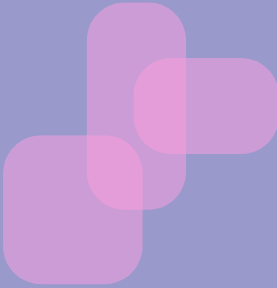
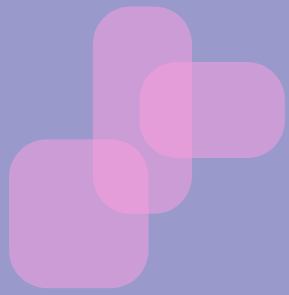


Konzepte von Betriebssystemkomponenten Disk-Caches und Dateizugriff

von

Athanasia Kaisa

- 
- **Grundzüge eines Zwischenspeichers**
 - Verschiedene Arten von Zwischenspeicher
 - Plattenzwischenspeicher in LINUX
 - Dateizugriff unter LINUX
 - Zusammenfassung



Langsame Speichermedien:

- Groß
- Billig

Schnelle Speichermedien:

- Klein
- Teuer



Strategie zur Beschleunigung des Datentransfers

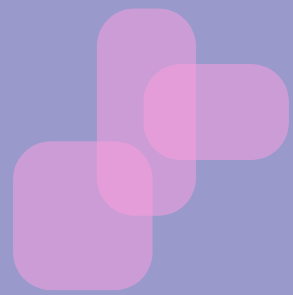
Langsamer:

- Viele Zugriffe
- Geringe Datenmenge pro Zugriff

Abhilfe

Schneller:

- Wenige Zugriffe
- Große Datenmenge pro Zugriff



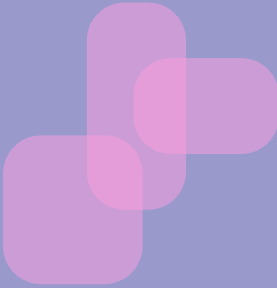
Wie kann die Leistung verbessert werden?

Vorausschauendes Einlesen mehrerer Datenblöcke

Bessere Methoden der Vorausschau

=> bessere Leistung des Rechners, bei richtiger Vorausschau

Vorausschau falsch: Unnötige Zugriffe auf Datenträger

- 
- ✓ Grundzüge eines Zwischenspeichers
 - **Verschiedene Arten von Zwischenspeichern**
 - Plattenzwischenspeicher in LINUX
 - Dateizugriff unter LINUX
 - Zusammenfassung



Was ist Zwischenspeichern?

Schneller Speicher nimmt Teile eines langsamen Speichers auf

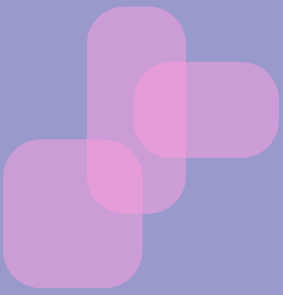
Verschiedene Arten von Zwischenspeichern:

- Hardwarezwischenspeicher
- Softwarezwischenspeicher



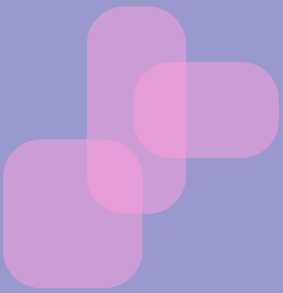
Hardwarezwischenpeicher in der Festplatte

- Bis zu 8 MB
- Enthält kürzlich übertragene Daten
- Mechanische Schreib-/Lesevorgänge werden vom Datenbus getrennt
- Großer Zwischenpeicher => wenige Blocksuchen



Hardwarezwischenpeicher im Festplattencontroller

- Bis zu 256 MB groß
- Zwischen Festplatten- und Softwarezwischenpeicher
- Entlastet SCSI-Bus



Softwarezwischenspeicher

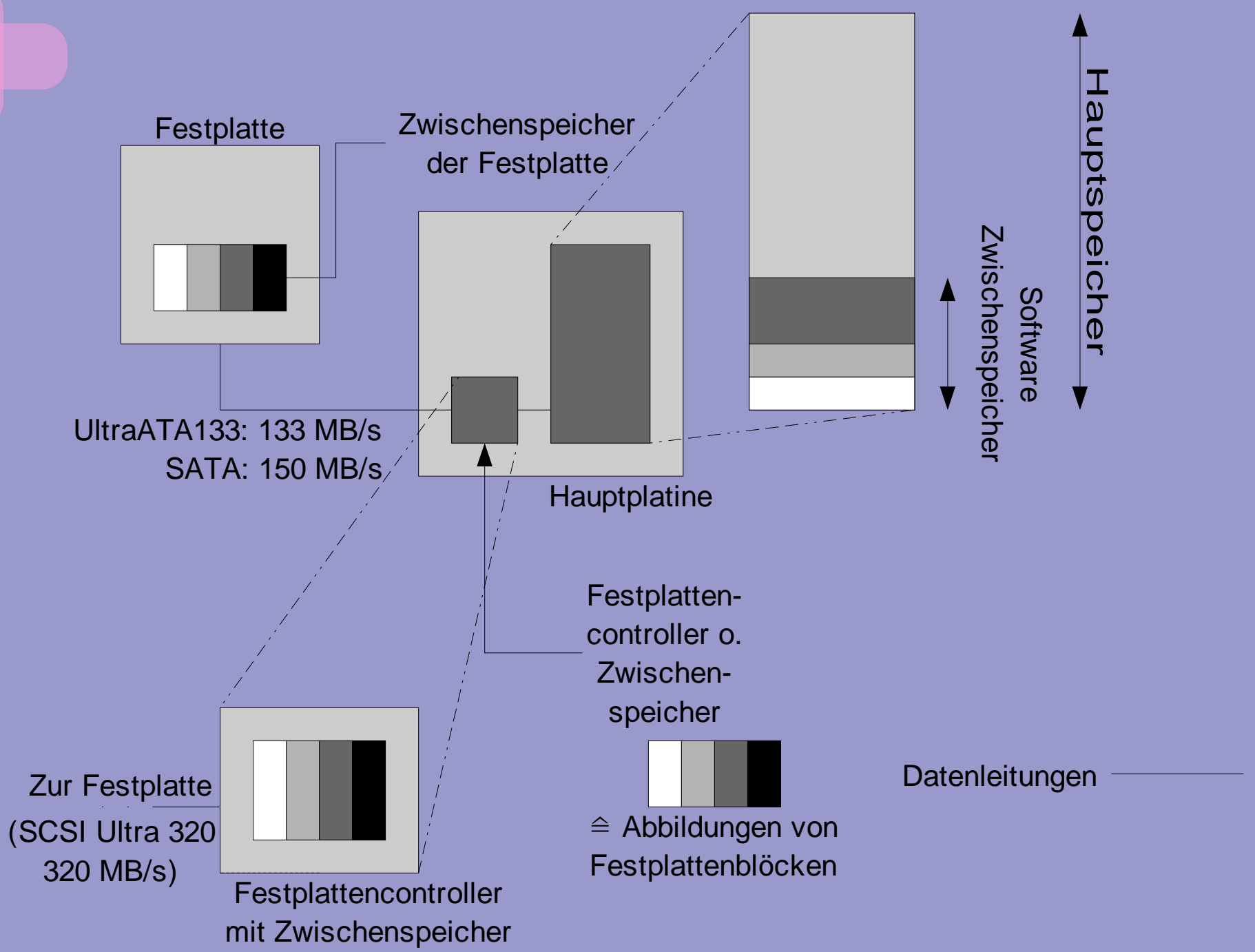
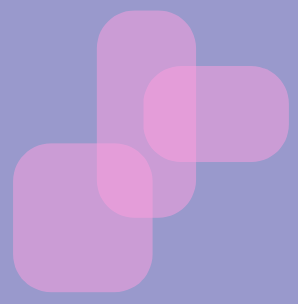
- Größe wird vom Betriebssystem festgelegt
- Wird vom Betriebssystem verwaltet
- Bei wiederholter Anforderung der gleichen Daten Festplattenzugriff nicht nötig

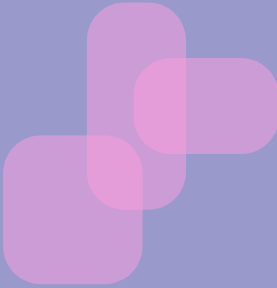


gebräuchlichste Technik:

- Der Blockcache

- Festplattenblöcke werden als Abbild in den Hauptspeicher geladen
- Modifizierte Blöcke müssen auf die Festplatte zurückgeschrieben werden
- Gefahr: modifizierte Blöcke werden nicht immer sofort zurückgeschrieben => Datenverlust bei Stromausfall



- 
- ✓ Grundzüge eines Zwischenspeichers
 - ✓ Verschiedene Arten von Zwischenspeicher
 - **Plattenzwischenspeicher in LINUX**
 - Dateizugriff unter LINUX
 - Zusammenfassung

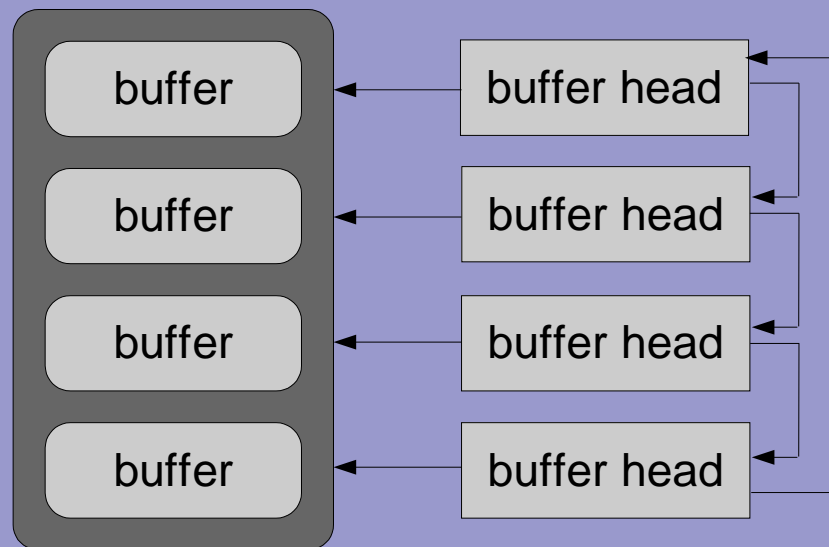


Wie macht es LINUX?

- Benutzt Blockcache
- Blöcke werden durch Pufferköpfe referenziert
- Blockgröße beträgt ein Vielfaches von 512 Byte
- Block
 - Laden => wenn er noch nicht im Zwischenspeicher vorhanden ist
 - Schreiben => wenn er modifiziert wurde
- oft angeforderte Blöcke bleiben länger im Zwischenspeicher

Pufferköpfe

- Datenstruktur
- Zu jedem Puffer ein Pufferkopf
- Enthalten u.a. Speicheradresse des zugehörigen Puffers
- Nicht jeder Pufferkopf muss verwendet werden, da der Kernel sich einen Vorrat von Pufferköpfen hält





Verschiedene Zustände der Pufferköpfe:

- Unbenutzter Pufferkopf
- Pufferkopf für einen freien Puffer
- Pufferkopf für einen verwendeten Puffer
- Asynchroner Pufferkopf



Zurückschreiben der veränderten Puffer

wird von Kernfäden übernommen:

- **kupdate**
 - alle 5 Sekunden werden Superblock- und Inodeinformationen zurückgeschrieben
 - alle 30 Sekunden werden modifizierte Puffer zurückgeschrieben
- **bdflush**
 - standardmäßig alle 5 Sekunden ausgeführt
 - sind 30% der Puffer modifiziert werden max. 64 Puffer zurückgeschrieben
 - sind min. 60% der Puffer modifiziert, wird nicht auf die 5 Sekunden gewartet



Listenstruktur des Zwischenspeichers

- Pufferköpfe werden in verschiedenen Listen verwaltet
- Freie Pufferköpfe werden in doppelt verketteten Ringlisten verwaltet
- Pufferköpfe benutzter Puffer gelangen in LRU-Listen (auch doppelt verkettet)



Zur Verwaltung dienen drei verschiedenen Listen:

BUF_CLEAN

- Pufferköpfe nicht modifizierter Puffer evtl. noch nicht mit Daten gefüllter Puffer

BUF_DIRTY

- Pufferköpfe modifizierter Puffer

BUF_LOCKED

- Pufferköpfe modifizierter Puffer, die ausgewählt wurden, um auf die Platte zurückgeschrieben zu werden



Verwaltung benutzter Pufferköpfe

- Werden in doppelt verketteten Hashlisten gespeichert
- Hashing wird zur Optimierung betrieben
- Puffer werden anhand von Geräte- und Blocknummer auffindig gemacht



Verdrängungsstrategien

- FIFO -> ältester Block im Cache wird entfernt
- Second-Chance -> Variante von FIFO
- LRU -> am längsten nicht mehr benutzter Block wird entfernt

LINUX benutzt LRU



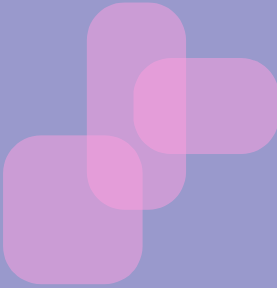
Verwendung des Zwischenspeichers

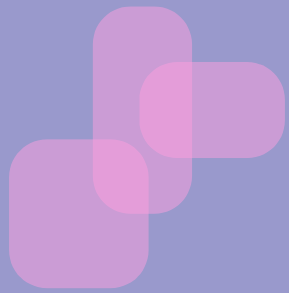
- Lesevorgang
 - Überprüfung ob Daten schon im Zwischenspeicher vorhanden sind
 - Wenn nicht, müssen Daten von der Festplatte gelesen werden
 - Prozesse müssen auf Beendigung des Lesevorgangs warten



- Schreibvorgang

- Durchsuchen des Zwischenspeichers nach modifizierten Puffern
- Pro gefundenem Puffer wird eine Schreib Anforderung an entsprechende Gerätetreiber gestellt
- Vorteil: Prozesse müssen nicht warten bis Schreibvorgang beendet ist

- 
- ✓ Grundzüge eines Zwischenspeichers
 - ✓ Verschiedene Arten von Zwischenspeicher
 - ✓ Plattenzwischenspeicher in LINUX
 - **Dateizugriff unter LINUX**
 - Zusammenfassung



Dateizugriff unter LINUX

- Bei Zugriff auf eine Datei, wird sofort das ganze Verzeichnis eingelesen und Informationen zu jedem Verzeichniseintrag gespeichert
- Diese gelangen in den Verzeichniscache
- Die Einträge werden DEntry genannt