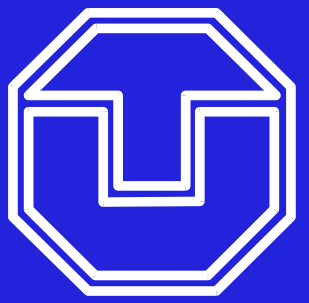
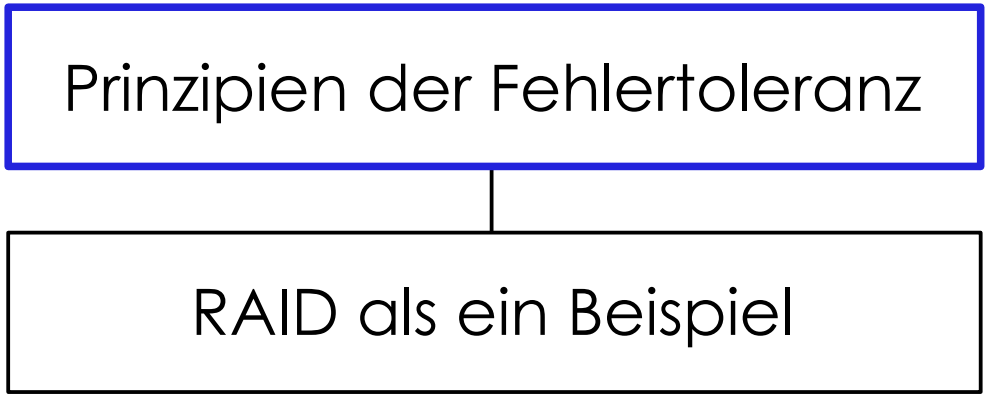
Fehlertoleranz

Betriebssysteme

Hermann Härtig
TU Dresden



Wegweiser



Begriffe

Grundprinzip

Konstruktion zuverlässigerer Systeme aus weniger zuverlässigen Komponenten

Fehler

- Ursprüngliche Wirkung
  z.B. durch kosmische Strahlung verursachte Bit-Flips
- Sichtbarwerdung/Manifestation
  z.B. pointer kracht
- Wirkung
  z.B. falsche Daten werden gespeichert, Rechner stürzt ab
- englische Fachliteratur: Fault, Error, Failure

Fehlerusachen

Entwurfsfehler

- fehlerhafte Anforderungsanalyse
- „Programmier“-Fehler

Ausfälle

- Äußere Effekte
  z.B. durch kosmische Strahlung verursachte Bit-flips
- Alterung (Badewannen-Kurve)

Grundelemente der Fehlertoleranz

- Entdeckung / Detection
- Isolation / Containment
- Behebung / Recovery
- Reparatur / Repair
- Redundanz / Redundancy

Grundelemente der Fehlertoleranz

Entdeckung

- Plausibilitätskontrollen, Prüfsummen
- MMU: Adressierungsfehler

Isolation

- Verhinderung der Ausbreitung von Fehlern
- Erkennen und Abbrechen

Recovery

- Herstellen eines korrekten oder konsistenten Zustands
- forward / backward error correction

Grundelemente der Fehlertoleranz

Reparatur

- Austausch defekter Komponenten und Eingliederung ins System

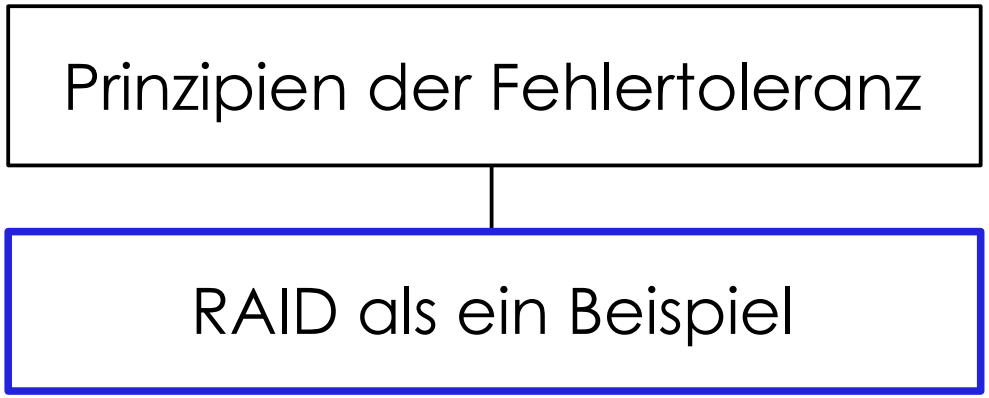
Redundanz

Zusätzliche Ressourcen für die Durchführung der Aufgaben

- Zeit: mehrfache Ausführung von Operationen, Vergleich der Ergebnisse
- Struktur: Nutzung mehrfach ausgelegter Funktionseinheiten (z.B. Speicherung von Daten auf 2 Festplatten)
- Information: Fehler-korrigierende Codes als Bestandteil der gespeicherten Daten

Zusätzliche Funktionalität fügt auch neue Fehlerquellen hinzu!

Wegweiser



Fehlermodell bei Festplatten

- Entwurfs-/Produktionsfehler
- Laufzeitfehler
- Beschädigung von Daten bspw. durch Magnetismus
  Reparatur: neu schreiben – transiente Fehler
- Alterungsfehler, bspw. Ausfall einzelner Blöcke oder des ganzen Laufwerks
  Reparatur: Austausch – permanenter Fehler

Transiente Lesefehler

- Jeder Block auf Platte hat Head/Tail mit redundanter Codierung: Error Correcting Codes (ECC)
- Fehlererkennung und -korrektur durch Auswertung dieses Codes
- Buchführung zu Wartungszwecken (SMART)
- statistische Erkenntnis: Festplatten mit höheren (transienten) Fehlerraten fallen auch bald komplett aus

Permanente Fehler: defekte Blöcke

- Erkennung durch redundante Codes, häufig schon bei Herstellung
- Sektoren werden durch Festplatten-Controller als defekt markiert
- Transparenter Ersatz durch Reserve-Sektoren (Spares)
- früher auch Lösungen in Software (FAT-Dateisystem)
- Fehler-Vermeidung statt -Entdeckung oder -Behebung

Ausfall von ganzen Laufwerken

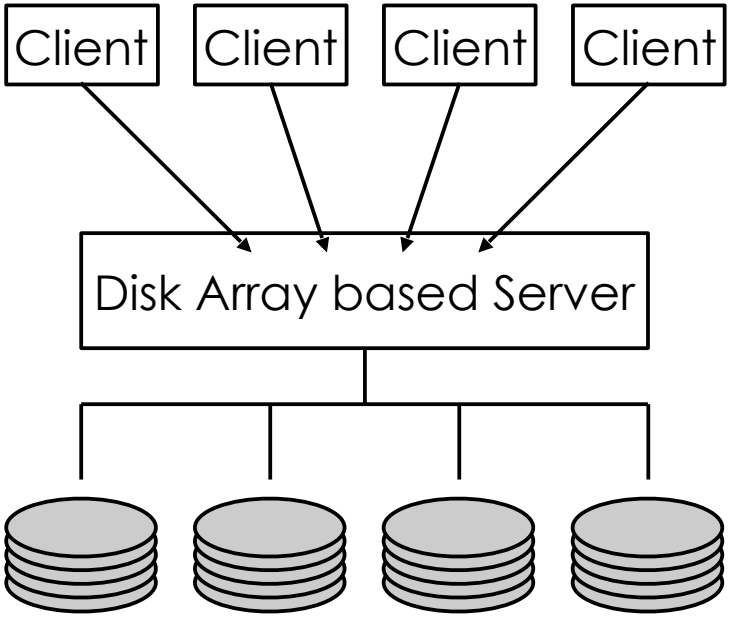
- Entdeckung: Timeout, sich häufende Lesefehler
- Recovery: Redundanz durch RAID
- Reparatur: Plattentausch

RAID: Redundant Array of Independent Disks

Zusammenbau mehrerer unabhängig ansteuerbarer Platten

Ziel

- Bessere Leistung durch parallele Zugriffe
- Ausgleich der Lücke zwischen schnelleren Prozessoren/Speichern und nach wie vor langsamen Platten



Nachteil

- Mittlere Zeit bis zum Ausfall des Feldes ist kleiner
- Maßnahmen zur Fehlertoleranz nötig

RAID 0: Striping

- Daten über n Platten verteilt, paralleles Lesen möglich
- keine Kosten für Redundanz
- keine Fehlertoleranz

Disk 1 . . . Disk n

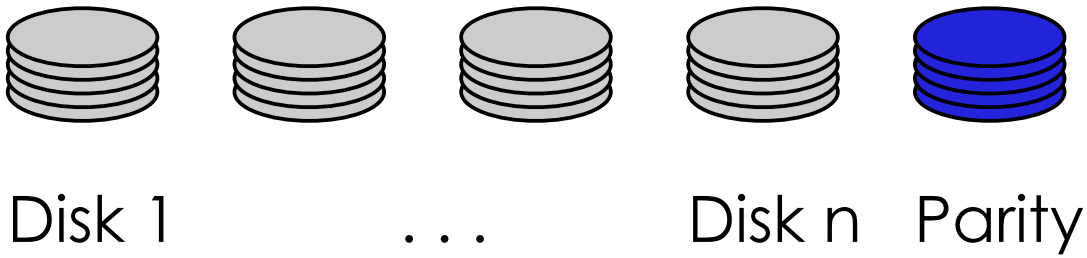
RAID 1: Mirroring

- jede Platte gespiegelt: identische Kopien
- Schreiben: alle Daten zweimal
- Lesen: einmal, von schnellster Platte
- Fehlertolerant (ein beliebiger Ausfall)
- hohe Speicherkosten

Disk 1 . . . Disk n Disk 1' . . . Disk n'

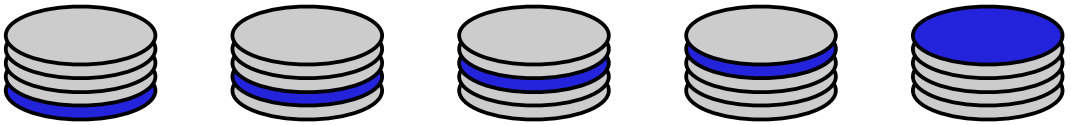
RAID 4: Striping with Parity Drive

- Nutzerdaten werden blockweise auf n Platten verteilt: Dateiblock 1 auf Disk 1, 2 auf 2, …
- blockweise Paritätsbildung:
  (Disk 1, Block 1) XOR (Disk 2, Block 1) … → (Disk P, Block 1)
- Fehlererkennung durch jede Platte individuell, Rekonstruktion durch Paritätsbildung



RAID 5: Striping with Distributed Parity

- Probleme mit RAID 4: Paritäts-Platte nicht ausgelastet beim Lesen, Flaschenhals beim Schreiben
- Block-Interleaved Distributed Parity: Paritätsblöcke über gesamtes Array verteilt



Weitere Techniken

- Double Modular Redundancy
- Triple Modular Redundancy (Flugzeuge)
- Replikation und Konsensbildung (Verteilte Systeme)
- Encoded processing (Redundanz für CPU-Fehler)