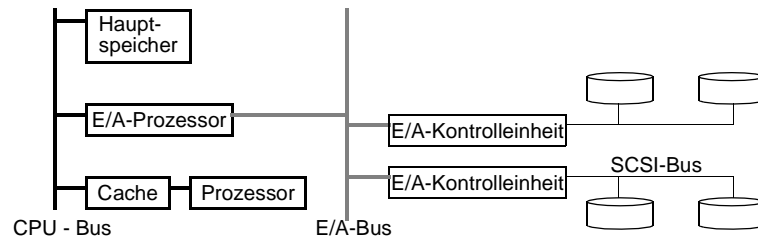
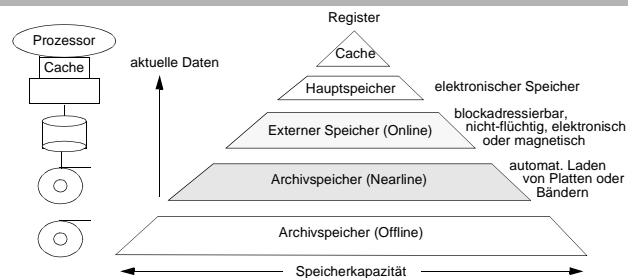


A. Speichertechnologie

- ☐ Speicherhierarchien
- ☐ Externspeichersysteme (Plattenspeicher)
- ☐ unterstützende Komponenten



Entwurf einer rekursiven Speicherhierarchie



- ❑ Ähnliche Verwaltungsaufgaben auf jeder Ebene der Speicherhierarchie:
 - Lokalisieren von Datenobjekten
 - Allokation von Speicherplatz
 - Ersetzung
 - Schreibstrategie (write-through vs. write-back)
 - ggf. Anpassung an verschiedene Transfergranulate

A.1 Speicherhierarchie

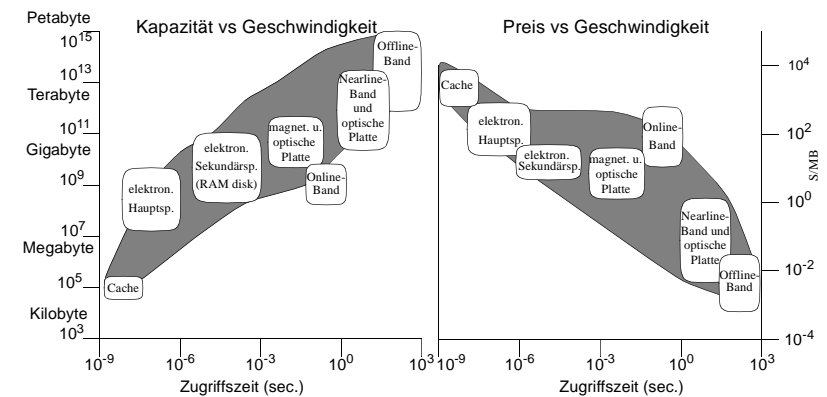
Idealer Speicher besitzt folgende Eigenschaften:

- ☐ nahezu unbegrenzte Speicherkapazität
- ☐ kurze Zugriffszeit (im Durchschnitt und im schlechtesten Fall) bei wahlfreiem Zugriff
- ☐ niedrige Zugriffskosten (Kosten/Zugriff/Sekunde möglichst niedrig)
- ☐ geringe Speicherkosten (Kosten pro MB möglichst niedrig)
- ☐ Nichtflüchtigkeit
- ☐ erlaubt logische und arithmetische Verknüpfungen

Eine Speicherhierarchie mit realen Speichermedien stellt ein Versuch der Approximation eines idealen Speichers dar, indem Lokalitätseigenschaften auf den Ebenen ausgenutzt werden.

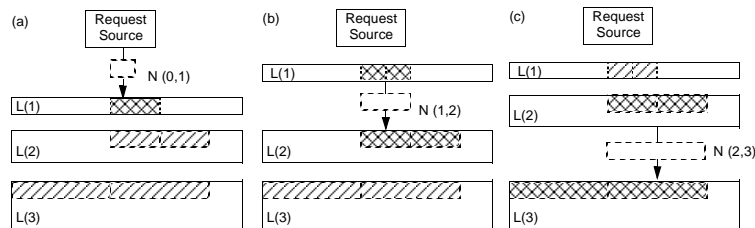
- ☐ Allokation von Daten mit hoher Zugriffswahrscheinlichkeit in schnelle (relativ kleine) Speicher.
- ☐ Mehrheit der Daten verbleibt auf langsameren, kostengünstigeren Speicher.

Preis, Kapazität und Zugriffszeit



Schreib- und Lesestrategien

- ❑ **Read-through:**
Auf Anforderung werden die Daten im entsprechenden Granulat stufenweise in der Hierarchie "nach oben" bewegt.
- ❑ **Write-through (FORCE):**
Sofort nach *Änderung* werden die Daten auch auf der nächsttieferen Stufe geändert.
- ❑ **Write-back:**
Bei *Ersetzung* werden die geänderten Daten stufenweise "nach unten" geschoben.



366.

Warum funktioniert das Konzept?

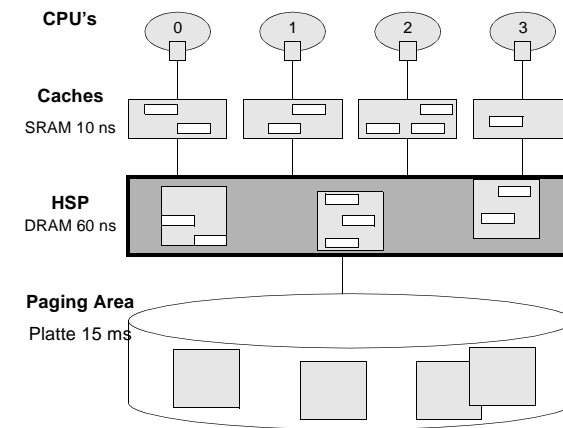
- ❑ Auf die gleichen Daten wird oft innerhalb eines kurzen Zeitraums mehrfach zugegriffen.
- ❑ Auf im Speicher räumlich nah beieinander liegende Daten wird innerhalb eines kurzen Zeitraums zugegriffen.
- ❑ Beurteilung des Nutzens eines Caches durch die Trefferrate:

$$T = \text{Treffer im Cache} / \text{alle Referenzen auf Cache}$$
 und durch die effektive Zugriffszeit:

$$T \cdot C + (1-T) \cdot Z$$
 wobei C die Zugriffszeit des Caches und Z die Zugriffszeit auf die nächst tiefere Ebene ist.
- ❑ Beispiel: $C = 0.01 \cdot Z$ ergibt eine effektive Zugriffszeit von etwa $(1-T) \cdot Z$
 \Rightarrow Hohe Trefferraten sind Voraussetzung für eine hohe Leistung
- ❑ Verbesserung der Trefferrate durch
 - Clusterung von gemeinsam referenzierte Daten (und Instruktionen)
 - größere Caches
- ❑ Kosten/Nutzen-Verhältnis muß berücksichtigt werden ("5-Minuten Regel")

368.

Speicherhierarchie in heutigen Rechnern



367.

Charakteristika von Speichertypen

Typ	Einsatz	Z.-Zeit	Ü.-Rate	Ü.-Breite	typ.Kapaz. pro Einheit	flüchtig	Überschr.
SRAM-CMOS	Prozessor-Cache	25 ns	—	4 B	4 Mbit	J	J
DRAM-CMOS	Haupt-speicher	100 ns	—	32 B	4 Mbit	J	J
FE-RAM		100 ns	—		256 Kbit		
DRAM	Erweiterungs-speicher	50 µs	100 MB/s	4 KB	4 Mbit	J	J
Solid State Disk (DRAM+Batterie)	Extern-speicher	~ 1 ms	4.5 MB/s	4 KB	600 MB	N	J
Magnetplatte	Extern-speicher	< 20 ms	3.0 MB/s	4 KB	1.8 GB	N	J
Magnetband	Archiv-speicher	> 1 s	1.5 MB/s	256 KB	2.3 GB	N	J
Optische Platte (WORM)	Archiv-speicher	< 0.2 s	<12 MB/s	4 KB	≤ 10 GB	N	J

369.

Entwicklungstrends

- ❑ Prozessoren (*Joy's Law*):

$$\text{SunMips}(\text{year}) = 2^{\text{year}-1984} \text{ Mips for year in [1984 ... 2000]}$$

- ❑ Elektronische Speicher (*Moore's Law*):

$$\text{MemoryChipCapacity}(\text{year}) = 4^{\frac{\text{year} - 1970}{3}} \text{ Kbit/chip for year in [1970 ... 2000]}$$

- ❑ Magnetische Speicher (*Hoaglands's Law*):

$$\text{MagneticAreaDensity}(\text{year}) = 10^{\frac{\text{year} - 1970}{10}} \text{ Mb/inch}^2 \text{ for year in [1970 ... 2000]}$$

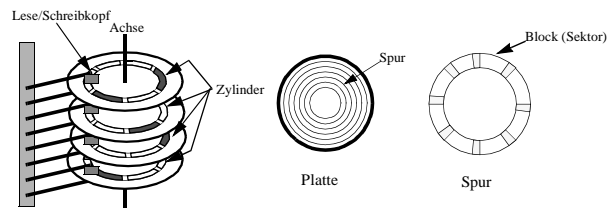
- ❑ derzeitige Trends

- kleinere Platten (2.25 und 3.5 Zoll)
- viele kleine Platten werden logisch als eine Platte verwendet (Disk-Array)

370.

A.3 Magnetische Plattenspeicher

- ❑ entspricht einem Plattenstapel (der Einfachheit halber: Disk oder Platte genannt)



- ❑ Sektoren (typischerweise 1, 2 oder 4 KB) sind die kleinste Schreib-/Leseinheit auf einer Platte
- ❑ Block entspricht einer zusammenhängenden Menge von Sektoren
- ❑ Adressierung
 - Zylindernummer, Spurnummer, Sektornummer
 - jeder Sektor speichert selbstkorrigierende Fehlercodes;
- ❑ bei defekten Sektoren erfolgt eine automatische Abbildung auf Ersatzsektoren

372.

A.2 Elektronischer Speicher

- ❑ enorme Kapazitätssteigerungen
 - 1970: 1Kbit/chip
 - 1992: 16 Mbit/chip
 - alle drei Jahre erhöht sich die Kapazität pro Chip um einen Faktor 4 (Gleichung von Moore).
- ❑ moderate Steigerungen bei der Geschwindigkeit
 - Zugriffszeit liegt derzeit bei 50-80 ns
 - Einführung von Cache-Speicher
- ❑ relativ teuer: 100 DM pro MB

371.

Eigenschaften heutiger Plattenspeicher

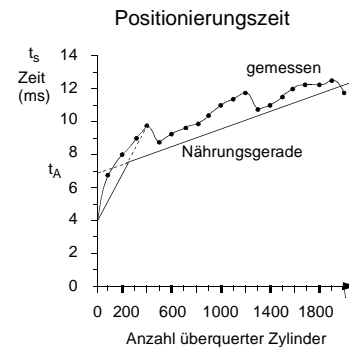
- ❑ Durchmesser: 4.6 cm - 20 cm (wird üblicherweise in Zoll bzw. Inch angegeben)
- ❑ Speicherdichte (außen höher als innen)
 - lineare Dichte: 50000 Bits/inch
 - Spuredichte: 2500 Spuren/inch
- ❑ Zylinder (Platte) entspricht einer linearen Folge von Blöcken
 - Spuren werden versetzt abgelegt (track skewing)
 - Beispiel einer kommerziell erhältlichen Platte:

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

373.

Lesen/Schreiben von Blöcken

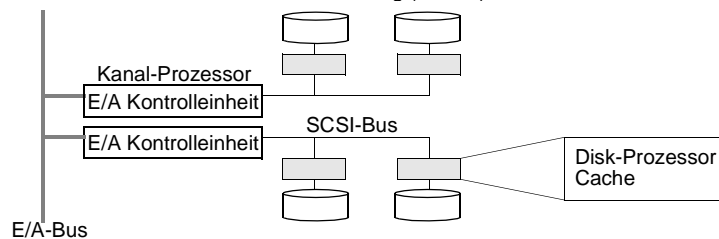
- ❑ Zugriffsbewegung (Positionierung) läuft in vier Phasen ab:
 - Beschleunigung
 - konstante Geschwindigkeit (nur bei langen Zugriffen)
 - Abbremsen
 - Feinjustierung
- ❑ Rotationsverzögerung
- ❑ Transferzeit
- ❑ Kopfschaltzeit
- ❑ Approximation der Positionierungszeit
- ❑ durch Funktion mit zwei Komponenten:
 - Wurzelfunktion
 - lineare Funktion



374.

Low-level Kontrolleinheiten

- ❑ Aufgaben
 - Umsetzen der E/A-Anforderungen (legt die Bedienungsreihenfolge fest)
 - physische Kontrolle des Plattenspeichers
 - Abbildung der linearen Folge von Blöcken auf die Platte (ist i.a. nicht 1:1, da defekte Blöcke ersetzt werden)
 - Verwaltung des Platten-Caches
 - Verwaltung des Verbindungsbus (SCSI = Small Computer Systems Interface)
- ❑ Overhead für die Kontrolleinheit ist niedrig (< 1 ms).



376.

Kennzahlen eines Plattenspeichers

Merkmal	typische Werte 1995
t_{smin} = Zugr.bewegung(Min)	1 ms
t_{sav} = " (Mittel)	10 ms
t_{smax} = " (Max.)	20 ms
t_{rev} = Umdrehungszeit	10 ms
Kopfschaltzeit	2 ms
T_{cap} = Spurkapazität	100 KB
T_{cyl} = #Spuren pro Zylinder	10
N_{dev} = #Zylinder	3000
Transferrate (brutto)	10 MB/s
Nettokapazität (2- o. 4KB-Blöcke)	3 GB

375.

Warum wird ein Cache an der Platte benutzt?

- ❑ Mehrere Platten teilen sich ein SCSI-Bus.
 - Problem ohne Cache:
 - Platte kann nicht Daten übertragen, da der Bus belegt ist.
 - Platte kann erst nach einer vollständigen Umdrehung übertragen.
 - Zugriffszeit erhöht sich!
 - Lösung: Block wird zunächst in Cache übertragen.
- ❑ Es werden oft benachbarte Blöcke hintereinander, aber einzeln, angefordert.
 - Problem ohne Cache:
 - Annahme: Blöcke B1 und B2 liegen hintereinander auf einer Spur und werden durch zwei Anforderungen eingelesen.
 - Zugriffszeit für Block B2 entspricht einer vollständigen Umdrehung.
 - Lösung:
 - **Prefetching:** Lese mehrere Blöcke in den Cache, obwohl nur einer explizit angefordert wurde.
 - Prefetching lohnt sich, wenn die Wahrscheinlichkeit hoch ist, daß einer der nächsten Anforderungen einen dieser Blöcke benötigt.

377.

A.4 Parallele Plattenspeichersysteme: Disk Arrays

Idee:

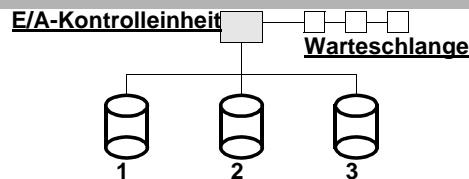
- ☐ Verwende mehrere Disks (Disk Array) statt einer großen Disk
- ☐ Transparente Verteilung von Daten über mehrere Platten

Eigenschaften

- ☐ parallele Verarbeitung von **unabhängigen** Lese- und Schreiboperationen (Verbesserung der E/A-Rate)
 - ⇒ Verringerung der Wartezeiten vor der Platte
- ☐ **eine** Lese- und Schreiboperation kann durch mehrere Platten bedient werden (Verbesserung der Transferrate)
- ☐ je mehr Platten im Diskarray desto größer der Gewinn
 - ⇒ Ausfallsicherheit nimmt linear mit der Anzahl der Platten ab.
 - ⇒ Speicherung redundanter Daten (damit Ausfallsicherheit verbessert wird).

378.

Vollständig Synchronisierte Platten



- ☐ Umdrehungsgeschwindigkeit ist für alle Platten gleich
- ☐ Plattenarm aller Platten liegt auf dem gleichen Zylinder
- ☐ Lesen und Schreiben geschieht stets von allen Platten

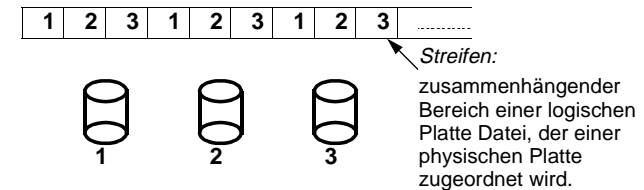
Vollständig synchronisierte Platten zeigen ein ähnliches Leistungsverhalten wie eine große Platte, jedoch wird

- ☐ die Übertragungsrate um den Faktor d gesteigert.
- ☐ die Clustering der Daten verbessert, da die Kapazität "eines" Zylinders um den Faktor d ansteigt.

380.

Prinzipielle Idee

- ☐ Eine logische Platte wird verstreut über mehrere Platten gespeichert:



d = Anzahl der physischen Platten.

Fragen:

- ☐ Was ist die geeignete Größe einer Streifeneinheit?
- ☐ Wie werden redundante Daten berechnet?
- ☐ Wie werden redundante Daten über die Platten verteilt?

379.

Entscheidender Nachteil vollständig synchronisierter Platten:

- ☐ E/A-Rate verbessert sich im wesentlichen nicht

Eignet sich nicht für die meisten Datenbankanwendungen, da

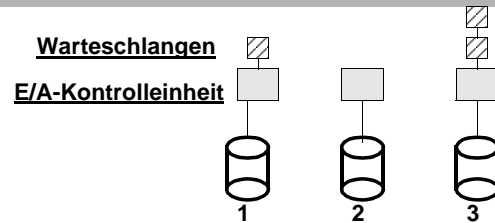
- ☐ zu viele Daten unter einem Plattenarm liegen
- ☐ wenn die Streifen zu groß sind, werden zu viele Daten übertragen
- ☐ wenn die Streifen zu klein sind, werden zu viele unabhängige Zugriffe auf die Platten benötigt
- ☐ Wartezeiten vor der Disk können sehr lang sein

Vollständig synchronisierte Platten sind effizient für Anwendungen mit folgenden Eigenschaften:

- ☐ ein Benutzer verwendet das System
- ☐ große Datenmengen müssen übertragen werden (Multimedia-Daten)
- ☐ z. B. Einlesen der Daten für die Lösung großer numerischer Probleme

381.

Vollständig unabhängige Platten



- ☐ Positionen der Arme sind unabhängig voneinander
- ☐ Umdrehungsgeschwindigkeit ist nicht gleich für alle Platten
- ☐ Verwendung eines oder mehrerer E/A-Kontroller ist optional

Vorteil:

- ☐ hohe E/A-Rate für kleine E/A-Operationen (kann bis zu d-mal höher sein)

Nachteil:

- ☐ hohe erwartete Zugriffszeit für große E/A-Operationen

382.

Wunscheigenschaften bei der Einführung von Redundanz

- ☐ niedriger Speicherplatzverbrauch
- ☐ Durchsatz und Antwortzeit sollen sich nicht verschlechtern
 - a) im Normalbetrieb
 - b) beim Ausfall einer Platte
- ☐ beim Ausfall einer Platte soll die Reparaturzeit kurz sein.
- ☐ erwartete Ausfallszeit des Gesamtsystems soll hoch sein.

Klassifizierung von Lösungsmethoden

D. Patterson, G. Gibson, R. Katz: "A Case for Redundant Arrays of Inexpensive Disks (RAID)", Proc. ACM SIGMOD 1988, 109-116.

- ☐ RAID 0: keine Redundanz
- ☐ RAID 1: Spiegelplatten
- ☐ RAID 4: Einführung einer Platte mit redundanter Information
- ☐ RAID 5: gleichmäßige Verteilung der redundanten Information über alle Platten

384.

A.4.1 Fehlertoleranz in Diskarrays

schlechte Eigenschaften von parallelen Platten:

- ☐ Annahmen
 - Ausfallwahrscheinlichkeit einer Platte ist unabhängig von der anderer Platten
 - konstante Ausfallsrate \Rightarrow Ausfallszeit folgt einer exponentiellen Verteilung
- ☐ Erwartete Aufallszeit $MTTF_d$ eines Diskarrays aus d Platten:

$$MTTF_d = MTTF_1/d$$

$MTTF_1$ (mean time to failure) ist die erwartete Ausfallszeit einer Platte

- ☐ Beispiel:
 $MTTF_1 = 60$ Monate, $d = 20$ ergibt also: $MTTF_{20} = 3$ Monate

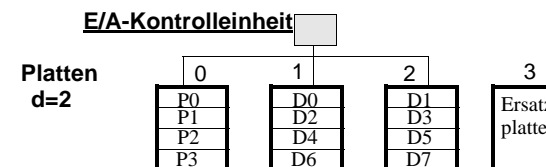
\Rightarrow Ausfallszeit eines parallelen Plattensystems ist zu niedrig

Anforderung:

- ☐ Alle Daten sollen im Fall eines Ausfalls (mindestens) einer Platte noch zugreifbar sein.

383.

RAID 4



- ☐ Bestimmung der Position des Blocks D_i
 - Platte (*Spalte*): $1 + i \bmod d$
 - lokale Position (*Zeile*): $i \div d$
- ☐ Die erste Platte enthält sogenannte Parityblöcke. Diese berechnen sich folgendermaßen:

$$P_i = D_{i*d} \text{ XOR } D_{i*d+1} \text{ XOR } \dots \text{ XOR } D_{(i+1)*d-1}$$

wobei XOR für jedes Bit der Blöcke ("Array of Bit") angewendet wird.

385.

Ausnutzung folgender Eigenschaft von XOR

Seien a , b und $c \in \{0,1\}$. Falls $a = b \text{ XOR } c$, dann gilt $c = a \text{ XOR } b$ und $b = a \text{ XOR } c$.

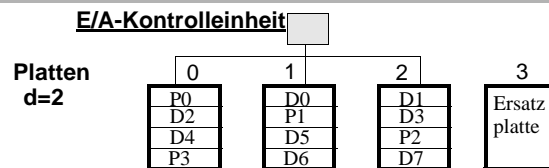
Diese Eigenschaft folgt sofort auch für Blöcke ("Array of Bit"):

Operationen:

- ☐ Normalbetrieb (alle Platten sind funktionsfähig)
 - Lesen eines Blocks D_i
 - Schreiben eines Blocks D_i (hier mit D_i^{neu} bezeichnet)
 - a) Lese den Block D_i (hier mit D_i^{alt} bezeichnet) und den Parityblock $P_{i \div d}$.
 - b) $P_{i \div d}^{\text{neu}} = (D_i^{\text{alt}} \text{ xor } P_{i \div d}) \text{ xor } D_i^{\text{neu}}$
 - c) Schreibe die Blöcke $P_{i \div d}^{\text{neu}}$ und D_i^{neu} auf die Platte.
- Für das Schreiben eines Blocks müssen also zwei Lese- und zwei Schreiboperationen ausgeführt werden. Dabei werden zwei der Operationen immer auf der Parityplatte ausgeführt.
- ⇒ Verschlechterung der Leistung (im Vergleich zu ohne Redundanz)
 - ⇒ Parityplatte wird zum Flaschenhals des Diskarrays.

386.

RAID 5



Idee:

- ☐ Zur Vermeidung des Flaschnhals einer einzelnen Parityplatte werden die Parityblöcke über alle Platten gleichmäßig verteilt.

Abbildung der Datenblöcke

- ☐ Datenblock D_i
 - lokale Position (Zeile): $i \div d$
 - Platte: $i \bmod (d+1)$, falls $i \bmod d < i \bmod (d+1)$
 $1 + i \bmod (d+1)$ sonst

388.

- ☐ Fehlerfall 1 (Parity-Platte ist ausgefallen)
 - Lesen und Schreiben wird wie bei einem Diskarray ohne Redundanz durchgeführt.
- ☐ Fehlerfall 2 (Datenplatte p ist ausgefallen)
 - Lesen eines Blocks $D_{i \div d + p}$ (von der Platte p)
 - Lesen der Blöcke $P_i, D_{i \div d + 1}, \dots, D_{i \div d + p-1}, D_{i \div d + p+1}, \dots, D_{i \div (d+1) - 1}$ und Rekonstruktion des Blocks $D_{i \div d + p}$
 - Schreiben eines Blocks $D_{i \div d + p}$ (auf die Platte p)
 Es wird nun analog zum Normalfall vorgegangen, wobei der "alte" Block zunächst rekonstruiert werden muß. Das Schreiben des Blocks entfällt.

Rekonstruktion einer Platte im Fehlerfall

- ☐ Bei einem Plattenausfall sollte möglichst schnell die Ersatzplatte in das Diskarray integriert werden. Dies kann beim on-line Betrieb des DiskArrays ablaufen.
 - ⇒ Leistungseinbußen während der Rekonstruktion

Speichermehraufwand

- ☐ Für ein Diskarray mit d Platten wird eine Parity- und eine Ersatzplatte verwendet.

387.

Abbildung der Parityblöcke

- ☐ Parityblock P_i
 - Platte: $i \bmod (d+1)$
 - lokale Position: i

Operationen

- ☐ Lesen und Schreiben eines Blocks entspricht RAID 4

Leistung im Normalfall

- ☐ Durch Verteilung der Parityblöcke ergibt sich im Vergleich zu RAID 4 eine verbesserte Leistungsfähigkeit.
- ☐ Im Vergleich zu einem RAID 0 (keine Redundanz) bleibt der Nachteil bestehen, daß ein Schreiben eines Blocks insgesamt 4 E/A Operationen benötigt.
 - ⇒ niedrigere E/A Rate (etwa um einen Faktor 4)

389.

Leistung bei Fehler

Erwartete Zeit für den Ausfall des Gesamtsystems:

$$MTTF_{RAID} = \frac{MTTF}{d+1} \times \frac{MTTF}{d \times MTTR}$$

- ☐ MTTR = erwartete Zeit bis das Diskarray repariert ist
- ☐ MTTF = erwartete Zeit bis eine Platte des Diskarray ausfällt

Beispiel:

MTTR = 3 Stunden, MTTF = 30.000 Stunden, d = 100: $MTTF_{RAID} = 30.000$ Stunden

Folgerung

- ☐ Bei einer hohen Anzahl von Platten sollten die Platten in Gruppen aufgeteilt werden und pro Gruppe jeweils eine Platte zur Speicherung der Parityblöcke geopfert werden.

390.

Eigenschaften

Leseoperation:

- ☐ Lese die Datenseite von einer der beiden verfügbaren Platten (kürzere Suchzeiten: nehme die Platte mit der geringsten Positionierungszeit)
- ☐ Es kann gezeigt werden, daß bei einer Leseoperation die durchschnittliche Suchzeit bei der Verwendung von k Spiegelpatten

$$1/(2k+1)$$

beträgt (relativ zu der Anzahl der Zylinder einer Platte)

Schreiboperation:

- ☐ Schreibe die Datenseite auf beide Platten ist teurer als das Schreiben auf eine Platte (Synchronisation der Plattenarme).
- ☐ Es kann gezeigt werden, daß bei einer Schreiboperation die durchschnittliche Suchzeit bei der Verwendung von k Spiegelpatten

$$1 - \prod_{1 \leq i \leq k} \frac{2i}{2i+1}$$

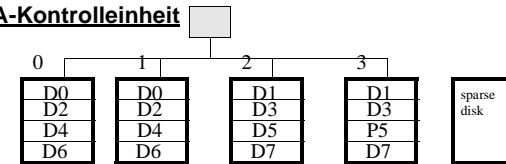
beträgt (relativ zu der Anzahl der Zylinder).

392.

RAID 1

D. Bitton, J. Gray: "Disk Shadowing", Proc. VLDB Conference, 1988, 331-338

E/A-Kontrolleinheit



- ☐ zu jeder Platte gibt es eine Spiegelplatte, auf der eine Kopie der Daten liegt

Abbildung der Blöcke:

- ☐ Gegeben ein Diskarray mit d+1 Platten (enthält eine leere Platte):
- ☐ Block D_i wird auf Position $(2i \div d)$ der Platten $(2i \bmod d)$ und $(2i \bmod d) + 1$ gehalten

391.

- ☐ Insgesamt werden bei einer Schreiboperation eines Blocks also 2 physische Schreiboperationen durchgeführt (wichtig für Durchsatz).

Leistung im Fehlerfall:

- ☐ diejenige Platte, die Partner der defekten Platte ist, hat eine zweimal so hohe E/A-Belastung im Vergleich zu einer Platte, die noch eine Partnerplatte besitzt.

Speichermehraufwand

- ☐ Bei d effektiven Platten im Diskarray werden insgesamt $(2d+1)/d$ Platten für RAID 0 benötigt.

Erwartete Zeit für ein Ausfall des Gesamtsystems:

$$\frac{MTTF}{d} \times \frac{MTTF}{MTTR}$$

393.

Vergleich RAID 5 und RAID 1

- ❑ Speicherkosten
 - RAID 1 ist wesentlich teurer als RAID 5
- ❑ Ausfallsicherheit
 - RAID 1 bietet eine wesentliche höhere Ausfallsicherheit
 - bei paralleler Benutzung vieler Platten empfiehlt es sich ein RAID 5 aufzuteilen
- ❑ Zugriffsgeschwindigkeit im Normalfall
 - für RAID 1 ist das Lesen eines Blocks effizienter
 - sowohl für RAID 1 als auch für RAID 5 ist das Schreiben eines Blocks teuer als bei einem Verzicht auf Fehlertoleranz (\Rightarrow asynchrone E/A, NVRAM)
- ❑ Zugriffsgeschwindigkeit bei Ausfall einer Platte
 - Lesen bei RAID 5 erfordert nun den Zugriff auf alle Platten (Forderung: kleine Anzahl von Platten in RAID 5)
- ❑ Restaurieren des parallelen Plattensystems ist oft aufwendig
- ❑ Kontrolleinheit
 - eine der wichtigsten und teuersten Komponenten eines RAID
 - wegen der Ausfallsicherheit gibt es mehrere (i.a. zwei) Kontrolleinheiten

Zusammenfassung

- ❑ Ziel einer Speicherhierarchie:
Geschwindigkeitsanpassung des jeweils größeren und langsameren Speichers an den schnelleren zu vertretbaren Kosten (Kosteneffektivität)
- ❑ dominierender Speichertyp als Externspeicher: Magnetplatte
 - Berechnung von Speicherkapazität und Zugriffszeit
 - weitere Steigerung der Speicherkapazität erwartet
 - jedoch nur geringfügige Steigerung der Zugriffszeit
- ❑ mittelfristige Konkurrenten der Magnetplatte:
überschreibbare optische Platte und nicht-flüchtiger Halbleiterspeicher
- ❑ Disk-Arrays:
 - neue Organisationsform für Magnetplatten
 - Verbesserung der E/A-Raten bzw. Bandbreiten durch "disk striping"
 - einstellbarer Grad an Fehlertoleranz (RAID n), $n = 0, \dots, 5$.
 - Einzelzugriffe langsamer als bei konventionellen Platten