

GDB Tutorium

Woche 09

Jigao Luo

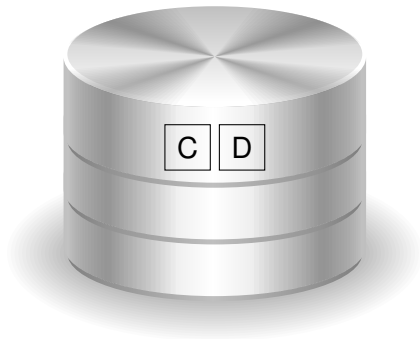
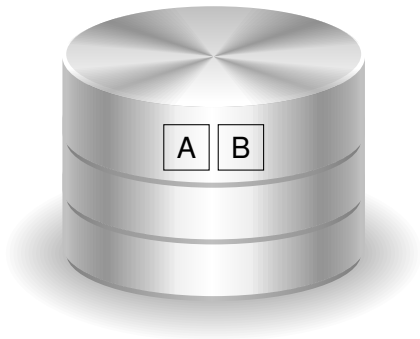
TUM

19. Dezember 2019

- RAID
- B+ Tree
- Erweiterbares Hashing

- RAID: redundant array of independent/inexpensive disks
- mehrerer Disks zu einem logischen Laufwerk
- Ausfallsicherheit
- geringere Zugriffszeit

- Nur Striping der Blöcke
- Keine redundanzfreie
- Doppelte Bandbreite
- Aber: Datenverlust

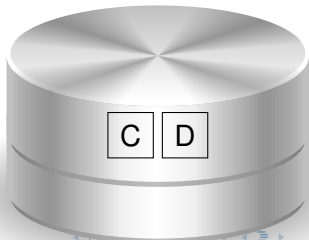
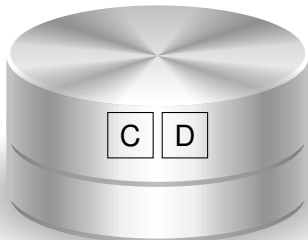
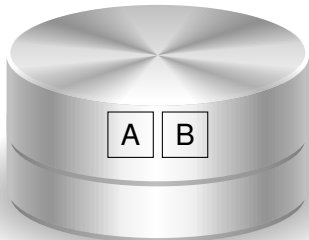
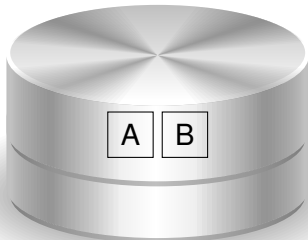


- Nur Mirroring der Blöcke: Redundanz
- Doppelter Speicherbedarf
- Ausfallsicherheit
- Aber: Write/Update/Delete sind aufwändiger



Wiederholung: RAID0+1 Striping+Mirroring

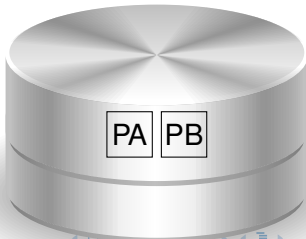
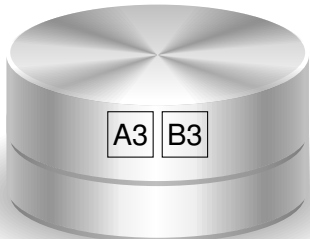
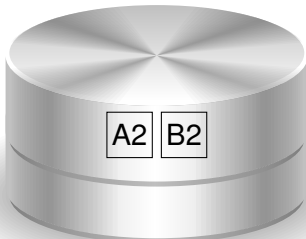
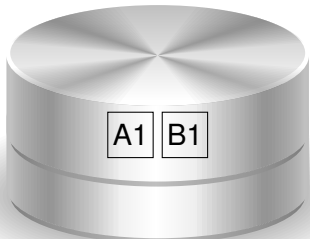
- Doppelte Bandbreite
- Ausfallsicherheit
- Doppelter Speicherbedarf



- zusätzlich auf einer Platte noch Fehlererkennung und Fehlerkorrektur
- In der Praxis nicht eingesetzt

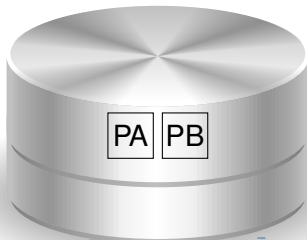
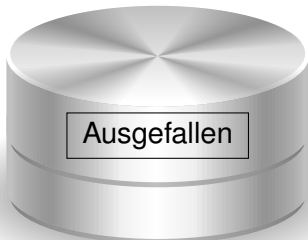
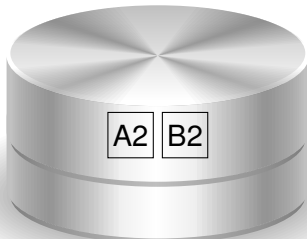
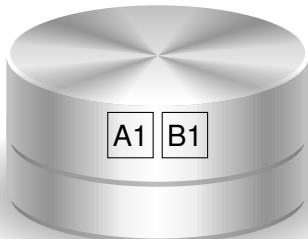
Wiederholung: RAID3 Bit level Striping

- Parität-Information auf eigener zusätzlich Festplatte
- Fehlerkorrektur
- $PA = A1 \text{ XOR } A2 \text{ XOR } A3$, $PB = B1 \text{ XOR } B2 \text{ XOR } B3$



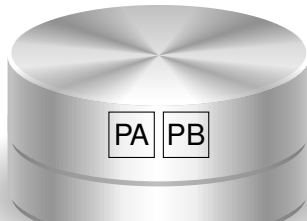
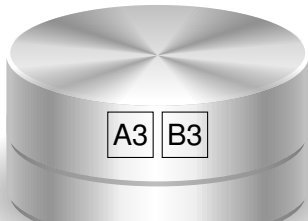
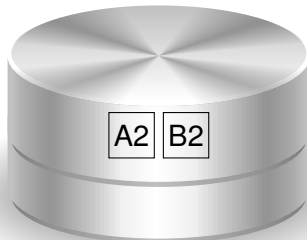
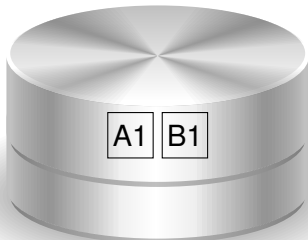
Wiederholung: RAID3 Bit level Striping

- Parität-Information auf eigener zusätzlich Festplatte
- Fehlerkorrektur
- $PA = A1 \text{ XOR } A2 \text{ XOR } A3 \Rightarrow A3 = A1 \text{ XOR } A2 \text{ XOR } PA$

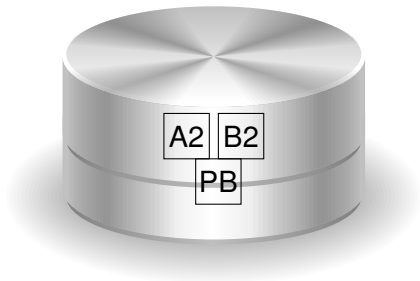
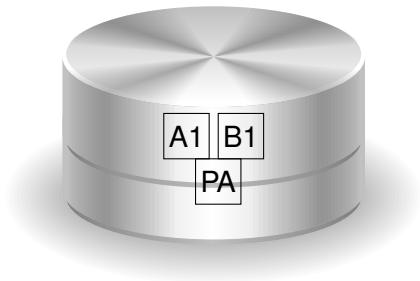


Wiederholung: RAID4 Block level Striping

- Parität-Information auf eigener zusätzlich Festplatte
- Fehlerkorrektur
- $PA = A1 \text{ XOR } A2 \text{ XOR } A3 \Rightarrow A3 = A1 \text{ XOR } A3 \text{ XOR } PA$
- Flaschenhals bei der Parität-Platte



- Parität-Information verteilt
- Fehlerkorrektur
- Kein Flaschenhals bei der Parität-Platte
- Wird in der Praxis eingesetzt



- Typ (k, k^*)
- Jeder Weg von Wurzel zum Blatt hat die gleiche Länge.
- Jeder innere Knoten hat mindestens k und maximal $2k$ Einträge.
- Jedes Blatt hat mindestens k^* und maximal $2k^*$ Einträge.
- Blätter sind in Double Linked List.
- Insert wie B Tree Insert + innere Knoten Behandlung beim Überlauf an inneren Knoten.
- B Tree im nächsten Übungsblatt.

- Dynamisches Wachsen
- Zusätzlich Indrektion über ein Directory
- Der Zeiger des Hash-Bucket enthält den Zugriff auf ein Directory
- Der Zugriff auf das Directory erfolgt über einen binären Hashcode
- Das Directory hat eine globale Tiefe t : erste t -Stelle für Directory relevant
- Jeder Bucket hat eine lokale Tiefe t' : erste t' -Stelle für den Bucket relevant
- Eine Aufteilung von Directory ist nötig beim Einfügen mit $t' = t$

Die folgende Abbildung zeigt einen Festplattenverbund bestehend aus vier Laufwerken, auf welchen die Datenblöcke A bis I gespeichert sind. Die Blöcke P i enthalten Paritätsin- formationen.

- a) Um welches RAID-Level handelt es sich?
- b) Wieviele Festplatten können ausfallen, ohne dass mit Datenverlust zu rechnen ist? Geben Sie eine allgemeine Lösung für einen Verbund bestehend aus n Festplatten an.
- c) Kann die Ausfallsicherheit erhöht werden? Begründung?
- d) Welchen weiteren Vorteil bietet das gezeigte RAID-System neben der Ausfallsicher- heit?
- e) Nach einem Festplattendefekt enthalten die Datenblöcke die folgenden Binärdaten. Rekonstruieren Sie die Datenblöcke der Disk 2 mithilfe der XOR-Verknüpfung.

Gegeben sei ein Array von 1.000.000.000 8-Byte-Integer-Werten und ein Programm, das alle Werte aufsummiert.

Das Programm wird auf einem System mit 16 GB Hauptspeicher und einer herkömmlichen Magnetfestplatte (Größe 1 TB), auf der alle Werte sequentiell gespeichert sind, ausgeführt. Ein Random Access auf die Festplatte dauert 10 ms, beim sequentiellen Lesen hat sie einen Durchsatz von 160 MB/s. Das Summieren zweier Werte im Hauptspeicher dauert 1 ns. ($1 \text{ MB} = 10^6 \text{ B}$ und $1 \text{ TB} = 10^{12} \text{ B}$)

- a) Gehen Sie davon aus, dass alle Werte bereits im Hauptspeicher liegen. Wie lange läuft das Programm?
- b) Nun liegen alle Werte ausschließlich auf der Festplatte. Wie lange läuft das Programm jetzt?
- c) Auf der Festplatte liegt jetzt zusätzlich nach jedem 100.000. Wert die Summe der 100.000 davorliegenden Werte. Wie lange läuft das Programm, wenn es nur diese Summen aufsummiert?

- Sequential I/O: read time
- Random I/O: seek time

- a) Fügen Sie in einen anfänglich leeren B + -Baum mit $k = 3$ und $k^* = 2$ die Zahlen eins bis fünfundzwanzig in aufsteigender Reihenfolge ein. In den Blattknoten werden TIDs verwendet. Was sind TIDs, wann lohnt sich ihre Verwendung, was ist die Alternative zu TIDs?
- b) Erläutern Sie die Vorgehensweise bei der Bearbeitung der folgenden Anfrage “Finde alle Datensätze mit einem Schlüsselwert zwischen 5 und 15.”

- Fügen Sie nacheinander die folgenden Einträge in eine anfangs leere erweiterbare Has-htabelle, welche 2 Einträge pro Bucket aufnehmen kann, ein. Es soll effizient nach der KundenNr gesucht werden können.

- Gegeben sei eine erweiterbare Hashtabelle mit globaler Tiefe t . Wie viele Verweise zeigen vom Verzeichnis auf einen Behälter mit lokaler Tiefe t' ?



T. Neumann, “Grundlagen: Datenbanken, Kapitel 05”,,



A. Kemper, “Grundlagen: Datenbanken, Kapitel 06”,,



F. Bauer, “<https://home.in.tum.de/~bauerf/gdb/index.html>”,,