellbasierte Anfrageoptimierung

6.6 Implementierung der Joinoperation

6.1 Einleitung

6.2 Indexstrukturen

6.3 Grundlagen der Anfrageoptimierung

6.4 Logische Anfrageoptimierung

6.5 Kostenmodellbasierte Anfrageoptimierung

6.6 Implementierung der Joinoperation

HW-Grundlagen

- Von-Neumann Rechner Architektur: Flaschenhalse

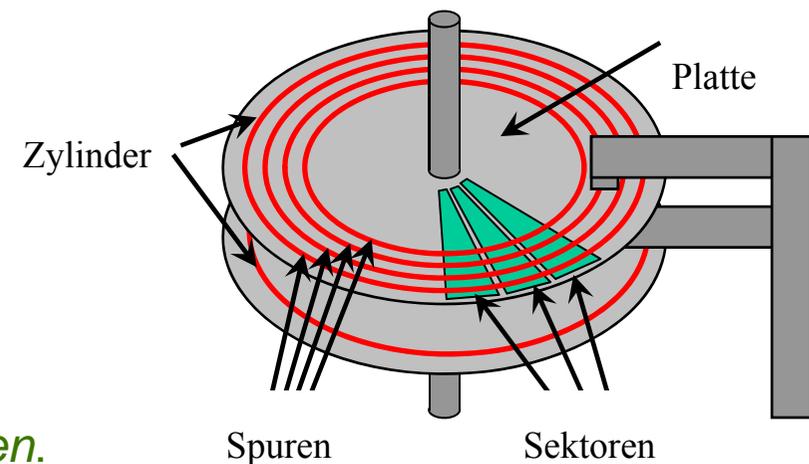


- Zur Vereinfachung unterscheidet man meist nur zwischen
 - **CPU-bound**
CPU, Arbeitsspeicher und Bus bilden den Hauptengpass
 - **I/O-bound**
Hintergrundspeicher und I/O bilden den Hauptengpass

HW-Grundlagen (cont.)

– Schematischer Aufbau einer Festplatte

- Ein Plattenspeichersystem besteht aus *Platten*
- Die Oberfläche der Platten besteht aus *Spuren*
- Die Spuren bestehen aus *Sektoren*.
- *Zylinder* = alle Spuren mit konstantem Radius
- Platten rotieren um gemeinsame Achse, der Arm ist in radialer Richtung bewegbar



HW-Grundlagen (cont.)

- Zugriff auf eine Seite:
 - Setze den Arm auf den gewünschten Zylinder (*Suchen*)
 - Warte bis die Platte so rotiert ist, dass sich der Anfang der Seite unter dem Arm befindet (*Latenz*)
 - Übertrage die Seite in den Hauptspeicher (*Transfer*)
- **Zugriffszeit = Suchzeit + Latenzzeit + Transferzeit**
- I/O-Rate = erwartete Anzahl von Zugriffen pro Sekunde
- Übertragungsrate = maximale Anzahl übertragener Bytes pro Sekunde (Bandbreite)

Speichermedien

- I/O-Engpass auch mit modernen Platten nicht überwindbar
- Lösungsansatz: Verwende statt einer großen Festplatte mehrere kleine, die parallel betrieben werden können

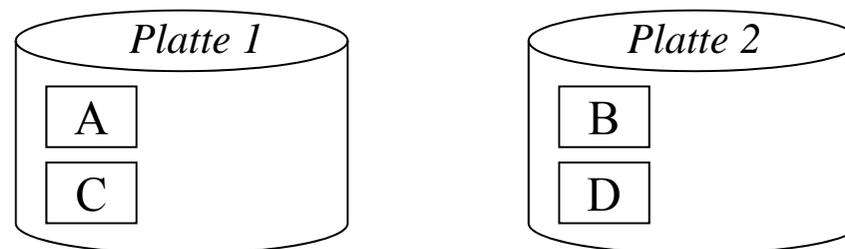
=> RAID-Systeme

- Die Komplexität wird durch den RAID-Controller nach Außen verborgen => es gibt nur ein (virtuelles) Laufwerk
- Acht verschiedene RAID-Level die unterschiedliche Zugriffsprofile optimieren

Speichermedien (cont.)

– RAID 0

- Datenmenge wird durch blockweise Rotation auf die Platten verteilt (**Striping**)
- Beispiel: Striping von 4 Blöcken (A,B,C,D) auf zwei Platten

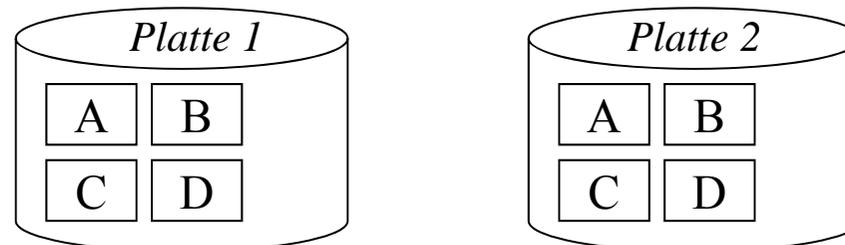


- Größtmögliche Beschleunigung: Anfrage an aufeinanderfolgende Blöcke kann parallel bearbeitet werden
- Fehleranfällig:
 - besteht eine Datei aus vielen Blöcken werden diese über die entsprechenden Platten verteilt
 - Ausfall einer Platte führt zur Beschädigung der Datei

Speichermedien (cont.)

– RAID 1

- Jedes Laufwerk besitzt eine Spiegelkopie (**Mirror**)
- Durch Redundanz ist Fehlerfall eines Laufwerks kein Problem
- Beispiel:

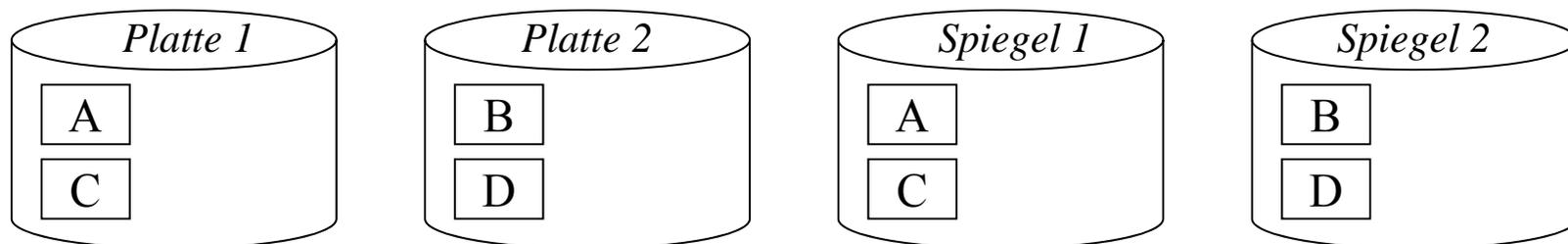


- Leseoperationen parallelisierbar (wie RAID 0)
- Schreiboperationen müssen auf beiden Mirrors (parallel) durchgeführt werden

Speichermedien (cont.)

– RAID 0+1

- Kombination aus RAID 0 und RAID 1
- Verteilung der Datenblöcke wie bei RAID 0
- Spiegelung der Platten wie bei RAID 1



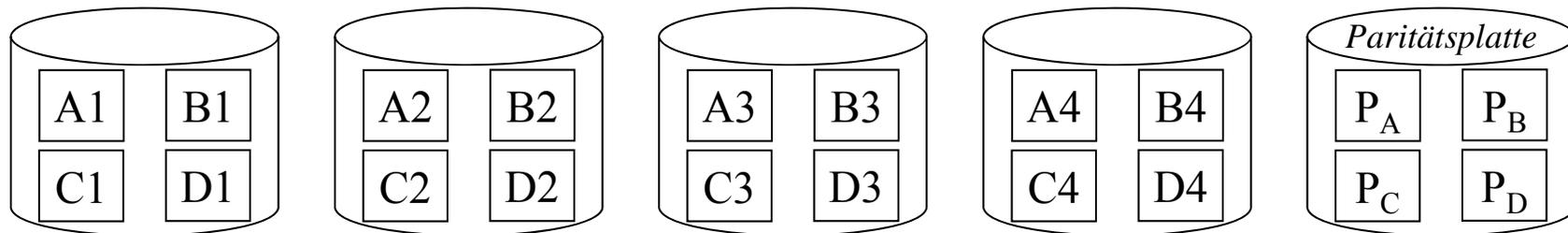
- Vereint Vorteile von RAID 0 und RAID 1
- ABER: Anzahl der benötigten Platten steigt!!!

Speichermedien (cont.)

- Ab RAID 2 wird Datensicherheit ökonomisch günstiger umgesetzt
- Hilfsmittel: Paritätsinformationen
 - Prüfsumme für mehrere Daten
 - Verwendbar, um Daten auf Korrektheit zu überprüfen
 - Verwendbar, um Daten im Fehlerfall zu rekonstruieren
 - Vorgehen:
 - Speichere zu N Datenbereichen auf unterschiedlichen Platten zusätzlich deren Prüfsumme auf einer anderen Platte
 - Ist einer der N Datenbereiche defekt kann dieser aus der Prüfsumme und den $N-1$ übrigen (intakten) Datenbereichen wiederhergestellt werden

Speichermedien (cont.)

- RAID 2
 - Striping auf Bitebene
 - Paritätsinformationen auf separaten Platten
 - In der Praxis meist nicht eingesetzt
- RAID 3 und RAID 4
 - Striping auf Bit- oder Byte-Ebene (RAID 3) bzw. blockweise (RAID 4)
 - Paritätsinformationen auf einer speziellen Platte

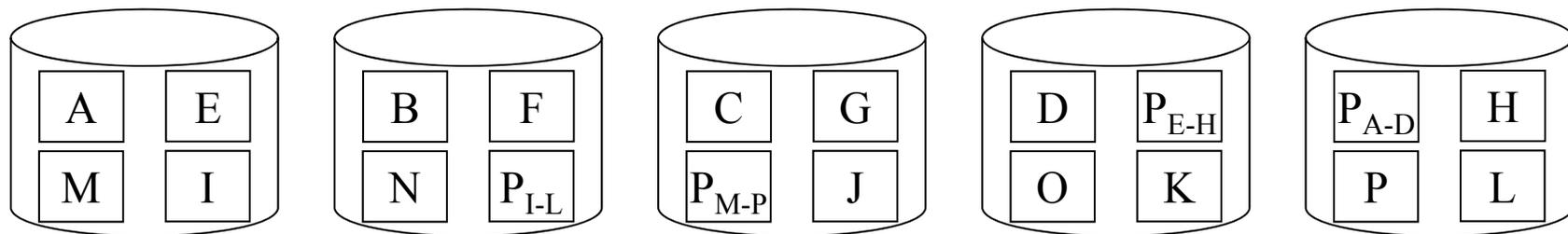


- Nachteil: jede Schreiboperation muss auf Paritätsplatte zugreifen

Speichermedien (cont.)

– RAID 5

- Striping blockweise (wie RAID 4)
- Verteilung der Paritätsinformationen auf alle Platten



- Damit ist der Flaschenhals der Paritätsplatte beseitigt
- Schreiboperationen setzt aber nach wie vor die Neuberechnung und das Ablegen des neuen Paritätsblocks voraus
- ACHTUNG: Paritätsblock P_x kann nur ein Fehler in den Daten X korrigieren!

Speichermedien (cont.)

- Abschließende Bemerkungen
 - Wahl des RAID-Levels hängt vom Anwendungsprofil ab (z.B. Anteil der Leseoperationen im Vergleich zu Schreiboperationen)
 - Kommerzielle RAID-Systeme erlauben typischerweise eine flexible Konfiguration
 - Viele DBS unterstützen Striping von Tupeln auf unterschiedliche Platten auch ohne Einsatz von RAID-Systemen
 - Trotz der Fehlertoleranz von RAID-Systemen ist der Einsatz von Recovery-Techniken (Kapitel 4) wichtig

Anfragebearbeitung

- Zu bearbeitende Seiten müssen vom HGSP in den DB-Puffer geladen werden
- Problem: Verwaltung der Daten auf einem Speichermedium sequentiell
 - Zeitaufwand für Bearbeitung einer Suchanfrage: $O(n)$
(im ungünstigsten Fall alle n Datensätze durchsuchen)
- Wird ein bestimmter Datensatz anhand eines Suchkriteriums gesucht, kann über eine Indexstruktur eine aufwändige Suche vermieden werden
- Der Index erlaubt es, die Position des Datensatzes innerhalb des Mediums schnell zu bestimmen.