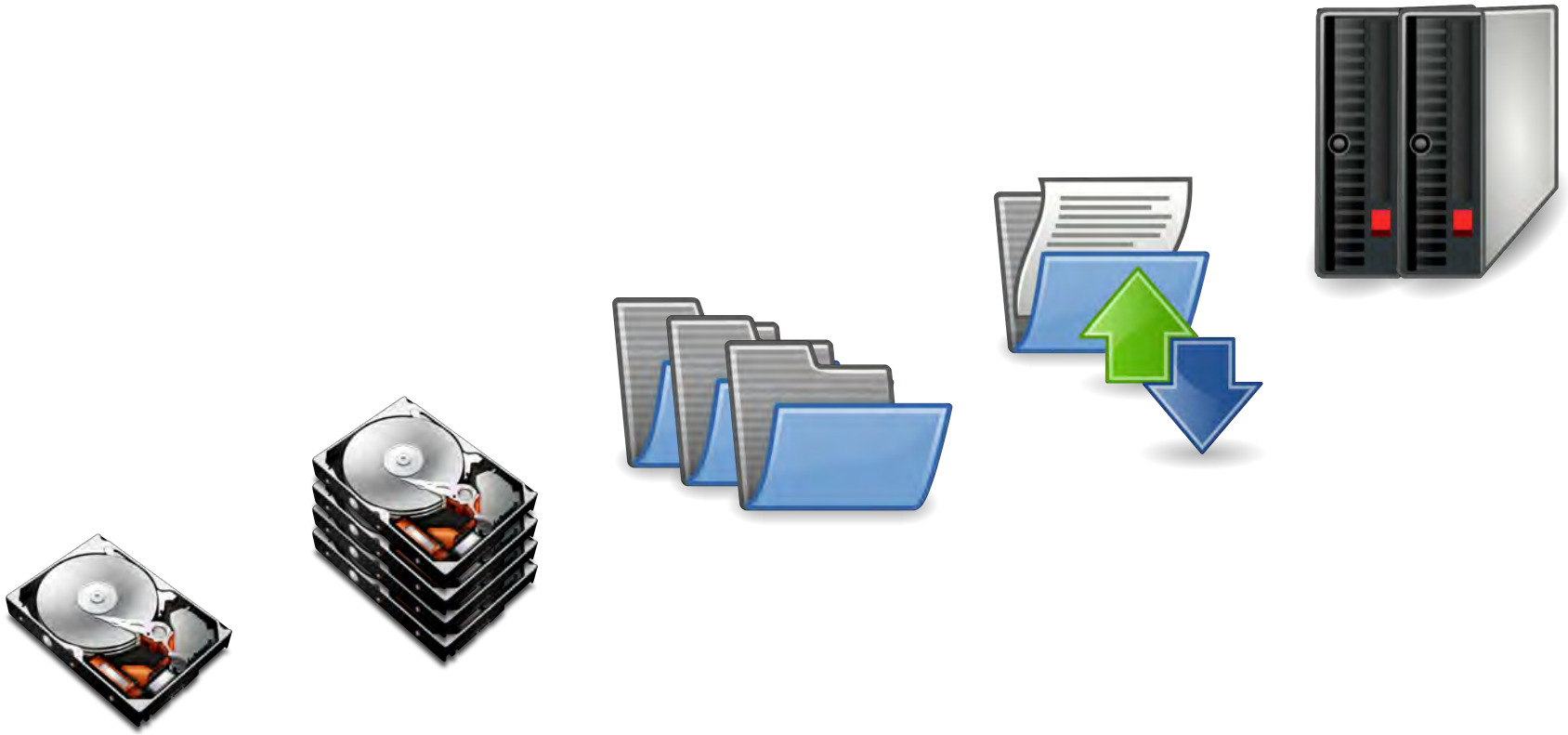
REGIONALES RECHENZENTRUM ERLANGEN [RRZE]

Storage / Filesysteme

Systemausbildung − Grundlagen und Aspekte von Betriebssystemen und System-nahen Diensten

Gregor Longariva, Marcel Ritter, 17.05.2017

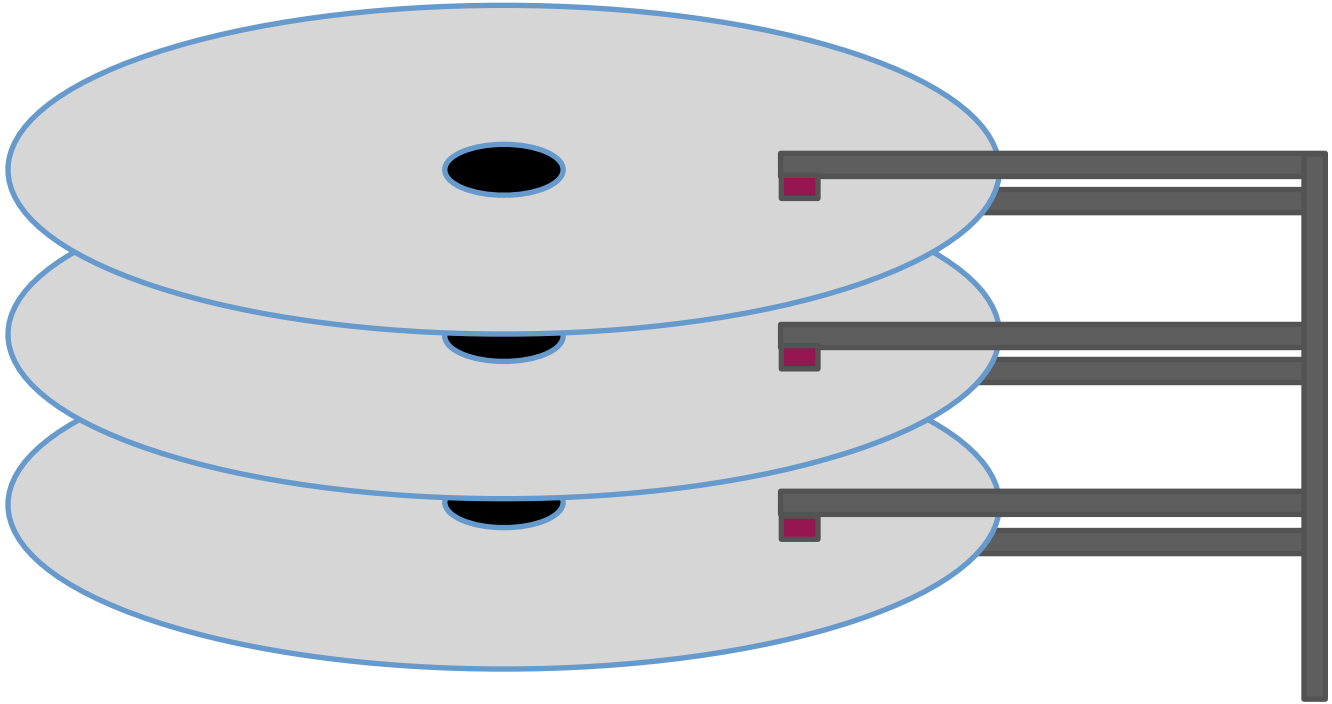
Agenda



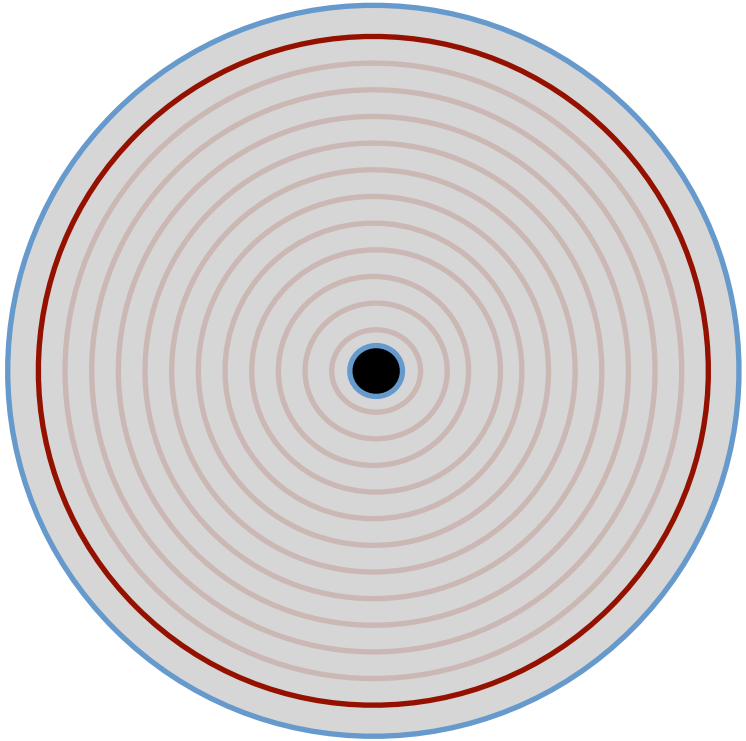
FESTPLATTEN

Prinzipieller Aufbau

Aufbau einer Festplatte

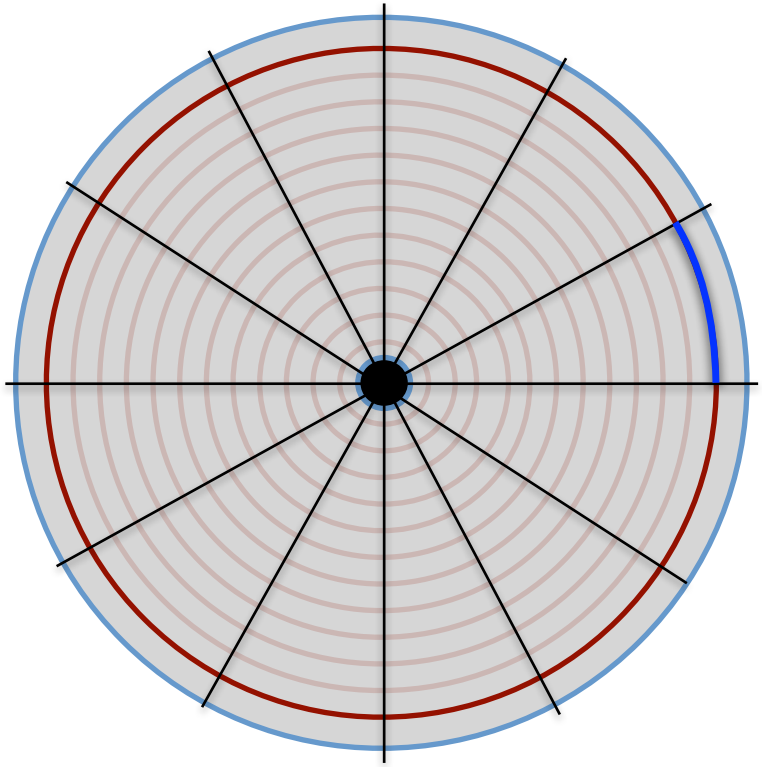


Aufbau einer Festplatte



Spur

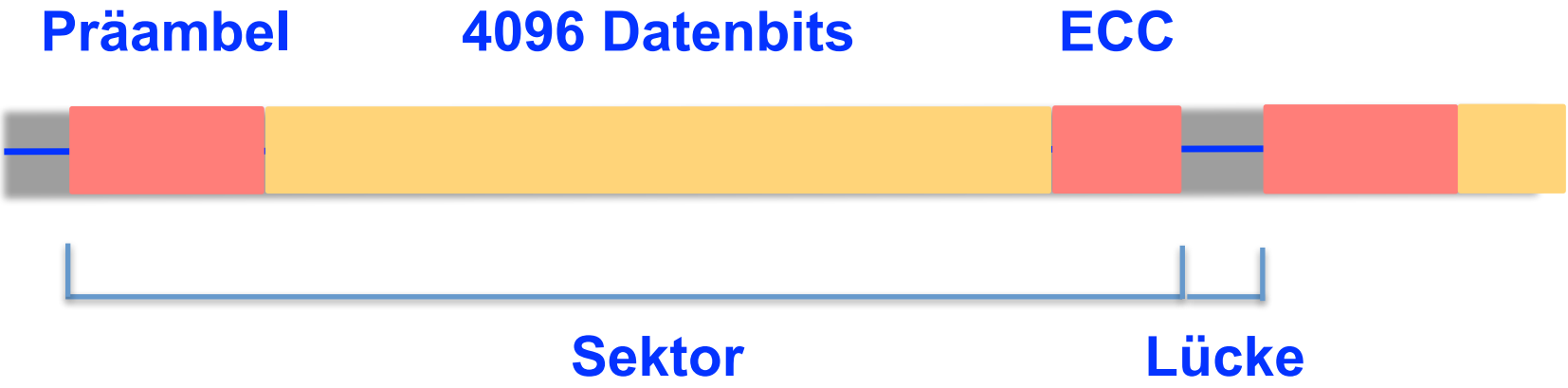
Aufbau einer Festplatte



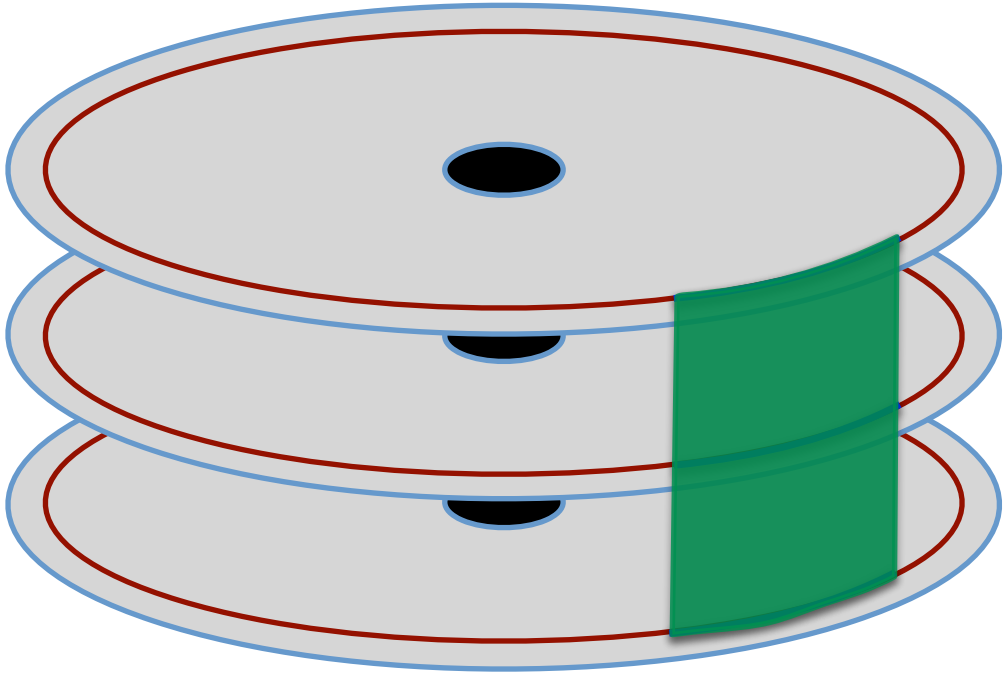
Spur

Sektor

Aufbau einer Festplatte



Aufbau einer Festplatte

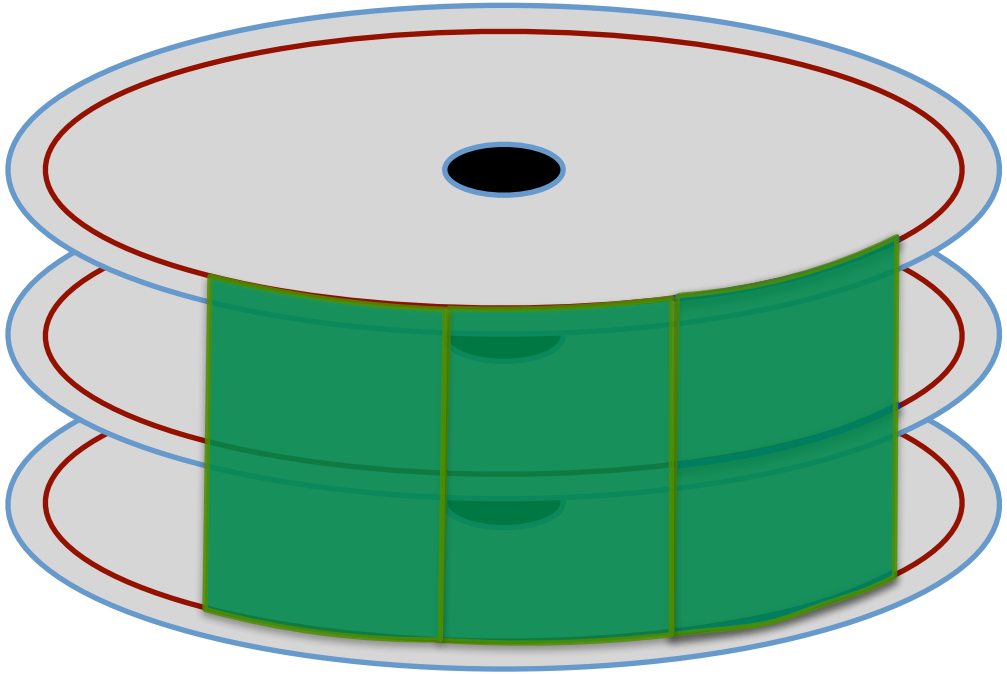


Spur

Sektor

Zylinder

Aufbau einer Festplatte

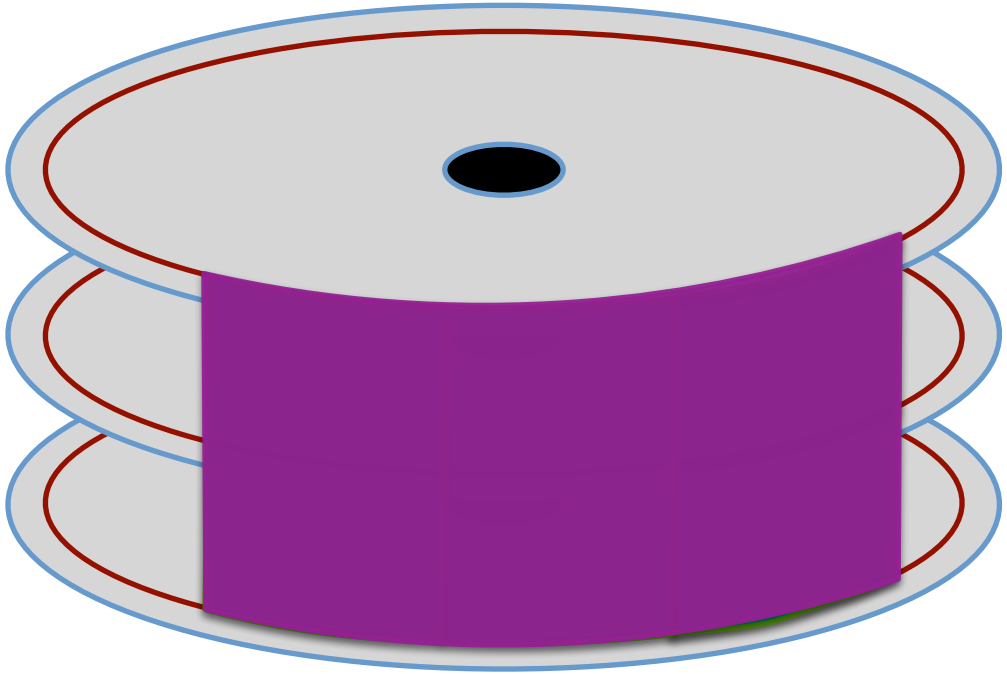


Spur

Sektor

Zylinder

Aufbau einer Festplatte



Spur

Sektor

Zylinder

Cluster

Wie schnell ist eine Platte (worst case)?

Festplatte mit 15k (= **15.000** Umdrehungen / Min)

Latenz: 60 sec / **15.000** = **0,004** sec -> 4ms

IOPS: 1 Operation / **0,004** sec = **250** Ops / sec

Bandbreite: **250** x 4096 Bytes pro Sektor = **1.024.000** bytes / sec

1MByte pro Sekunde!

Wie schnell ist eine Platte (best case)?

1.024.000 bytes / sec x 6 Köpfe = **6.144.000** Bytes / sec

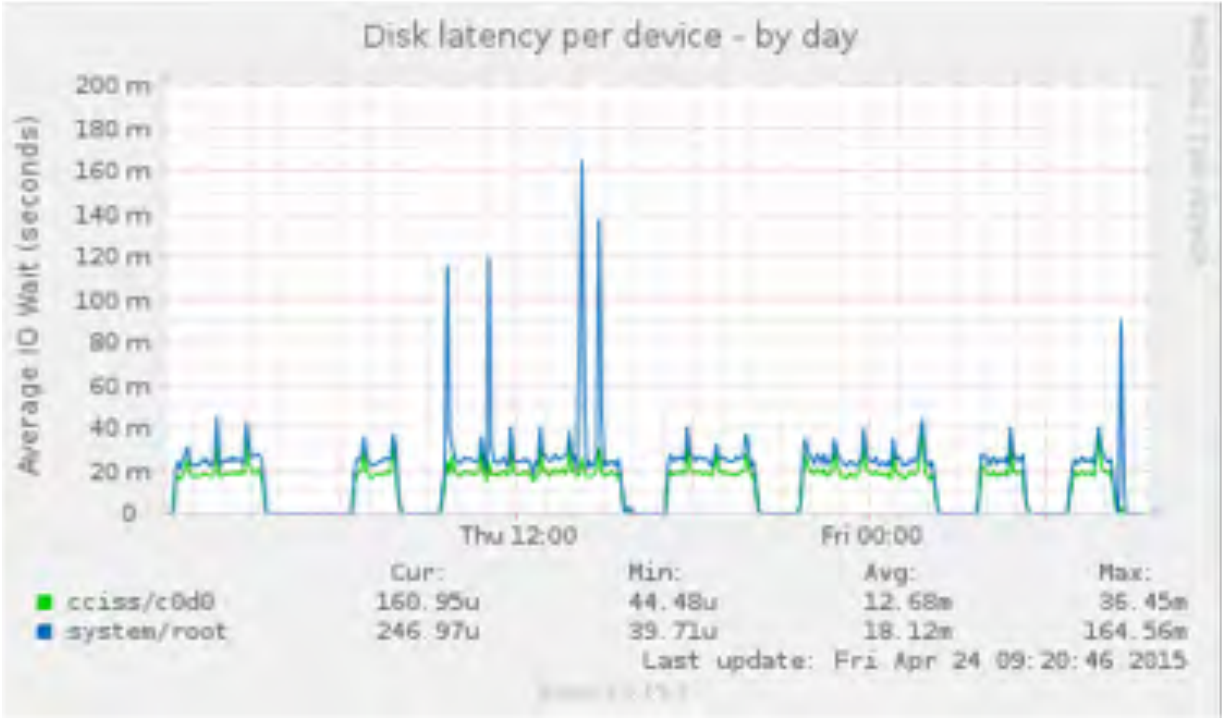
6.144.000 Bytes / sec 30 (Zylinder pro Cluster) =

184.320.000 Bytes / sec

also ca. 180 MByte pro Sekunde

(aber immer noch ohne Plattencaches)

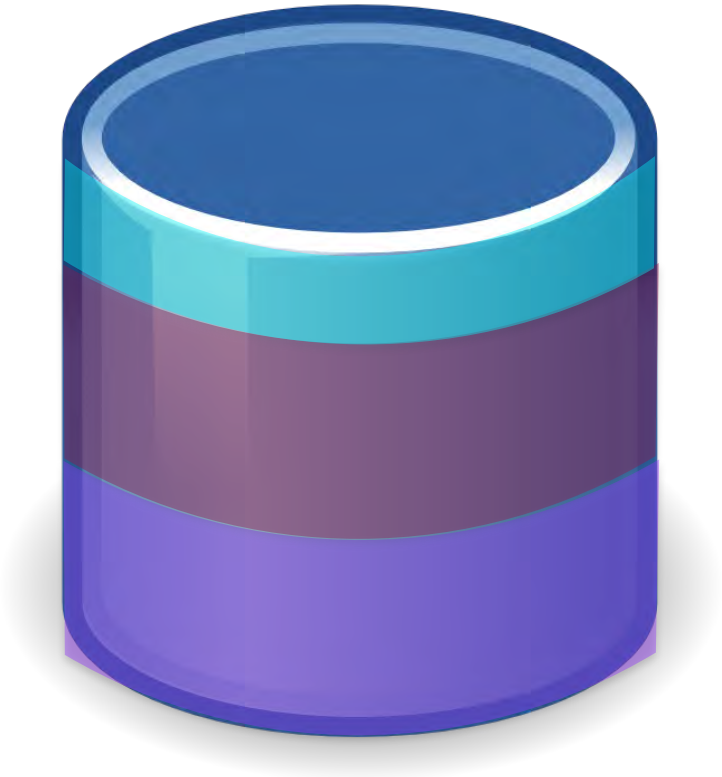
Plattenzugriffe beschleunigen - Cache



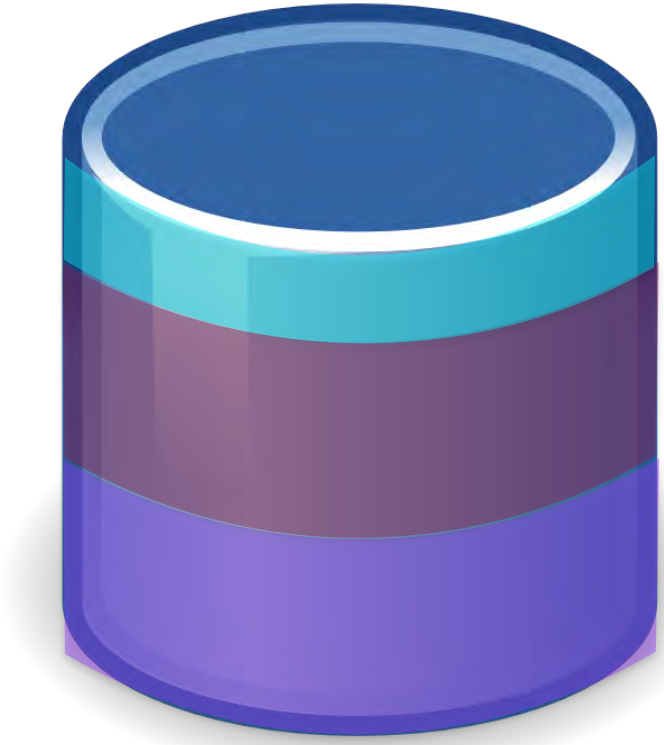
FESTPLATTEN

Partitionierung

Partitionieren



Partitionieren – warum?



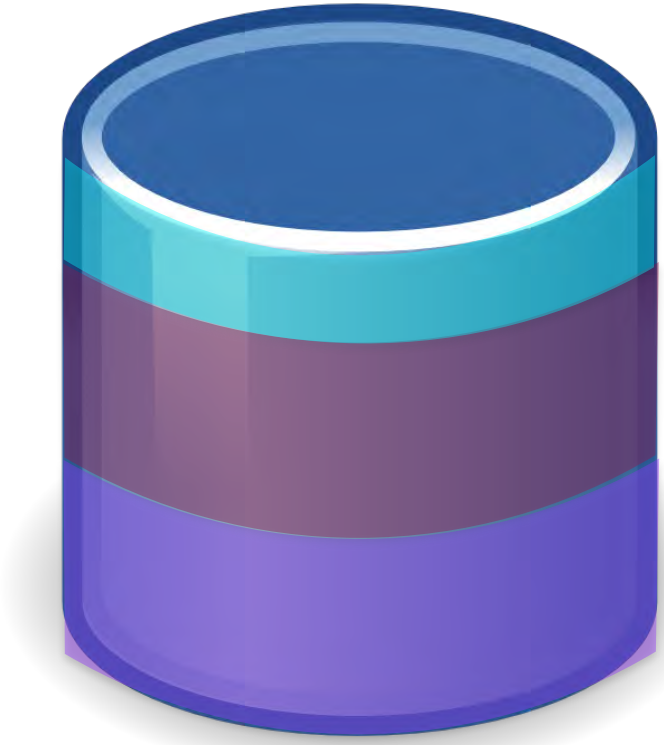
FreeBSD

Linux

Windows

verschiedene Betriebssysteme

Partitionieren – warum?



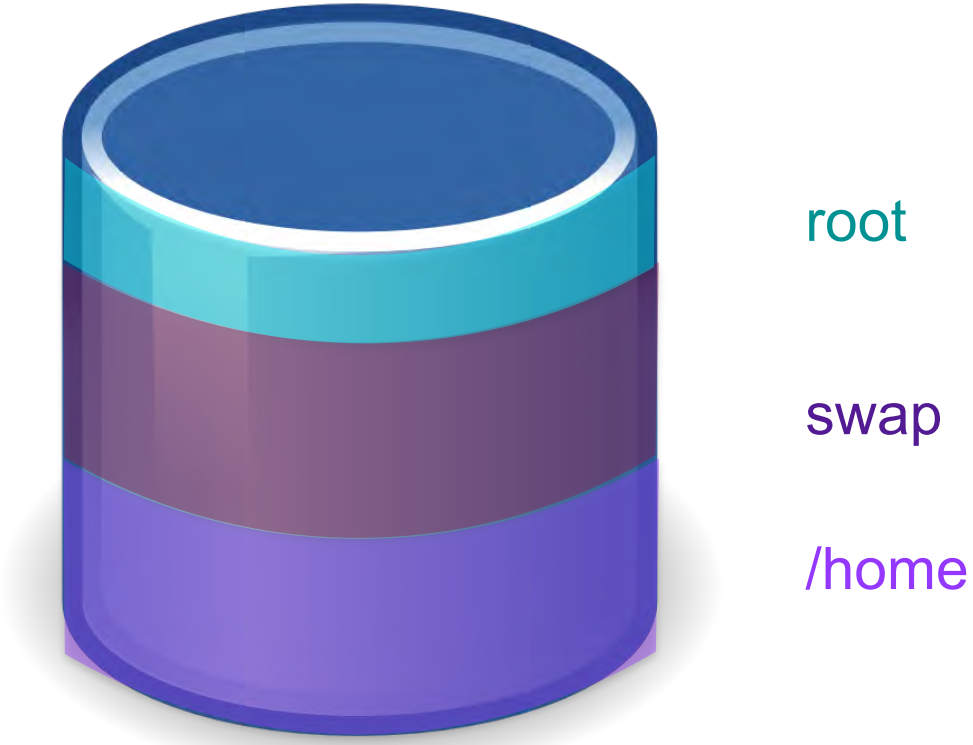
Fotos

Filme

Windows

Trennung Daten und Betriebssystem

Partitionieren – warum?



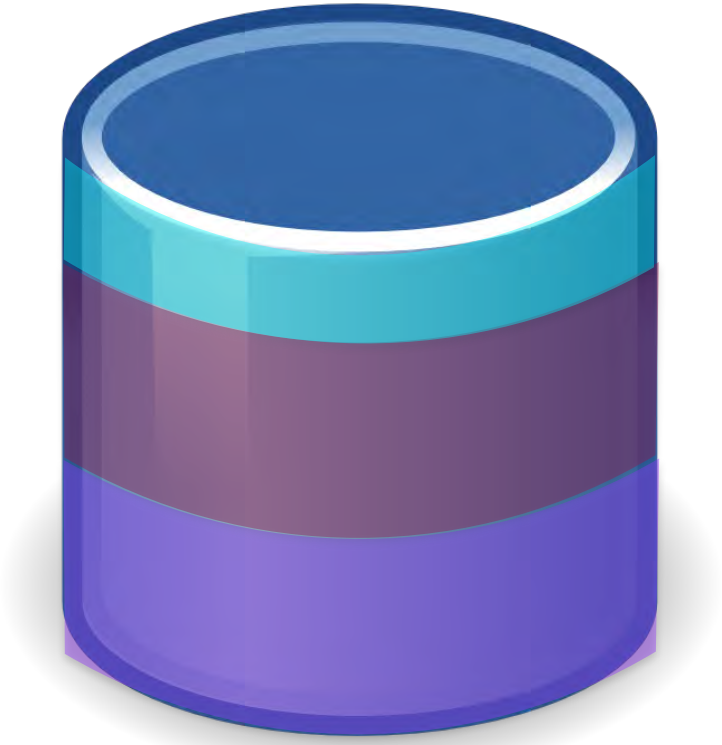
root

swap

/home

verschiedene Bereiche eines OS

Partitionieren – warum?



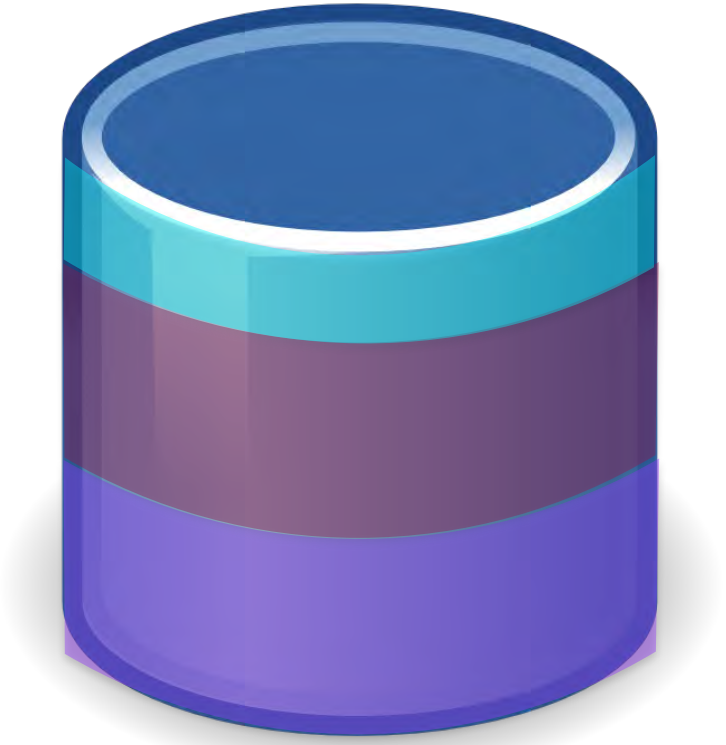
Backup

Windos 8 Devel

Windows 8

Arbeitskopien und Backups

Partitionieren – warum?



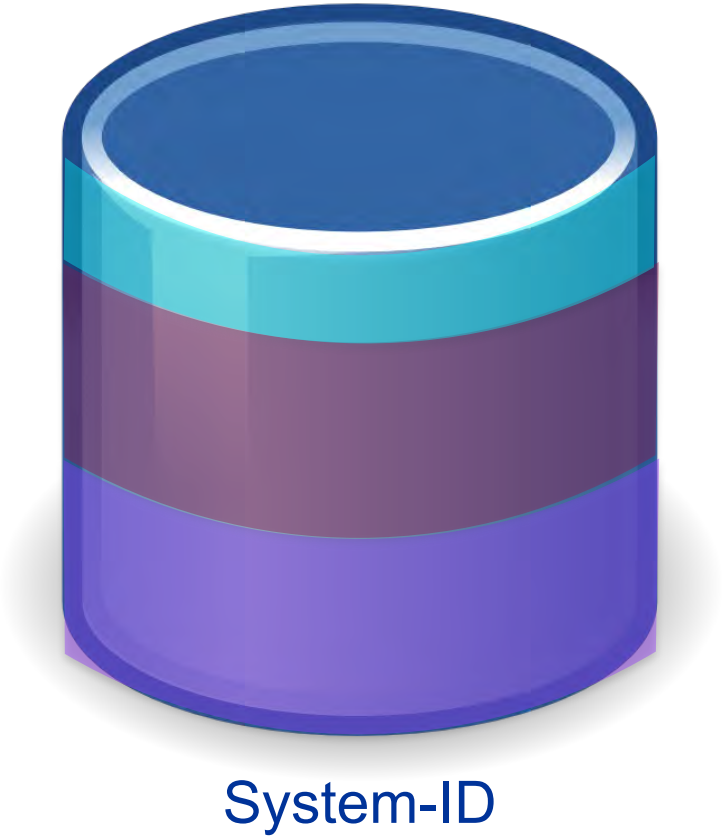
MS-DOS

Win 95a

Windows 8

Verkleinern der Platte

Partitionieren



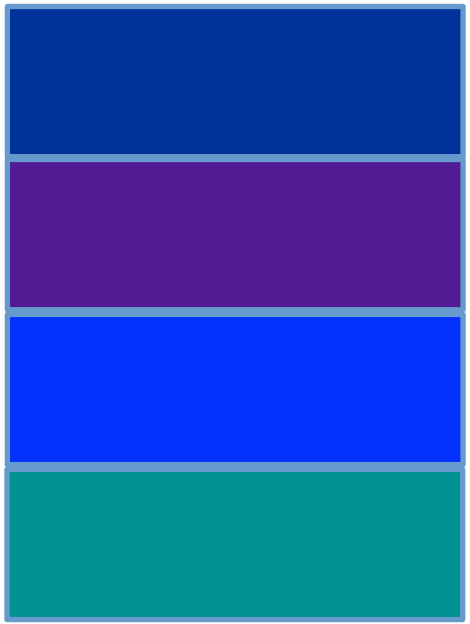
87 - NTFS

bf - Solaris

83 - Linux

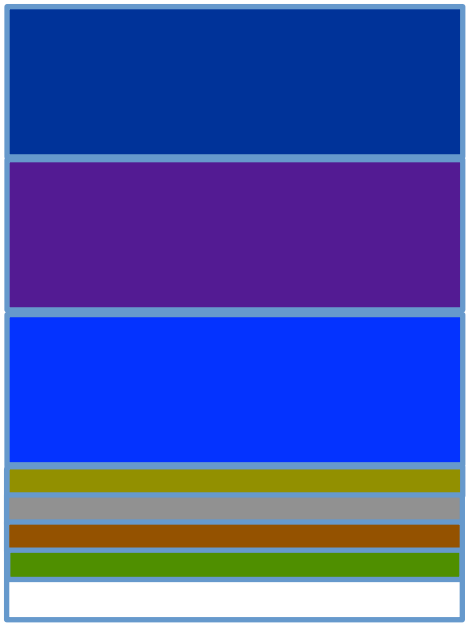
System-ID

Partitionen am PC



4 Primärpartitionen

oder



3 Primärpartitionen
beliebige erweiterte Partitionen

Klassischer Bootsektor MBR vs. GPT

MBR	GPT
BIOS	EFI
512 Bytes	min. 16 384 Bytes
eine Partitionstabelle	Primäre Partitionstabelle
	Backup Partitionstabelle

Partitionen anderer Systeme (Solaris)

```
        label - write partition map and label to the disk
        !<cmd> - execute <cmd>, then return
        quit
partition> p
Current partition table (original):
Total disk cylinders available: 14087 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 14086      136.71GB    (14087/0/0) 286698624
  1 unassigned    wu       0                0         (0/0/0)             0
  2     backup    wu       0 - 14086      136.71GB    (14087/0/0) 286698624
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0

partition>
```

PLATTEN ZUSAMMENFASSEN

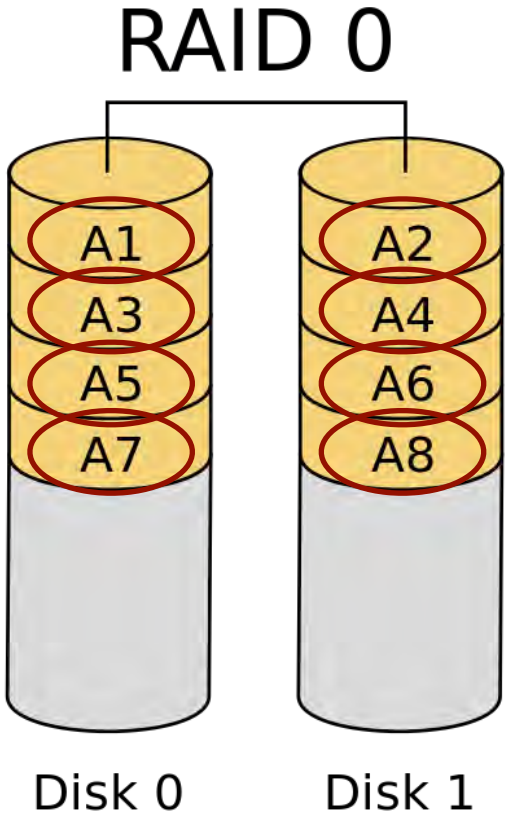
RAID -
Redundant Array of Independent Disks

Warum RAID

mehr Speicherplatz

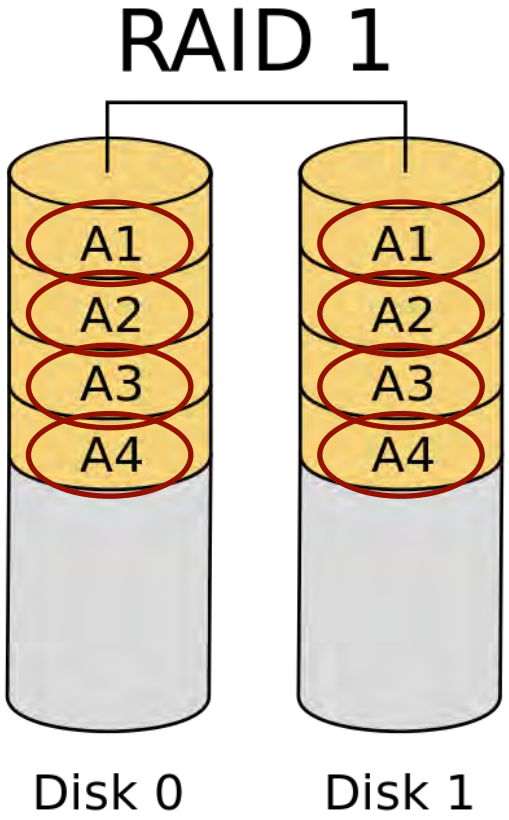
Sicherheit gegen Datenverlust*

RAID 0



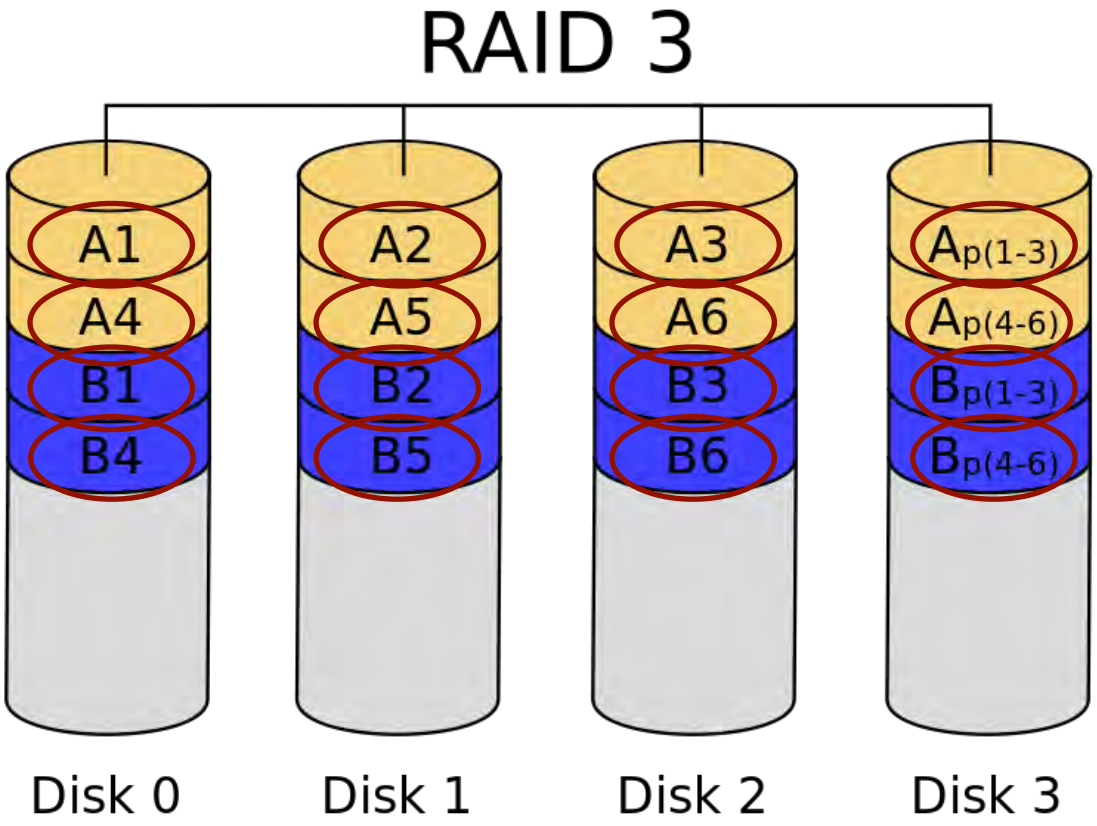
Quelle: Wikimedia

RAID 1 - Mirror



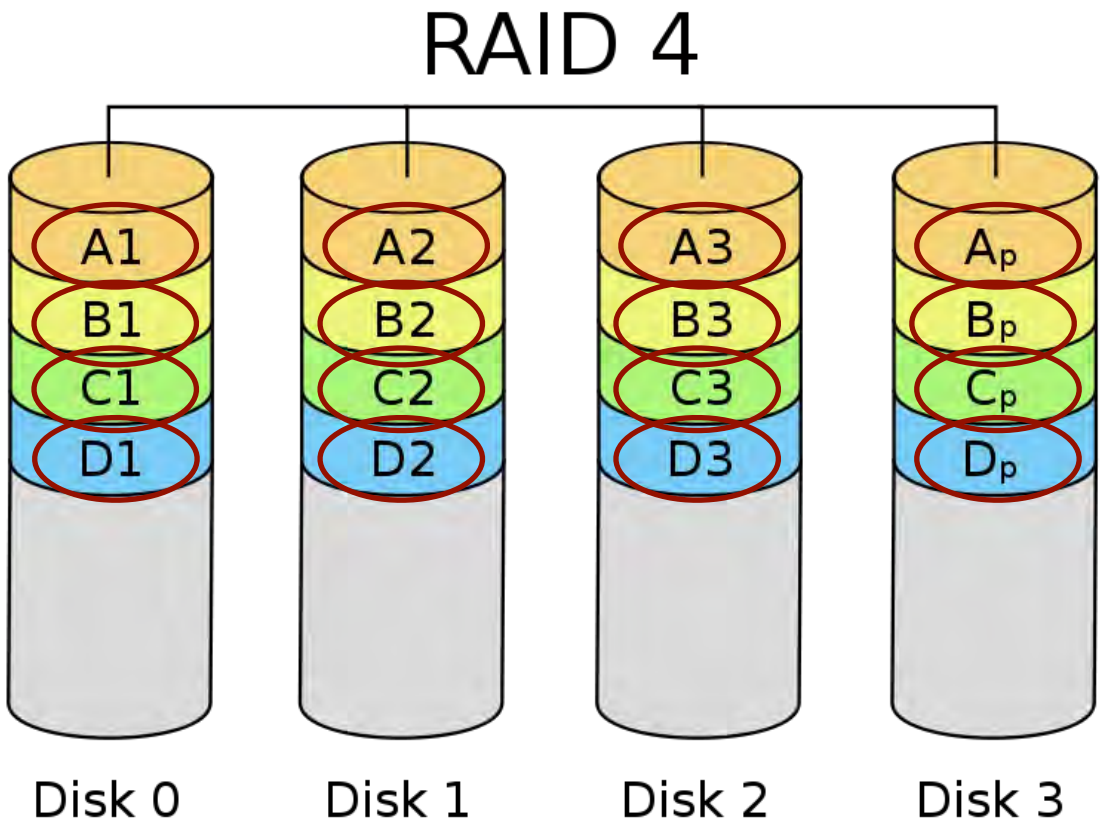
Quelle: Wikimedia

RAID 3



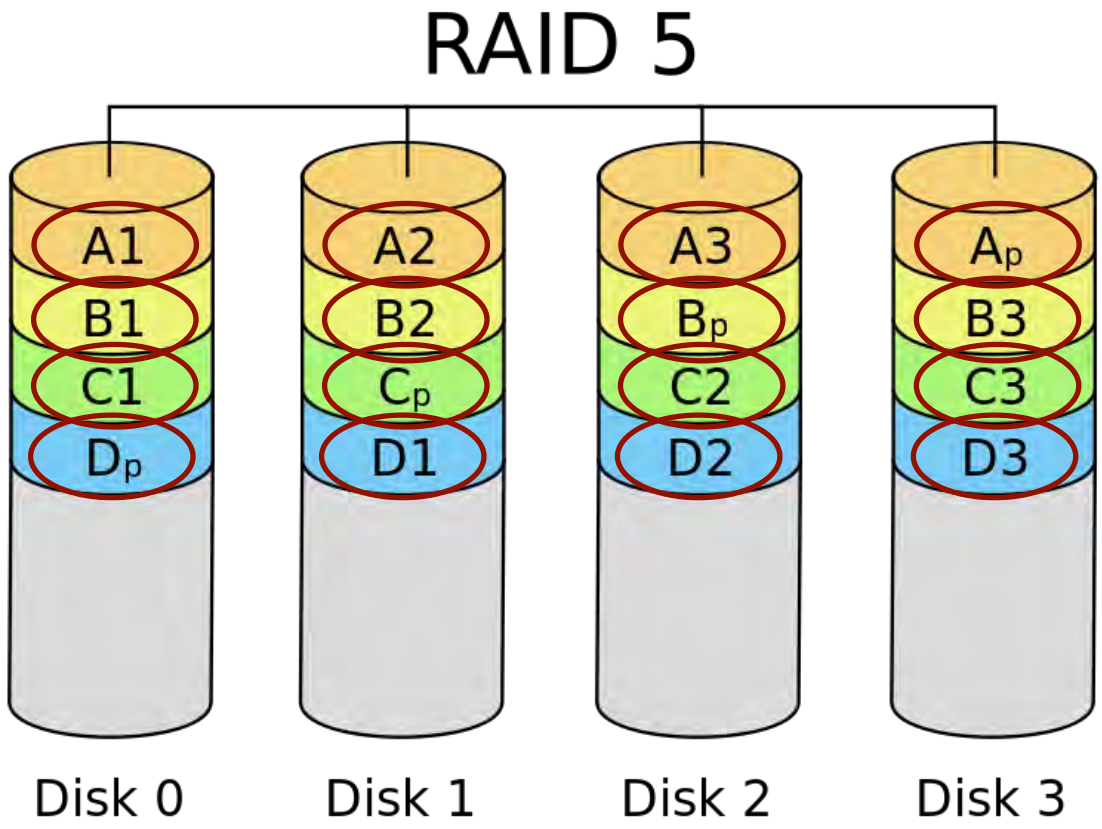
Quelle: Wikimedia

RAID 4



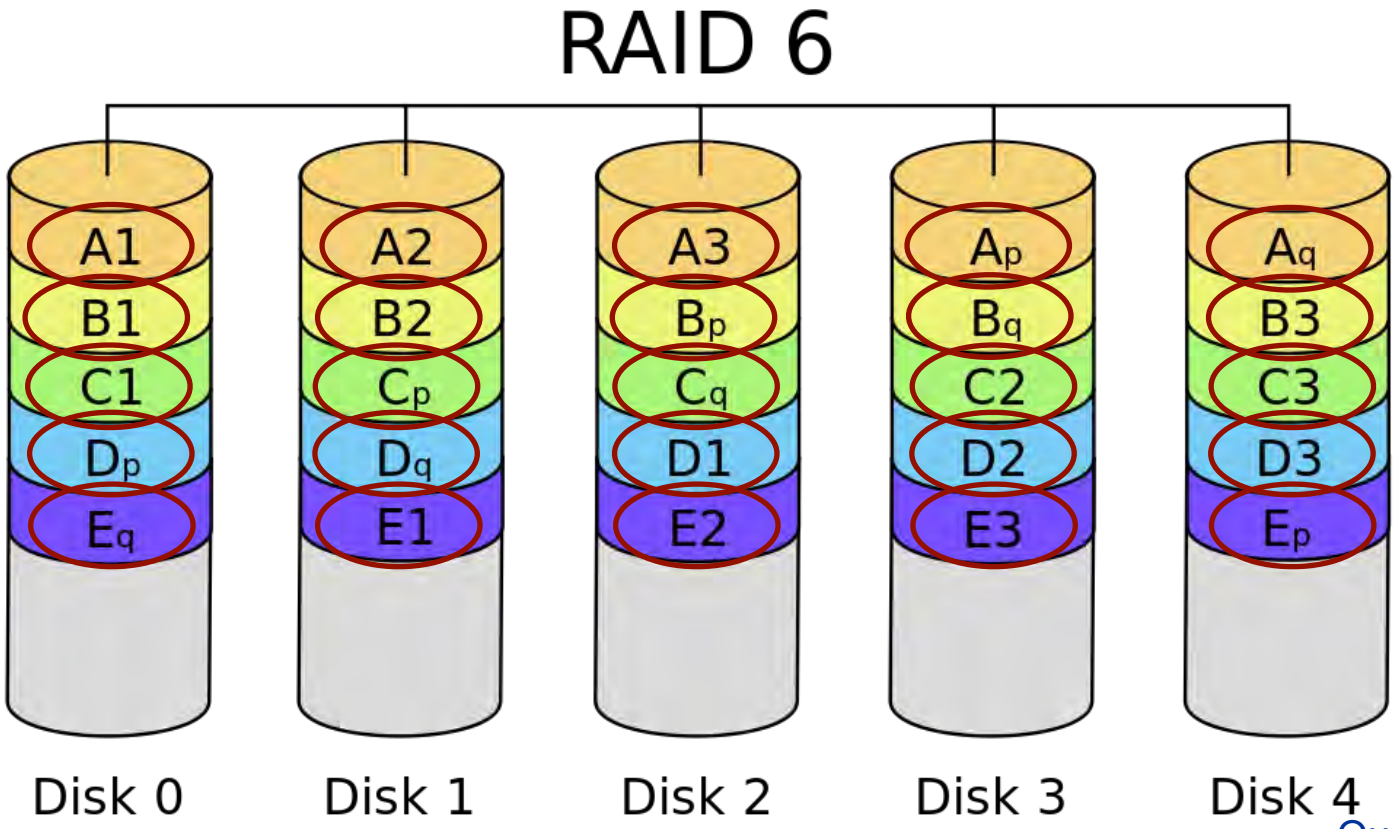
Quelle: Wikimedia

RAID 5



Quelle: Wikimedia

RAID 6



Quelle: Wikimedia

RAID 5 + HotSpare oder RAID 6?

- Verschnitt an Speicherplatz ist gleich
- RAID5: Hotspare wird „geschont“
- Im Fall eines Plattendefekts:
  - RAID5 keine Redundanz (entspricht langsames Raid0)
  - Nach Einspringen der HotSpare werden alle Daten von allen verbliebenen, intakten Platte gelesen um Parity neu zu berechnen
  - Treten Lesefehler auf, ist Rebuild ohne Datenverlust unmöglich
  - Zeitfenster für Rebuild bei großen Festplatten enorm (2 TB bei 100 MB/s = 6 Stunden!)
  - Fehlerwahrscheinlichkeit durch atypisches Lesen aller Disks ebenfalls

RAID 5 + HotSpare oder RAID 6?

Fazit:

RAID 6 ist RAID 5 + HotSpare vorzuziehen

DATEISYSTEME

Speicherung von Daten

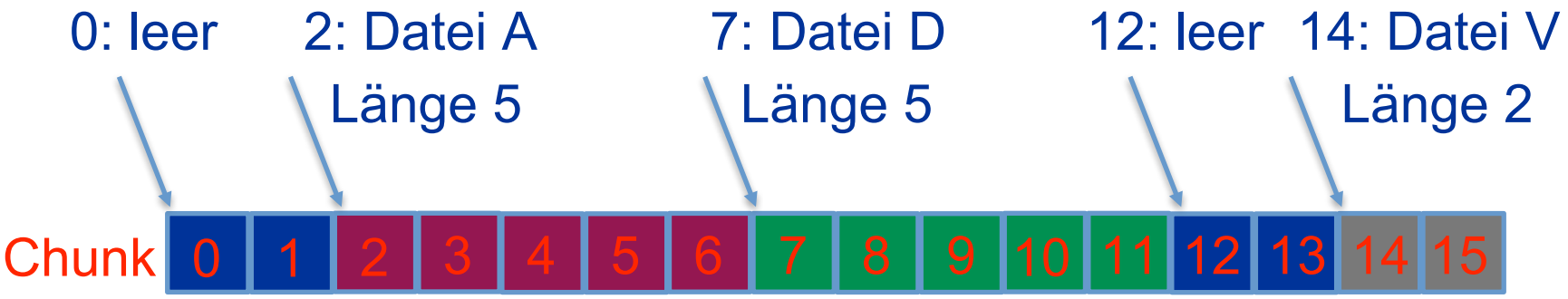
Probleme beim Speichern von Daten

Dateisysteme verwenden Cluster

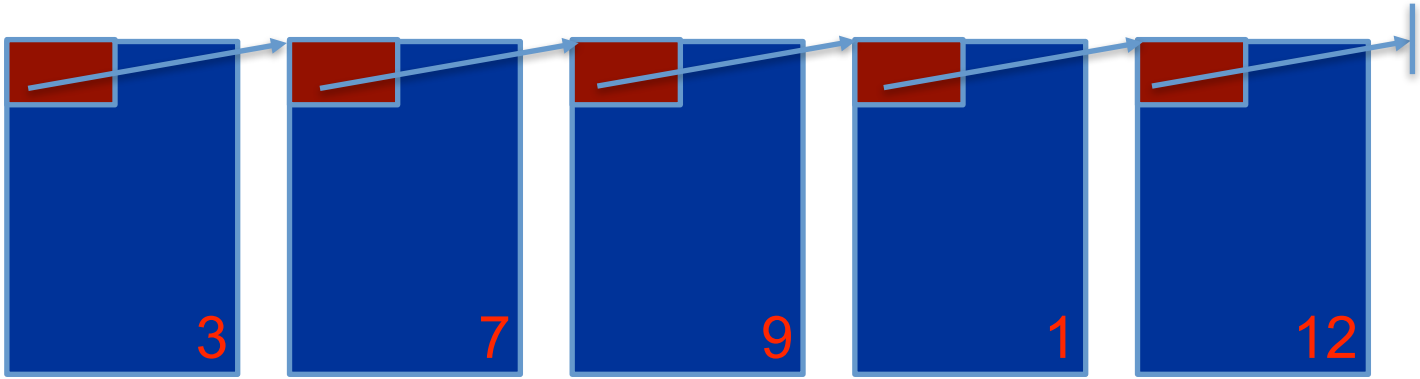
Dateien sind oft größer (oder kleiner) als ein Cluster

Wie kann man nun gespeicherte Daten adressieren?

Kontinuierliche Speicherung

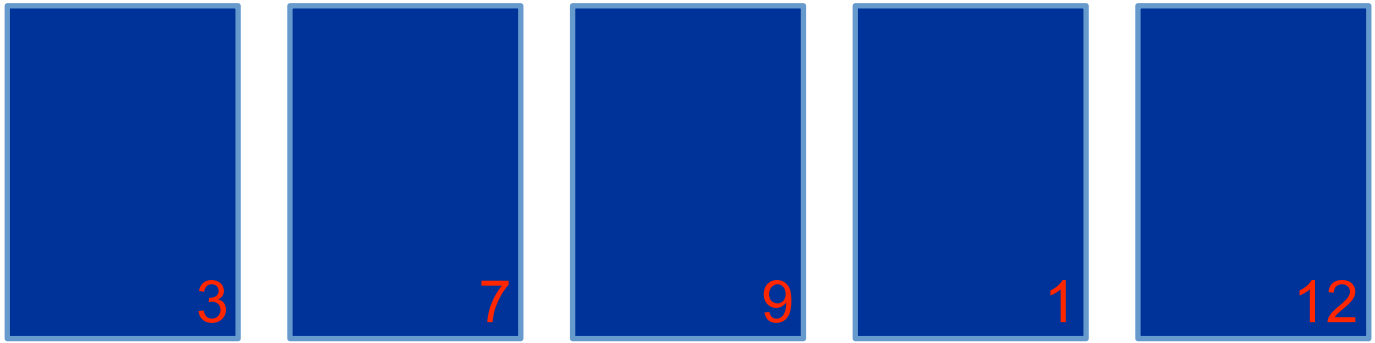


Verkettete Speicherung



Indizierte Speicherung

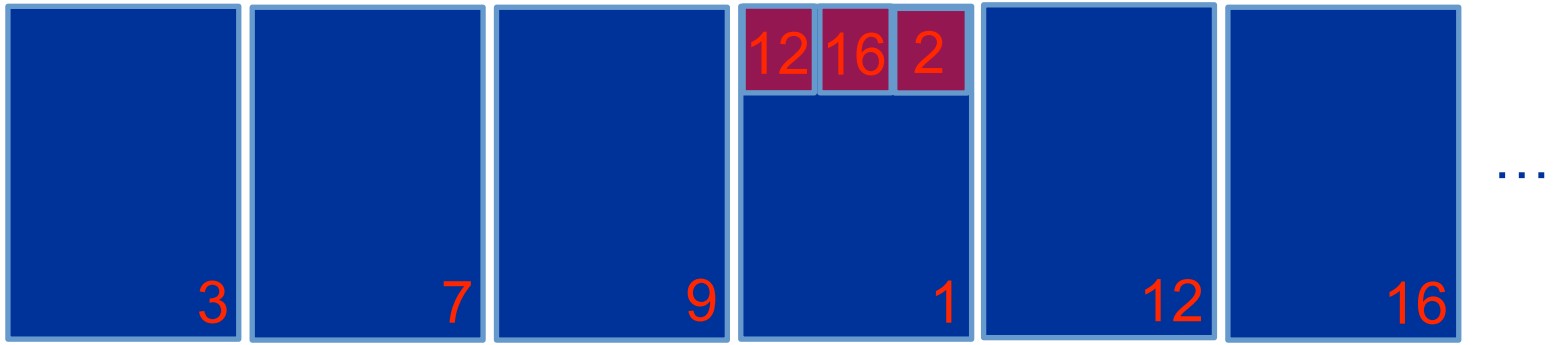
Index-Cluster



Daten-Cluster der Datei

Indizierte Speicherung, mehrstufige Indizierung

Index-Cluster



Daten-Cluster mit einem zusätzlichen Index Cluster

DATEISYSTEME

Beispiele anhand gängiger Dateisysteme

FAT

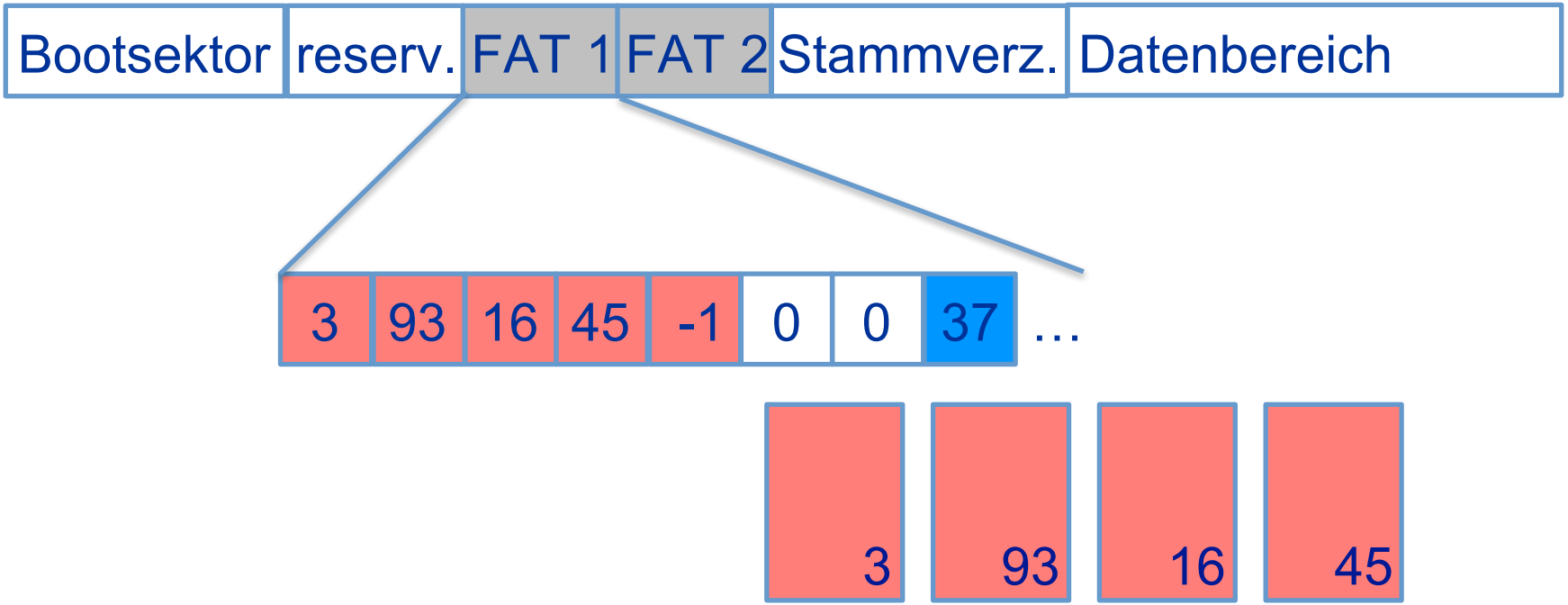
FAT

FAT

FAT

FAT

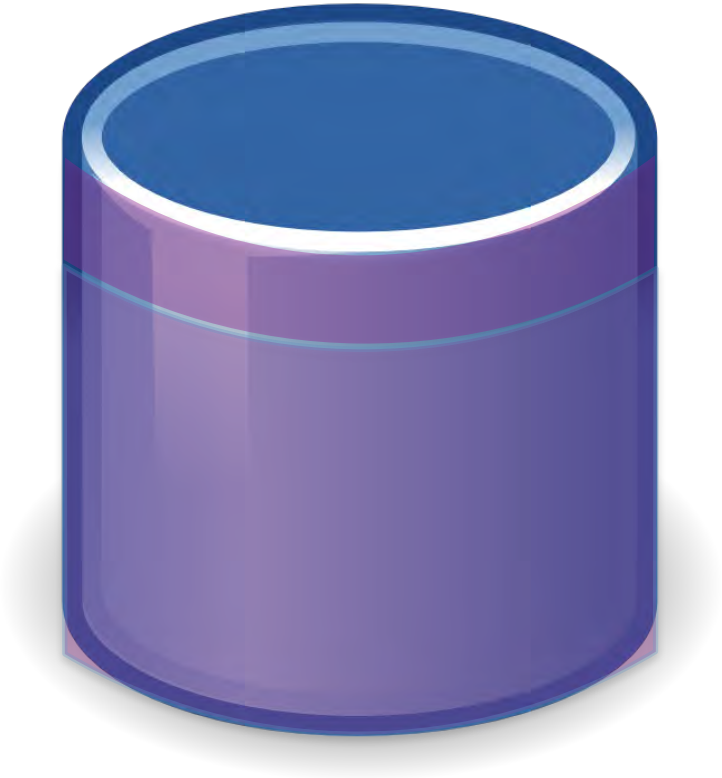
FAT


Bootsektor
reserv.
FAT 1
FAT 2
Stammverz.
Datenbereich
3
93
16
45
-1
0
0
37
...
3
93
16
45

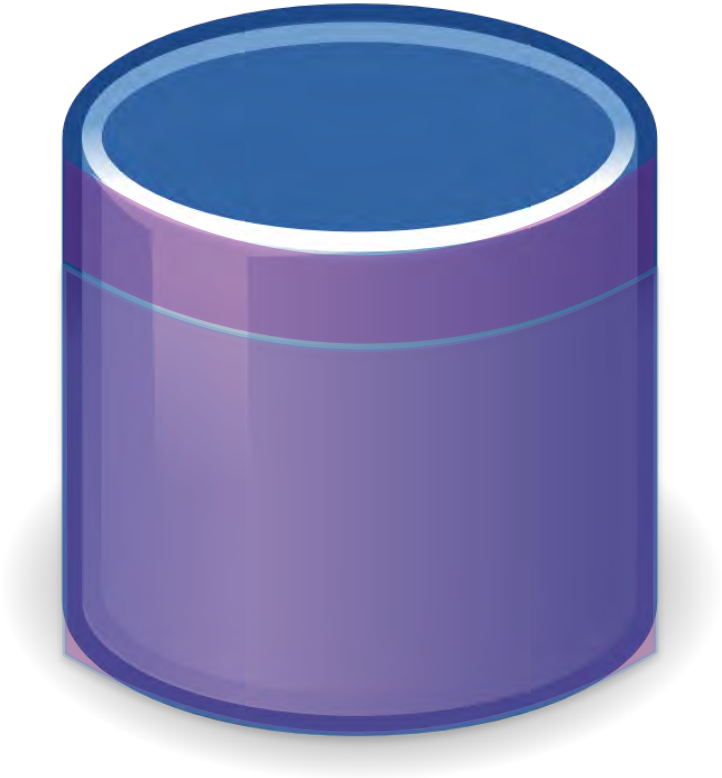
FAT


Bootsektor
reserv.
FAT 1
FAT 2
Stammverz.
Datenbereich
Dateiname
Erweit.
Attrib.
erstellt
letzt. Zugr.
letzt. Änd.
Start Clust.
Größe

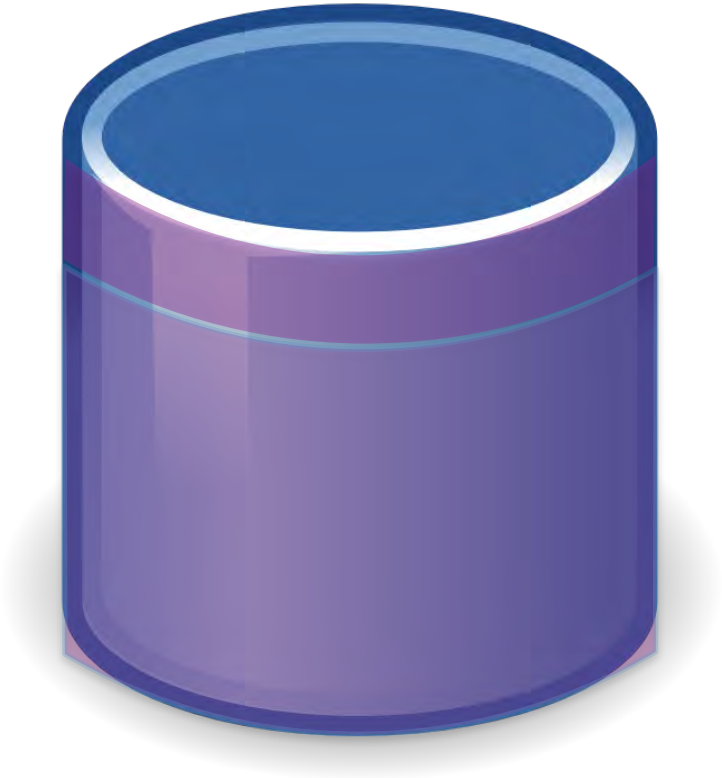
NTFS - Next Technology File System



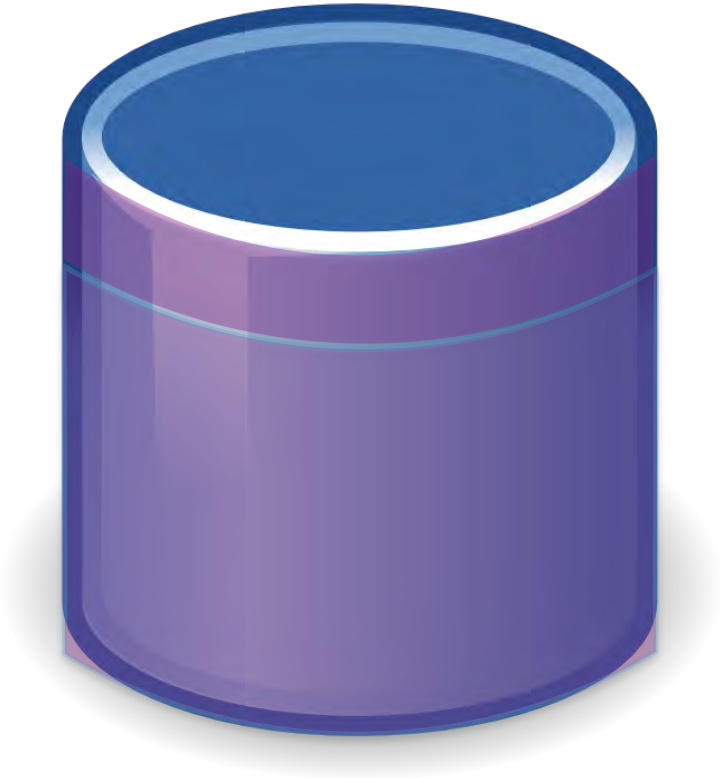
NTFS - Next Technology File System



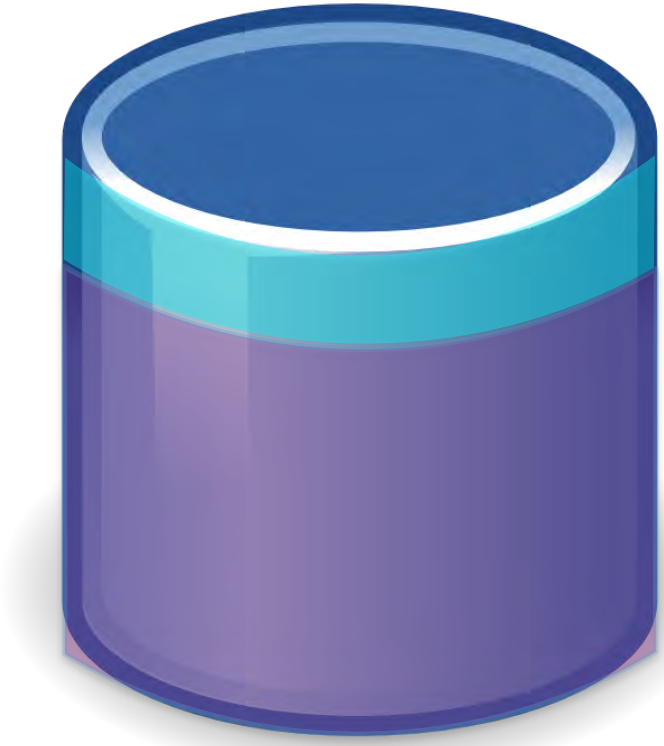
NTFS - Next Technology File System



NTFS - Next Technology File System



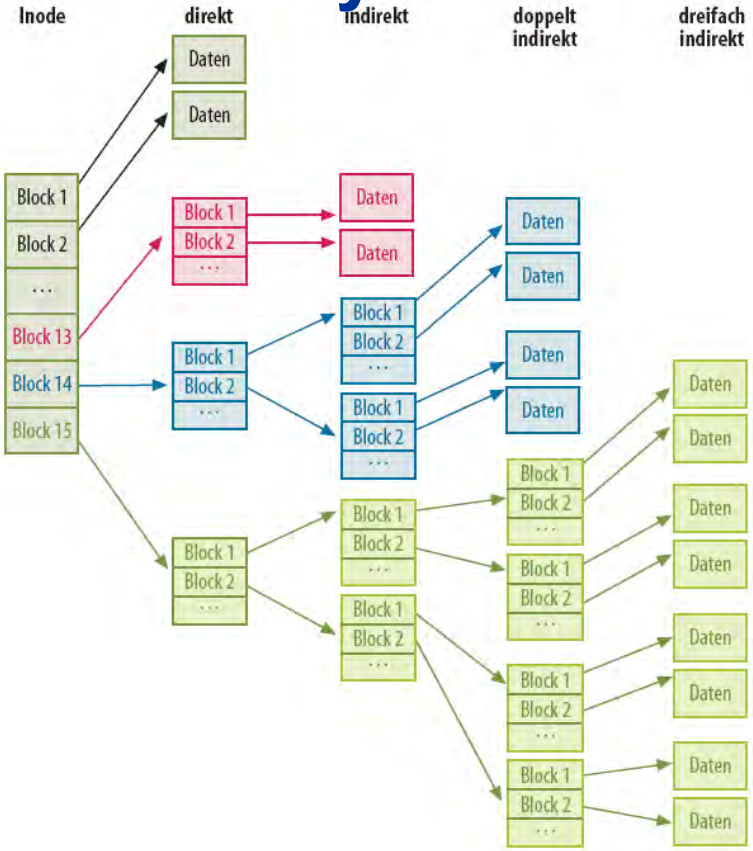
NTFS - Next Technology File System



Master File Table (12,5%)

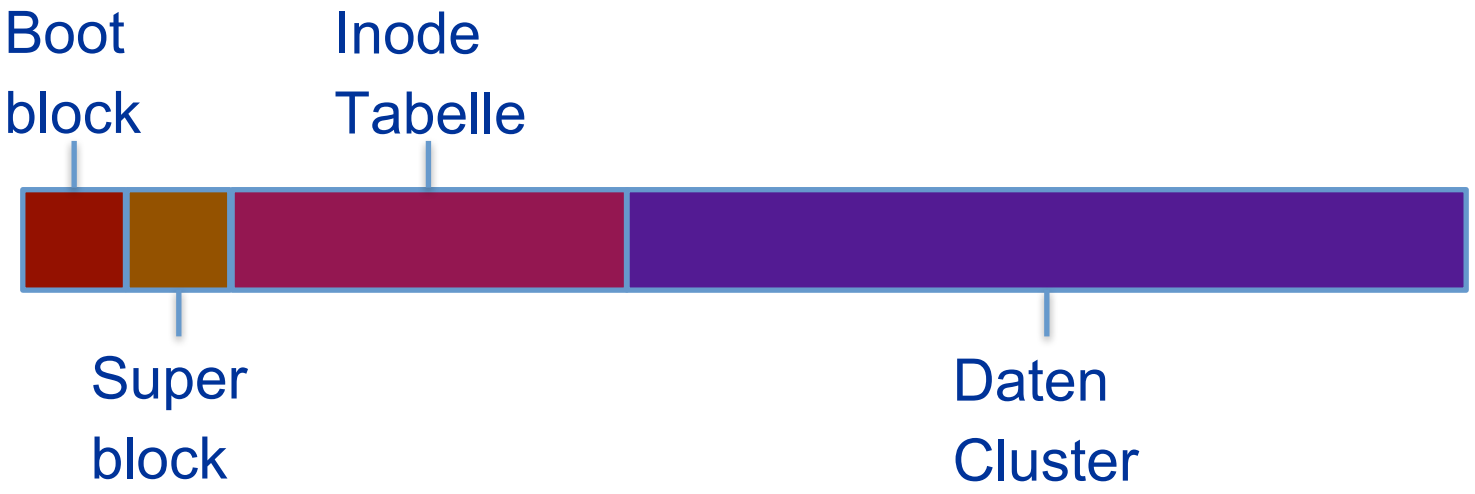
Datenbereich

Klassische Unix Dateisysteme



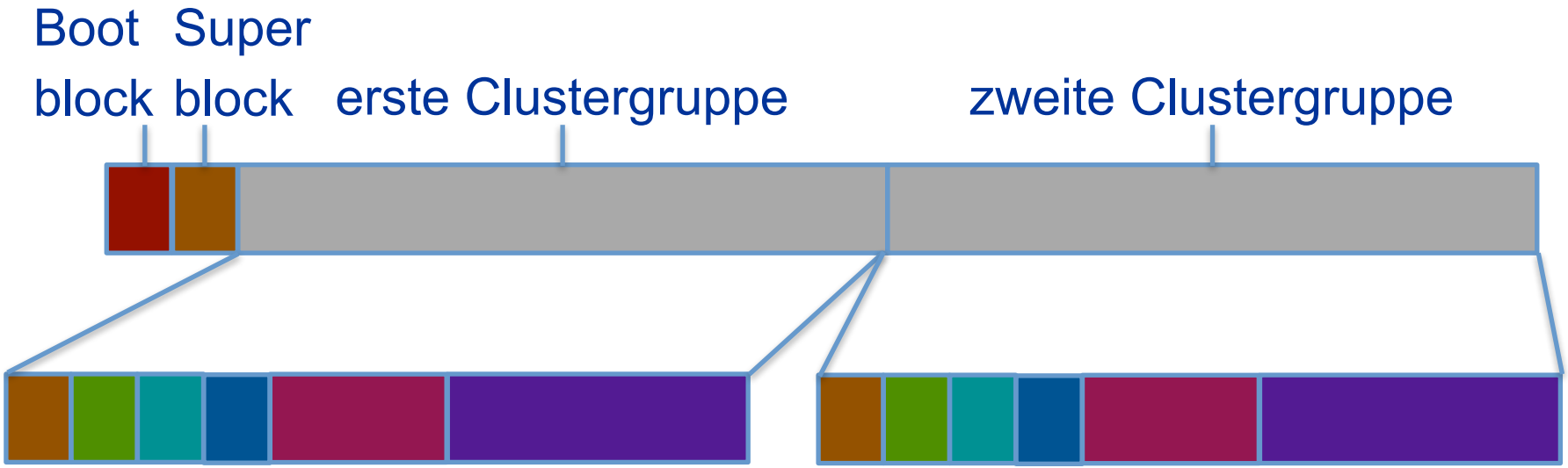
Quelle: heise.de

System V File System

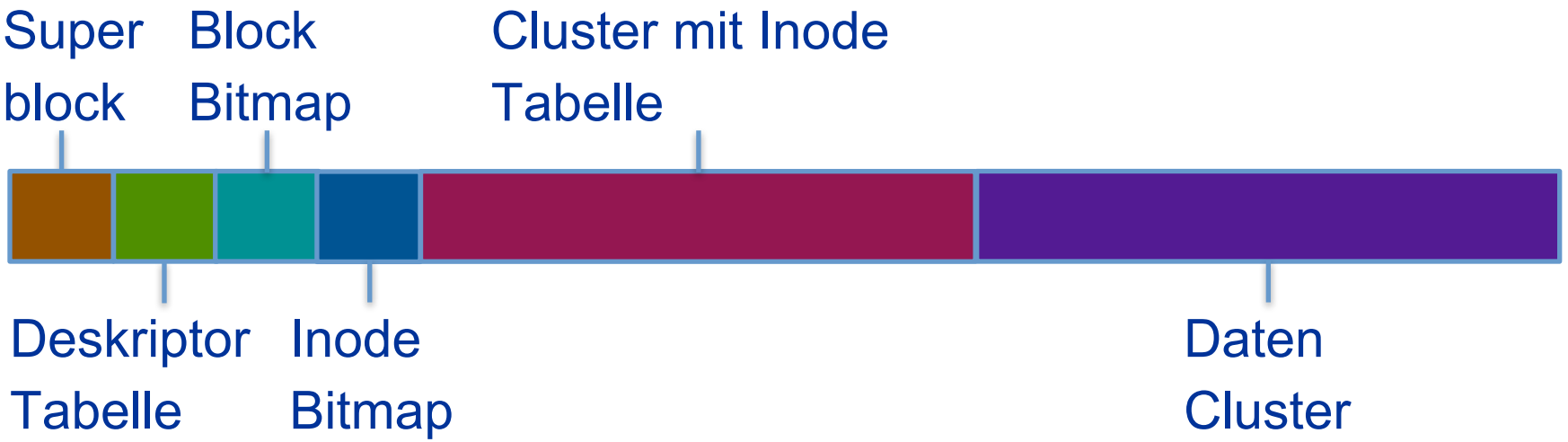


Linux ext2 / ext3 Dateisystem

Linux ext2 / ext3 Dateisystem



Linux ext2 / ext3 Dateisystem



DATEISYSTEME

Konzepte um Datenintegrität zu garantieren

Journaling

Metadaten - Journaling

Metadaten

Daten

Vollständiges Journaling

Metadaten

Daten

Ordered - Journaling

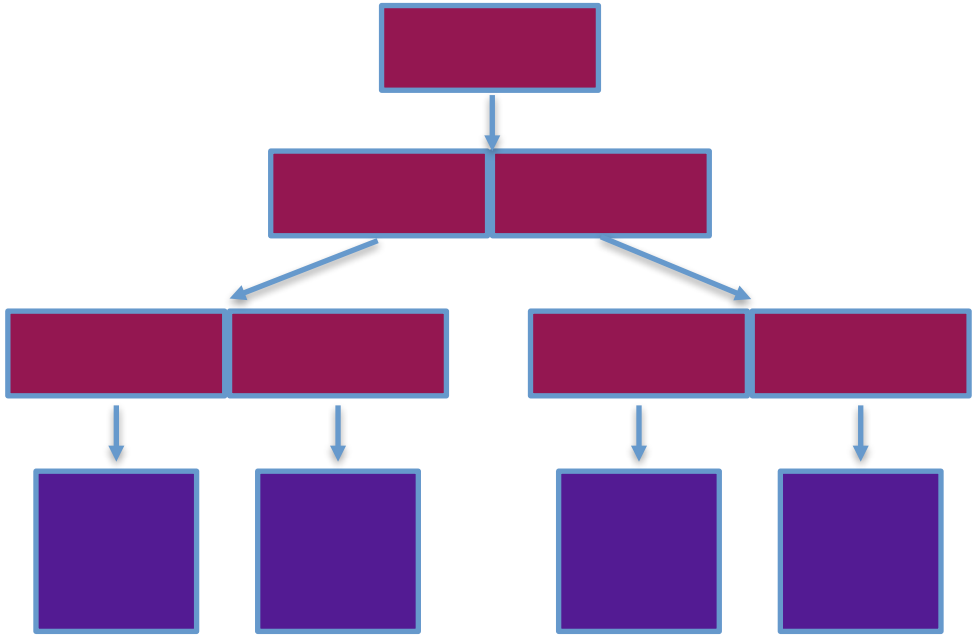
Metadaten

Daten

copy on write

Daten und Metadaten werden immer in freie Blöcke geschrieben:
es werden keine Daten überschrieben

ZFS - Beispiel für copy on write



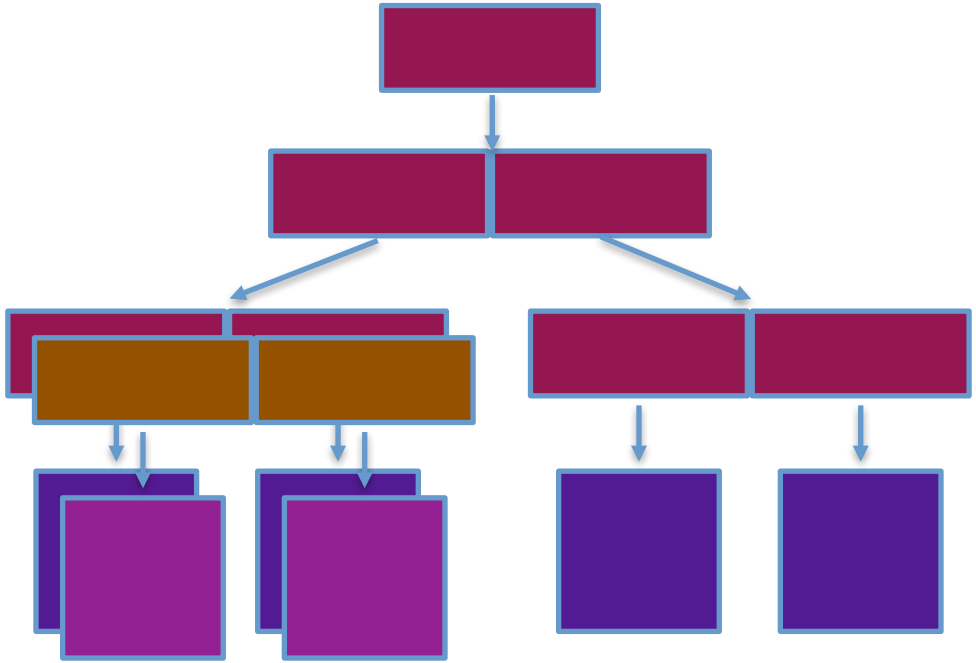
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



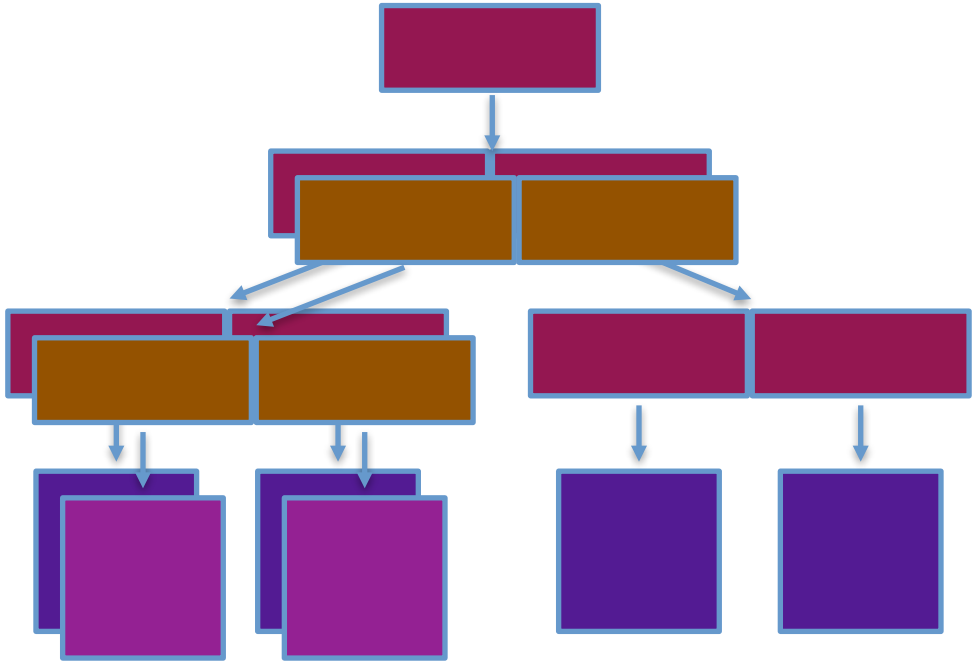
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



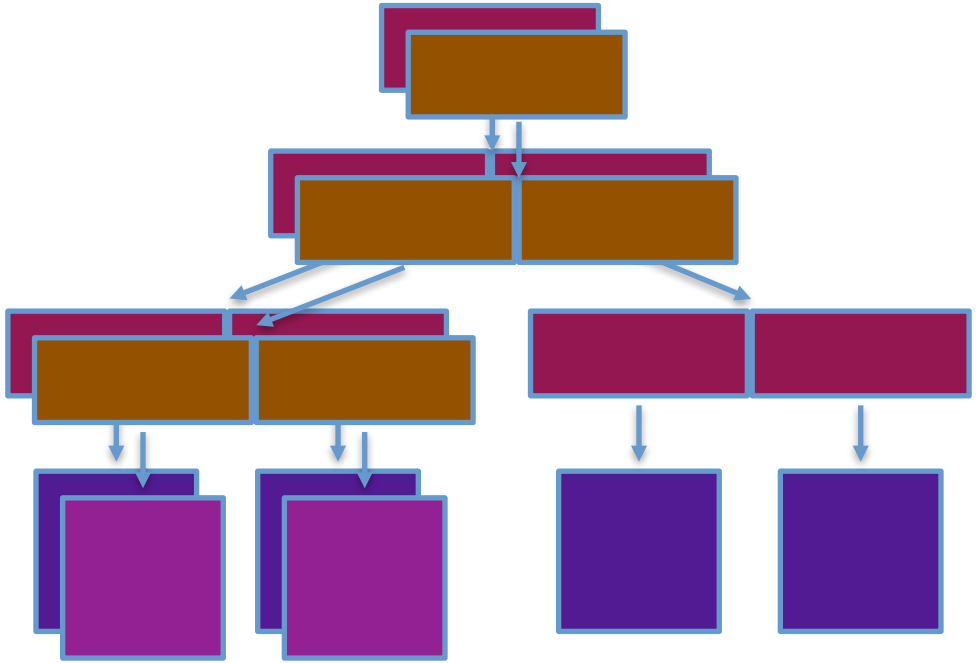
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



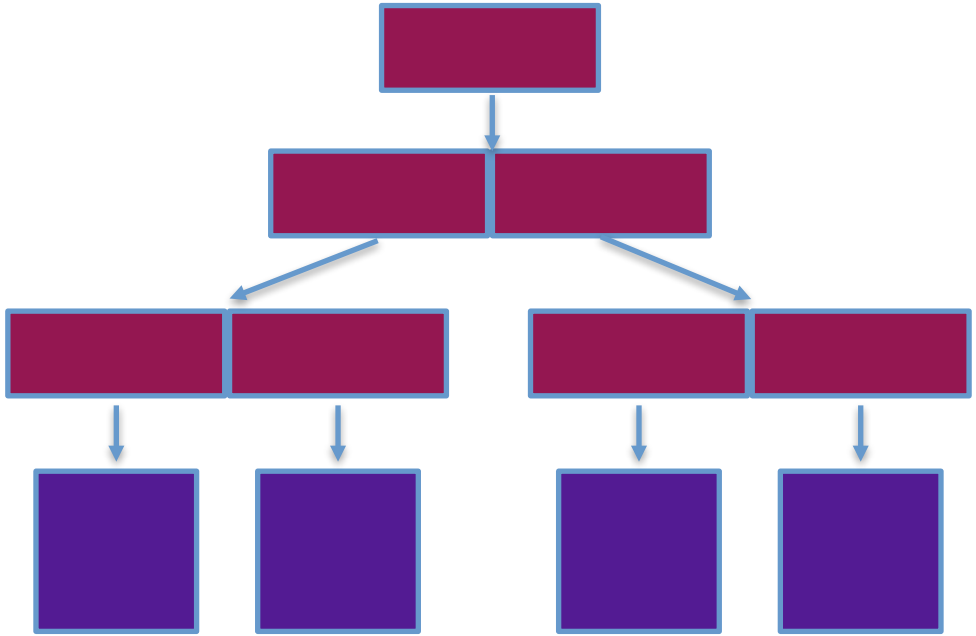
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

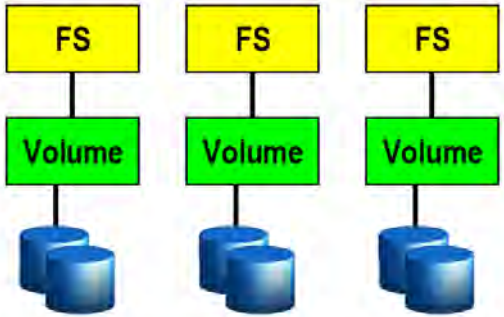
Daten

ZFS - mehr als nur ein Dateisystem

FS/Volume Modell vs. ZFS

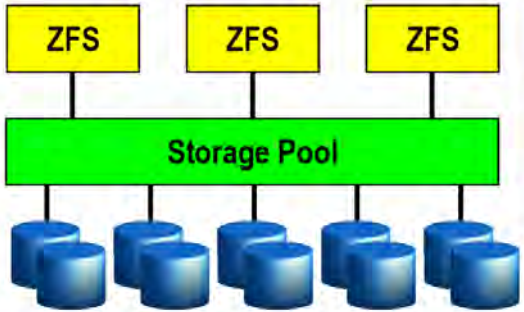
Traditionelle Volumes

- Abstraktion: virtuelle Disk (fest)
- Volume für jedes Filesystem
- Grow/shrink nur koordiniert
- Bandbreite / IOs aufgeteilt
- Fragmentierung des freien Platzes



ZFS Pooled Storage

- Abstraktion: Datei (variabel)
- Keine feste Platzeinteilung
- Grow/shrink via Schreiben/Löschen
- Volle Bandbreite / IOs verfügbar
- Freier Platz wird geshart

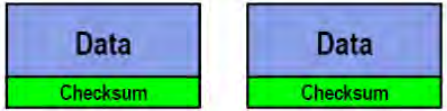


Quelle: Sun / RRZE

ZFS - Datenintegrität

Disk Block Prüfsummen

- Prüfsummen bei Datenblock
- Auf Disks meist kurz (Fehler unentdeckt)
- Einige Disk Fehler bleiben unentdeckt

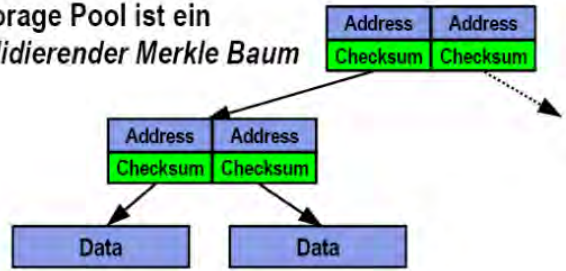


Nur Fehler auf Medium erkennbar

- ✔ Bit rot
- ✗ Phantom writes
- ✗ Misdirected reads and writes
- ✗ DMA parity errors
- ✗ Driver bugs
- ✗ Accidental overwrite

ZFS Daten Integrität

- Prüfsumme bei Adresse
- Gemeinsamer Fehler: unwahrscheinlich
- Storage Pool ist ein *validierender Merkle Baum*



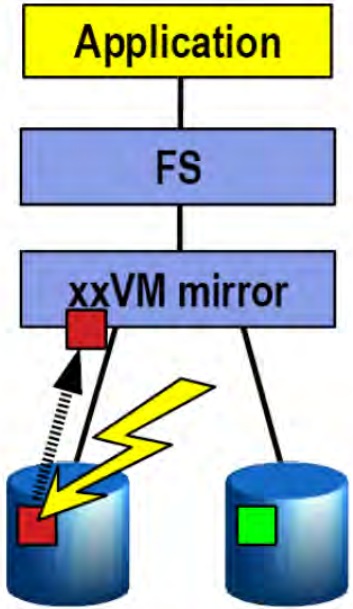
ZFS validiert alle Blöcke

- ✔ Bit rot
- ✔ Phantom writes
- ✔ Misdirected reads and writes
- ✔ DMA parity errors
- ✔ Driver bugs
- ✔ Accidental overwrite

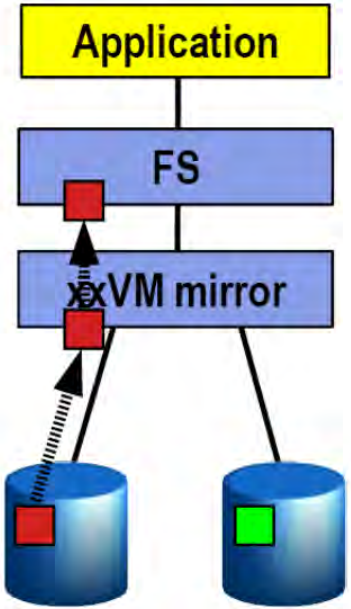
Quelle: Sun / RRZE

ZFS - Datenintegrität

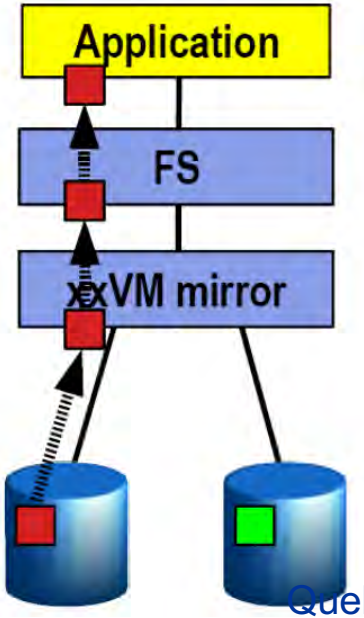
1. read liefert defekten Block



2. Falsche Metadaten:
Filesystem hat Probleme,
Absturz OS möglich



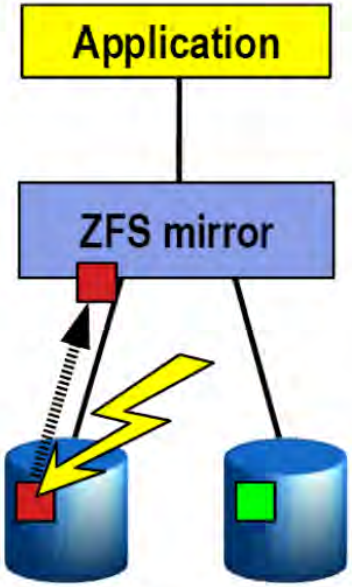
3. Falsche Daten:
Applikation bekommt Probleme
oder rechnet falsch
(ggf. unbemerkt!!!)



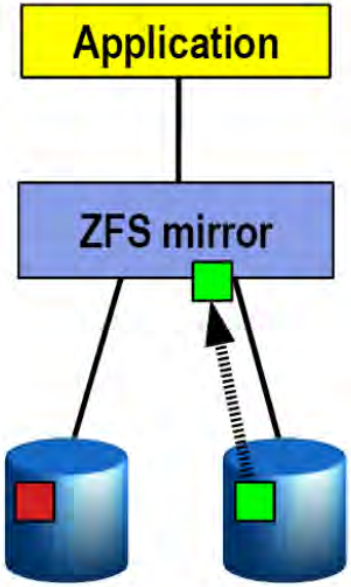
Quelle: Sun / RRZE

ZFS - Datenintegrität

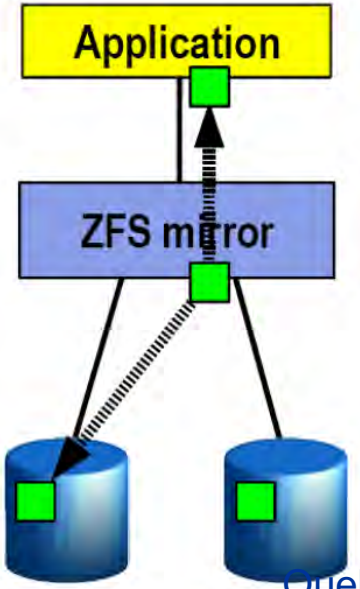
1. read liefert defekten Block



2. ZFS berechnet Prüfsumme; da diese falsch ist, wird der Spiegel gelesen (Metadaten sind also korrekt)



3. ZFS liefert korrekte Daten an die Applikation; UND korrigiert defekten Block!

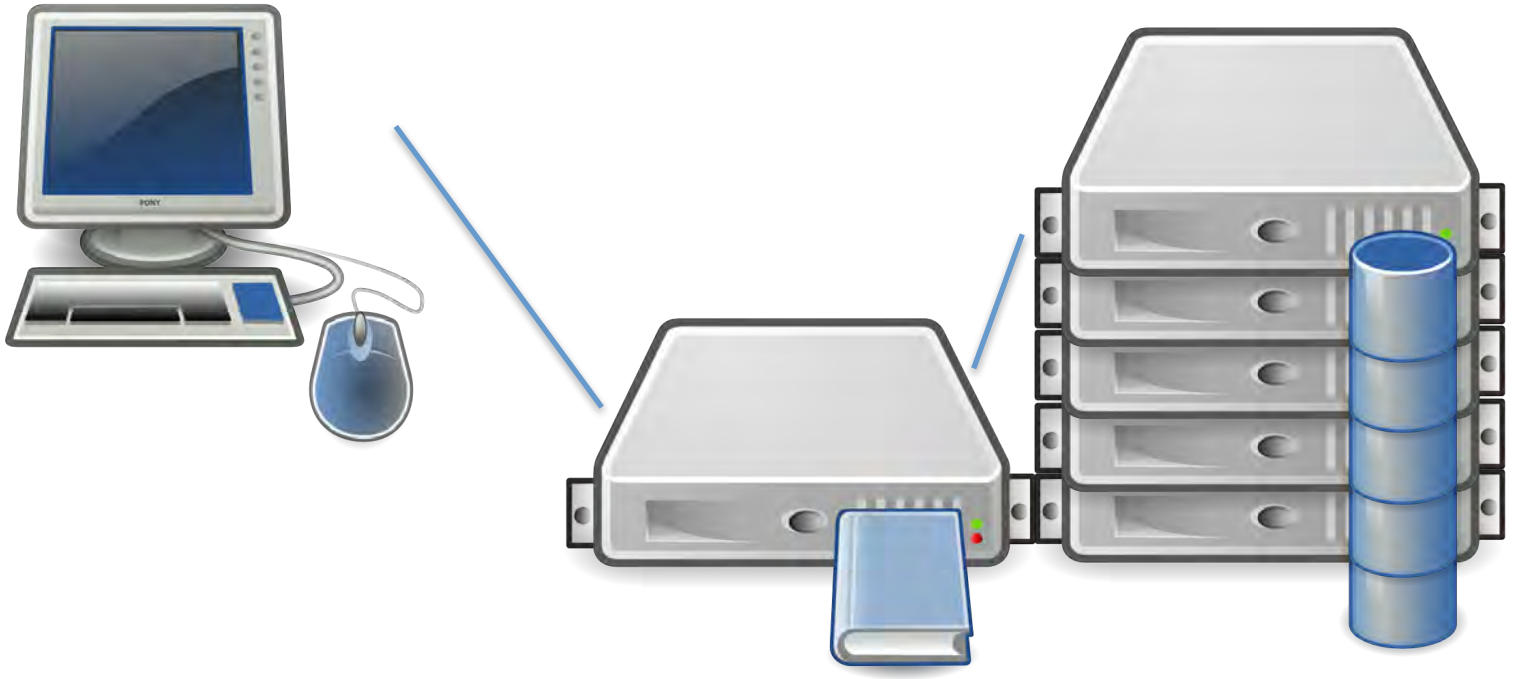


Quelle: Sun / RRZE

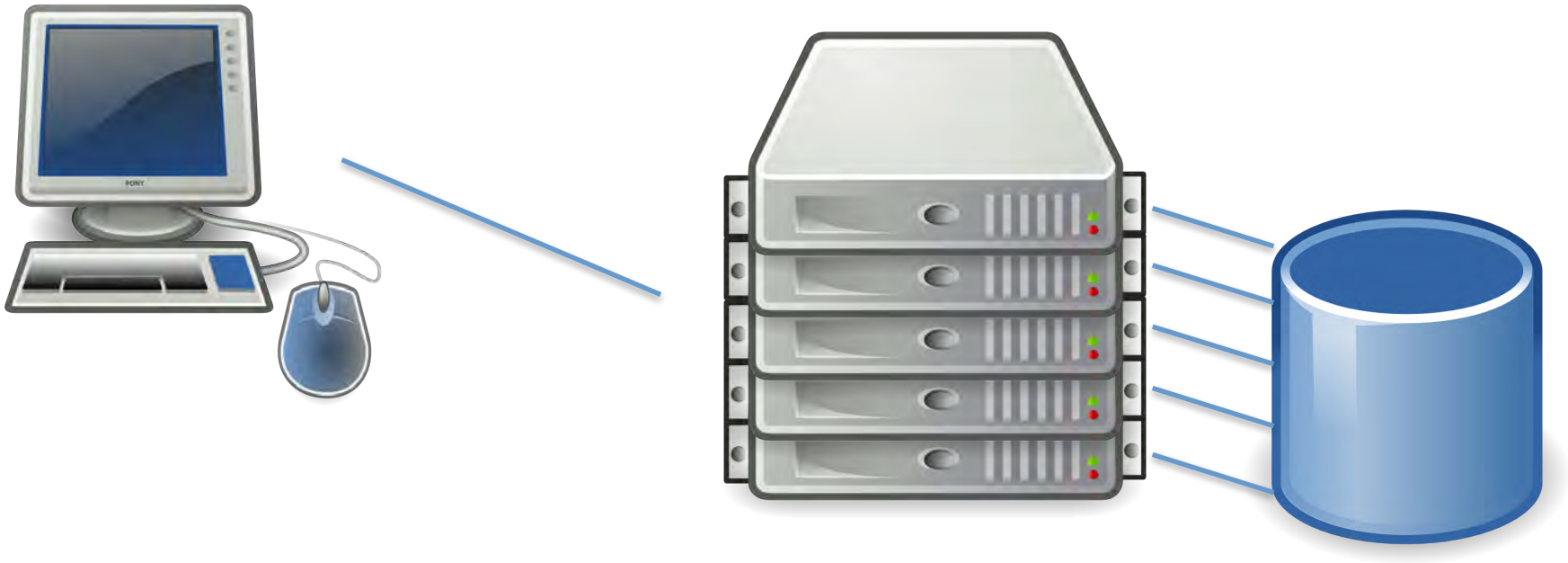
DATEISYSTEME IM NETZ

Verteilte- und Cluster- Dateisysteme

Verteilte Dateisysteme



Cluster- und andere Dateisysteme



NETWORK ATTACHED BLOCK

Blockgeräte über Storage Attached Netze verwenden

Blockprotokolle über SAN

Fibre Channel

FCoE

iSCSI

AoE

NAS - PROTOKOLLE

Netzwerk-File-System-Protokolle

Netzwerk-Filesystem-Protokolle

2 „Klassiker:

- Windows-Welt: CIFS/SMB
  - Common Internet Filesystem / System Message Block
  - Ursprung: IBM / Microsoft
- Unix-Welt: NFS
  - Network Filesystem
  - Ursprung: Sun Microsystems

Netzwerk-Filesystem-Protokolle – CIFS/SMB

- SMB
  - Version 1.0
- CIFS
  - Version 2.0 (2006) (>= Windows Vista / Server 2008)
    - Vereinfachung (Subcommands: > 100 => 19)
    - Neu: Symbolische Links, Größere Blockgröße, Unicode
  - Version 2.1 (>= Windows 7 / Server 2008 R2)
    - Performance
  - Version 3.0 (ehemals 2.2, >= Windows 8 / Server 2012)
    - SMB Direct (SMB over RDMA)
    - SMB Multichannel
    - End-to-End encryption

Netzwerk-Filesystem-Protokolle – NFS

- NFS – Version 2
  - Basierend auf RPC (Remote Procedure Call)
  - Portmapper (Port 111):
    - Vermittelt Dienste auf dynamischen Ports (Firewall!)
    - UDP (später erst: auch TCP)
  - 32 bit (max. 2 GB Filegröße)
- NFS – Version 3
  - UDP + TCP
  - 64 bit Support

Netzwerk-Filesystem-Protokolle – NFS

- NFS – Version 4
  - IETF
  - Single Standard Port 2049 => kein Portmapper mehr notwendig
  - NFSv4 ACLS (ähnlich Windows/CIFS ACLs)
  - RPCSEC_GSS (Kerberos)
- NFS – Version 4.1
  - pNFS

Netzwerk-Filesystem-Protokolle – NFS

- Sicherheit:
  - Beschränkung Host-basiert (AUTH_SYS / AUTH_UNIX)
  - ro / rw, (no_)root_squash, (in)secure (NAT VMs!)
  - Client-Server Mapping uid/gid-basiert (Sicherheit!)
  - Posix ACLs (nur RFC, kein Standard!)
- Ab Version 4.0:
  - Client-Server Mapping „String“-basiert (idmap!)
  - Starke Verschlüsselung / Authentifizierung
    - krb5: Authentication Only
    - krb5i: Integrity
    - krb5p: Privacy

AGENDA

- Hardware
- Storage
- Zugriffs-Protokolle
- Filesysteme

Storage / Filesysteme

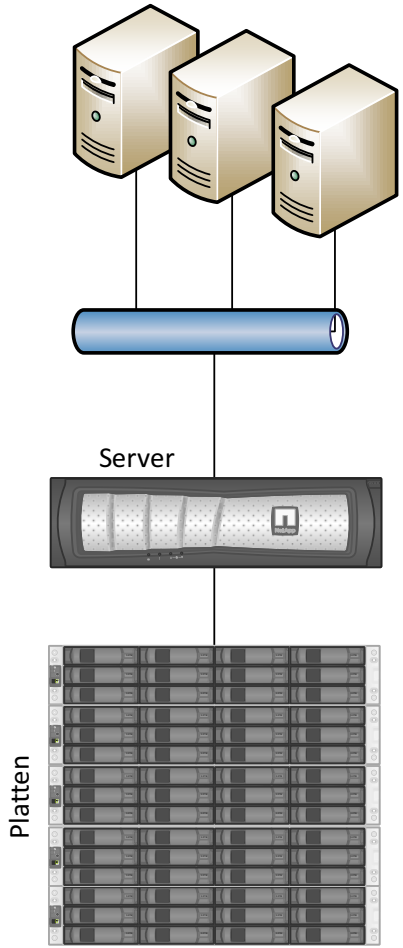
- Heute mal aus der anderen Richtung
- Von der Hardware …
- … bis zum Bit

TOP 1

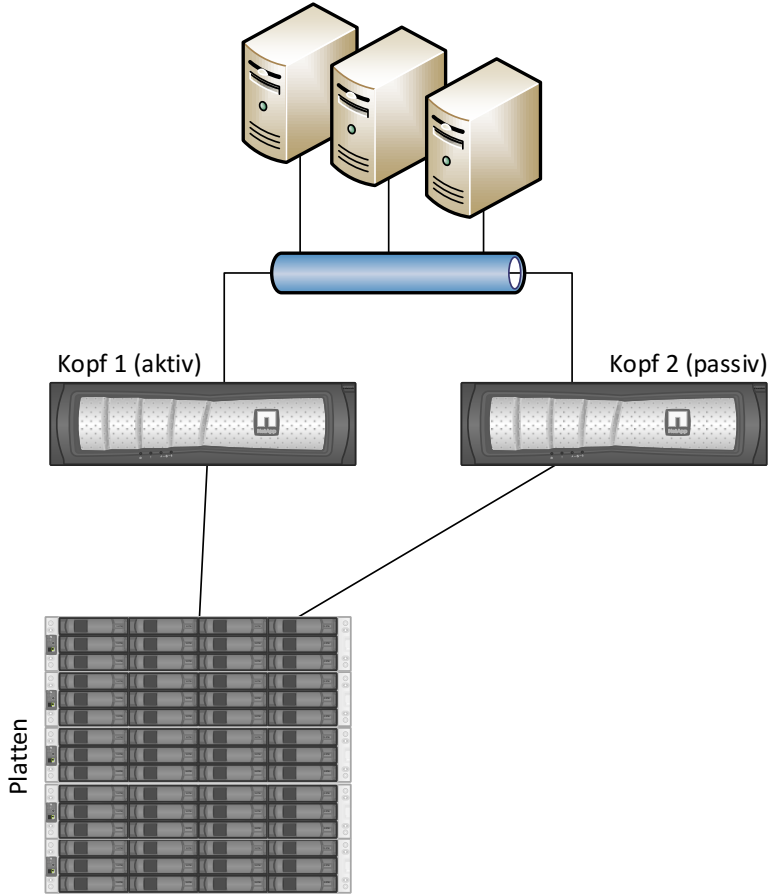
Hardware

- Typischer Aufbau / Storage Appliances

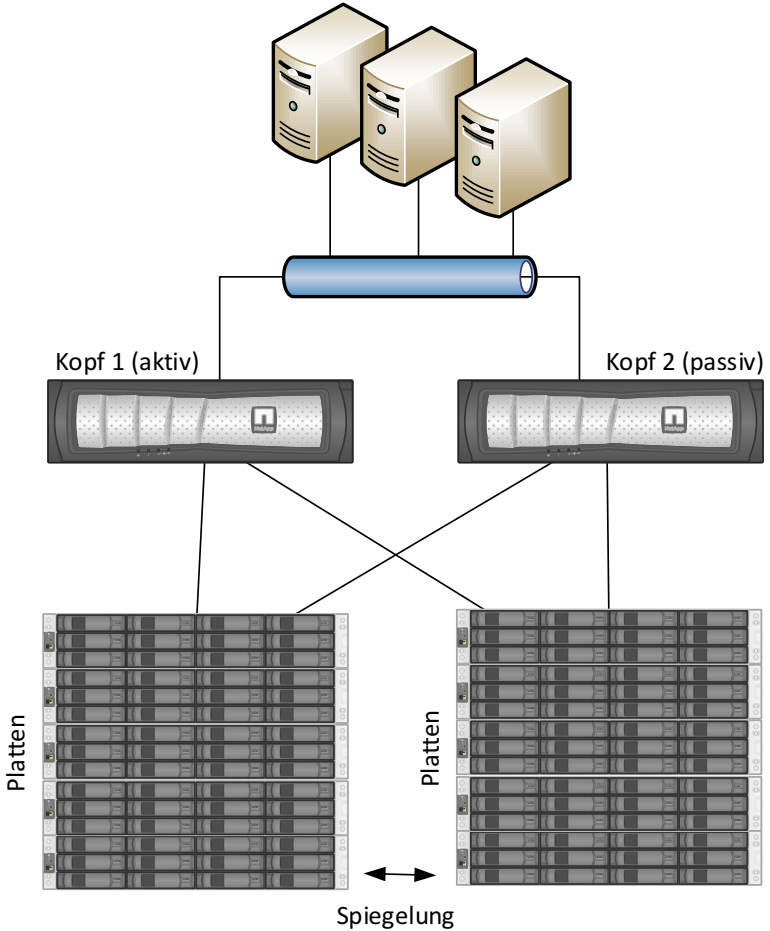
DAS – Direct Attached Storage



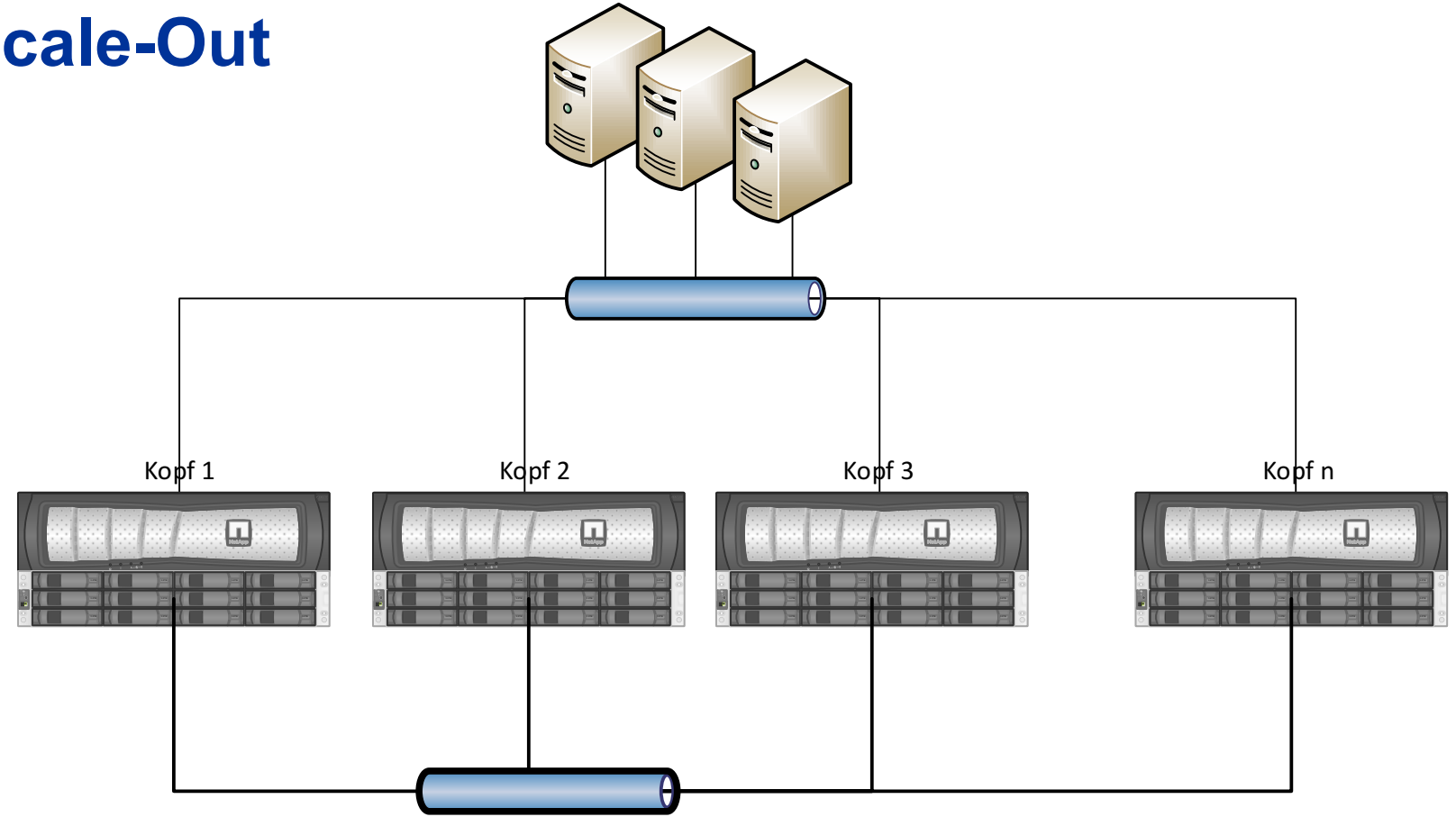
Klassisch



Klassisch (inkl. Spiegelung)



Scale-Out



Unterscheidung: SAN ↔ NAS

- SAN (Storage Area Network)
  - Block-Level Zugriff
  - Protokolle:
    - Eigene Verkabelung: Fibre Channel (FC), SAS
  - Basierend auf klassischem TCP/IP Netzwerk:
    - iSCSI, FCoE, AoE

- NAS (Network Attached Storage)
  - File-Level Zugriff
  - Mehr dazu später …

TOP 2

Storage-Grundlagen

- Verbund von Speichermedien:
  - RAID, Volume Manager

RAID (Redundant Array of Independent Disks)

- Zusammenfassung mehrerer Festplatten für
  - mehr Speicherplatz (am Stück)
  - mehr Ausfallsicherheit (nicht immer!)
  - RAID 0: Striping
  - RAID 1: Mirroring
  - RAID 4: 1 Parity (dedizierte Parity-Platte)
  - RAID 5: 1 Parity (verteilt Parity, Platz v. 1 Platte f. Parity)
  - RAID 6: 2 Parity (verteilt, Platz v. 2 Platten f. Parity)
  - Implementierungen in Hard- und Software
    - › Vor- und Nachteile!

Raid Levels

Raid-Level	Data Disks	Parity Disks	Spare Disks	Disk Errors	Speed W / R	Usable Space
0	N	0	0	0	++ / ++	N
1	N	N	0	1	0 / +	N/2
4	N	1	0	1	- / 0	N/(N+1)
5	N	1	0	1	0 / 0	N/(N+1)
5 + Spare	N	1	1	1	0 / 0	N/(N+2)
6	N	2	0	2	- / 0	N/(N+2)
10	N	N	0	1	++ / ++	N/2
50	N	2	0	1	0 / +	N/(N+2)
60	N	4	0	1	0 / +	N/(N+4)

Raid Levels / Bsp: 8x 1TB Disks

Raid-Level	Data Disks	Parity Disks	Spare Disks	Disk Errors	Usable Space
0	8	0	0	0	8 TB
1	4	4	0	1	4 TB
4	7	1	0	1	7 TB
5	7	1	0	1	7 TB
5 + Spare	6	1	1	1	6 TB
6	6	2	0	2	6 TB
10	4	4	0	1	4 TB
50	6	2	0	1	6 TB
60	4	4	0	1	4 TB

Fragen aus der Praxis: Raid5 + Hotspare oder Raid6

- Raid6 oder RAID5 mit Hotspare?
  - Verschnitt an Speicherplatz ist gleich
  - RAID5: Hotspare wird „geschont“
  - Aber im Fall eines Plattendefekts:
    - › Bei RAID5 besteht keinerlei Redundanz (effektiv: langsames Raid0)
    - › Nach Einspringen der HotSpare werden alle Daten von allen verbliebenen, intakten Platte gelesen um die Parity neu zu berechnen

Fragen aus der Praxis: Raid5 + Hotspare oder Raid6

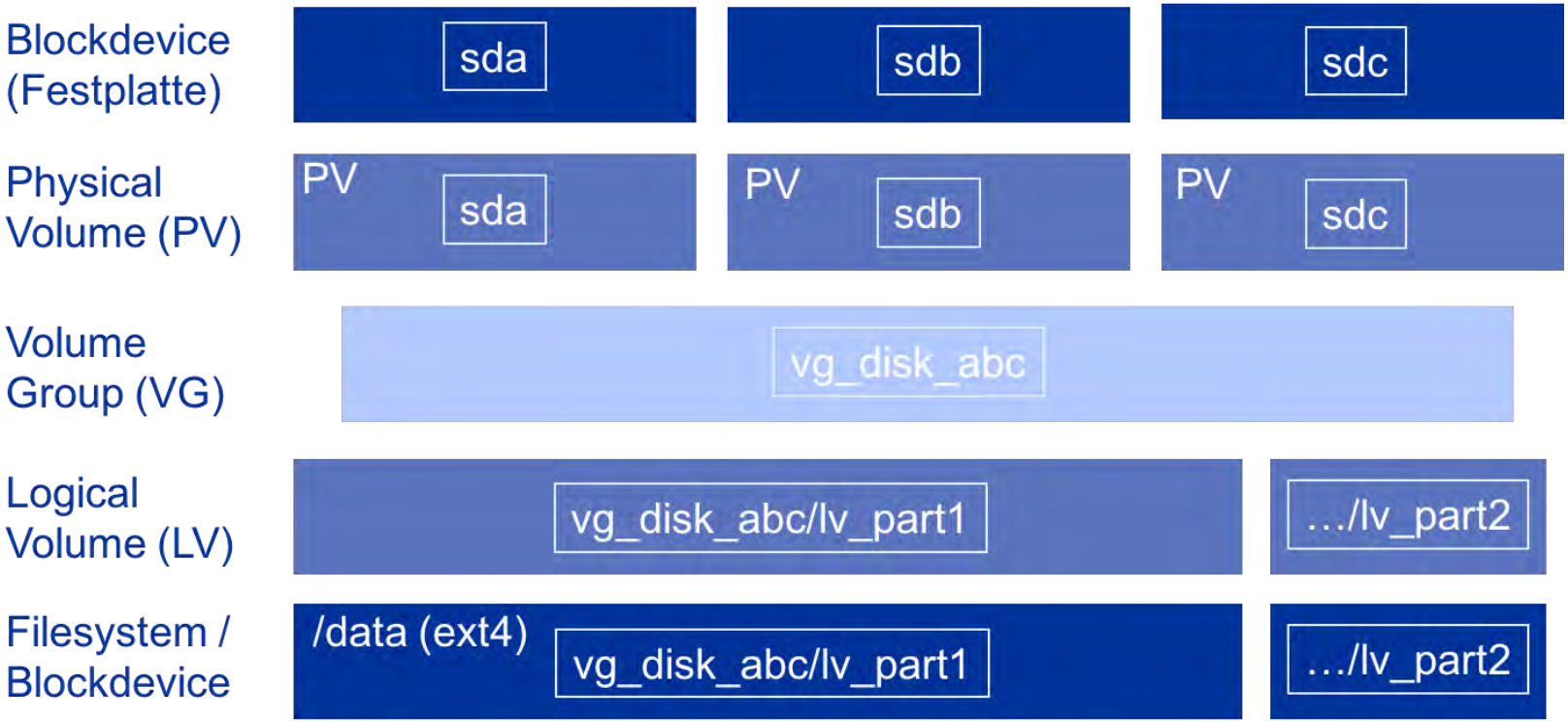
- Treten dabei Lesefehler auf, ist ein Rebuild ohne Datenverlust unmöglich
- Zeitfenster für Rebuild bei großen Festplatten enorm (2 TB bei 100 MB/s = 6 Stunden!)
- Fehlerwahrscheinlichkeit durch atypisches Lesen aller Disks ebenfalls!

- RAID6: Eine weitere Platte kann ausfallen / Lesefehler produzieren

LVM

- LVM (Logical Volume Manager)
  - Zusammenfassung mehrere Festplatten, meist flexibler als RAID
- Auch bei Windows:
  - Software-Raid über Systemsteuerung (auch System, Raid1)
  - Neuer: Storage Spaces (Resilient Storage)
    - Mirror (➔ Raid 1)
    - Parity (1/2 ➔ Raid 5/6)
    - Simple (➔ Raid 0)

LVM (Logical Volume Manager)Bsp: Linux LVM2



TOP 3

Storage-Grundlagen

- Grundlagen: Speichermedien

Storage: Grundlagen Speichermedien

- Block-basierte Speichermedien
  - Mechanisch: Festplatten
  - Speicher-basiert: SSDs, USB-Stick

- Begriffe
  - Latenz / Zugriffszeit
    - Spurwechselzeit
  - IO-Größe
  - IO-Rate
  - Bandbreite

Storage: Speichermedien – Mechanisch

- Bsp.: Festplatte 15k (15.000 Umdrehungen / Min)
  - Latenz: 60s / 15.000 = 4 ms
  - IOPs: 1 / 4 ms = 250 / s

  - Bandbreite:
    - Random / Worst Case:
      - 250/s x 4k Blöcke pro Sekunde -> 1000 KB/s (!)
    - Linear Read/Write
      - 150 MB/s

Storage: Speichermedien – Speicherbasiert

- Keine mechanisch bedingten Latenzen
- Potentiell höhere Bandbreiten
  - Random I/O weniger „schmerzhaft“
- Aber:
  - Limitierte Anzahl Schreib-Zyklen pro Zelle
  - Wear-Leveling notwendig
    - › Gleichmäßige Verteilung von Schreiboperationen für max. Lebensdauer
  - Verantwortlich: eigener Controller
  - Problem: nicht länger genutzte Sektoren erkennen
    - › Neues Kommando: TRIM

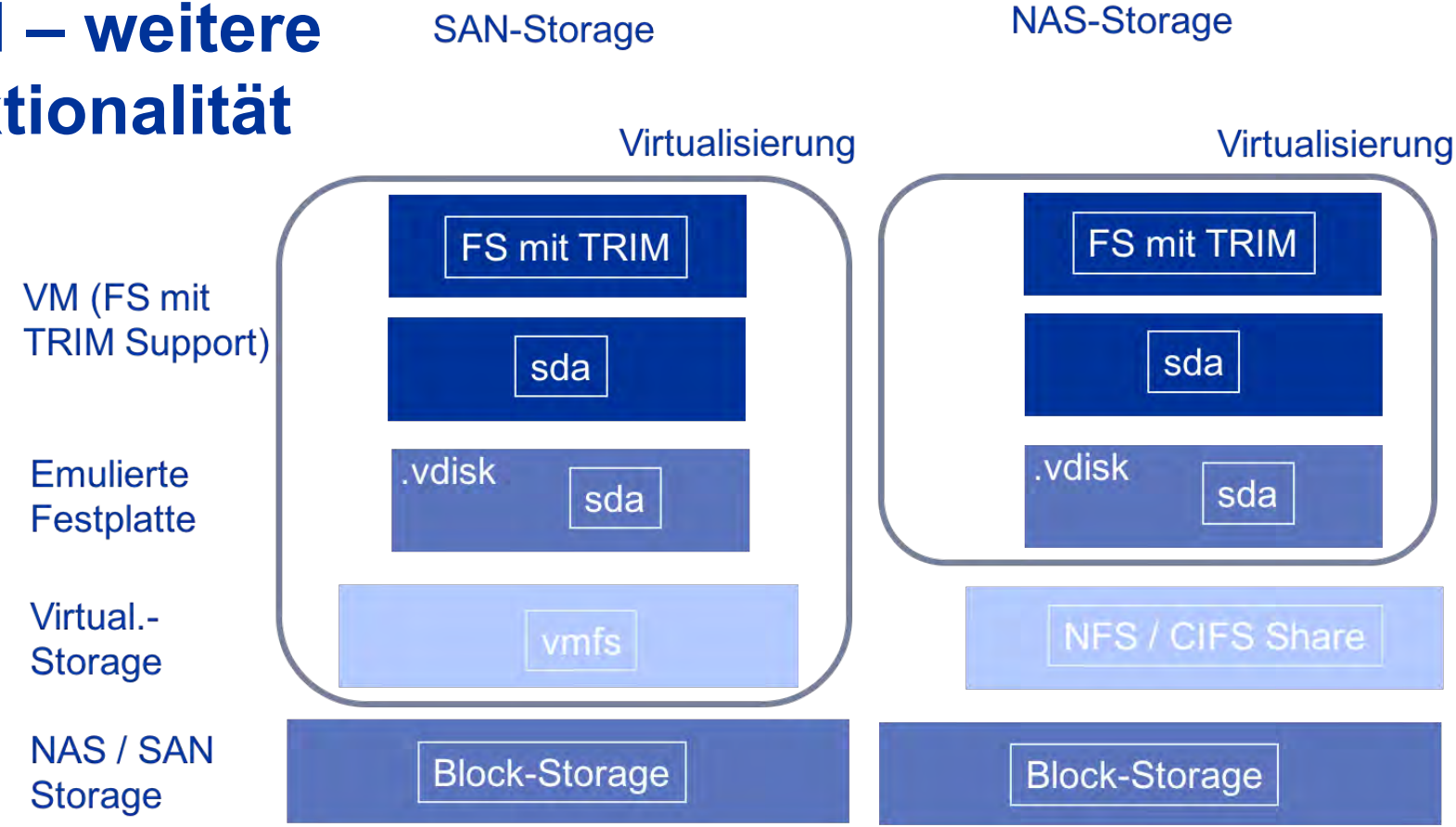
Storage: Vergleich Speichermedien Mechanisch / SSD / NVDIMM

Typische Leistungsdaten:

Type	Latenz+Seek (Theorie)	IOPs (Theorie)*	Bandbreite (Theorie)**	R/W IOPs (Praxis)
3,5“ 15k SAS	4 ms	250	1.000 KB/s	180 / 165
2,5“ 15k SAS	4 ms	250	1.000 KB/s	200 / 190
2,5“ 10k SAS	6 ms	166	664 KB/s	150 / 140
2,5“ 7.2k SATA	8 ms	120	480 KB/s	80 / 74
2,5“ 5.4k SATA	11 ms	90	360 KB/s	52 / 50
2,5“ eMLC SSD SAS	0.5 ms	-	-	~ 50.000
SSD NVMe (TOP!)	0.01 ms	-	-	~ 300.000
NVDIMM(-N) ***	0.00001 ms	-		> 1.000.000

* 1s / (Latenz+Seek) (max. random) ** bei 4k Blöcken (max. random) *** „-N“: DRAM-based, „-F“ FLASH-based „-P“ Mixed – Coming with DDR5?

TRIM – weitere Funktionalität



TOP 4

- Filesysteme

Eigenschaften

- Quota, Rechte (xattr, acl),
- Journal (Filesystem-Check!)
- Kompression, Verschlüsselung, Checksummen
- Snapshots, Klonen, Deduplikation
- int. RAID / LVM
- Beschränkungen:
  - Dateigröße, Filesystemgröße, Länge Dateiname
  - Vergrößern/Verkleinern

Journal

- Teil fast aller aktueller Linux-Dateisysteme
- Änderungen werden erst in Journal geschrieben
- Später (im Hintergrund) „sauber“ ins Dateisystem integriert
- Warum?
  - Ohne: Prüfung des kompletten Dateisystems bei ungeplantem Reboot / Ausfall (kann Stunden dauern!)
  - Journal reduziert diese Zeit auf wenige Sekunden

Kompression / Verschlüsselung / Checksummen

- Kompression
  - Daten werden beim Speichern (oder in einem nachgelagerten Prozess) komprimiert um Platz zu sparen
- Verschlüsselung
  - Daten werden beim Speichern verschlüsselt
  - Bei Diebstahl der Platten kein Zugriff auf Daten möglich
  - Aber: Eingabe eines Passworts vor dem Zugriff notwendig
- Checksummen
  - Datenblöcke werden beim Schreiben mit Checksumme versehen
  - Fehler beim Lesen können dadurch erkannt werden (8 TB Festplatten!)

Snapshot / Klonen / Dedup / Scrubbing

- Snapshot
  - (Schnelles) Einfrieren des Zustands eines Dateisystems
- Klonen
  - Schnelles „Kopieren“ von Dateien
  - COW (Copy-on-Write)
- Deduplikation
  - Zusammenfassen von Datenblöcken mit gleichem Inhalt
  - Speicherhungrig!
  - Durchbruch erst mit SSDs (weniger Penalty!)
- Scrubbing
  - Regelmäßiges Prüfen der Daten (ggf. inkl. Schreibtest)

Beschränkungen

- Dateisystemgröße
  - z.B. ReiserFS: 16 TB
- Dateigröße
  - z.B. FAT32: 2 GB (USB Sticks!)
- Länge Dateiname/-pfad
  - z.B. DOS: 8.3 Zeichen „autoexec.bat“
- Vergrößern / Verkleinern des Dateisystems
  - Online / Offline?
- Einsatzzweck?

TOP 5

Network Attached Storage (NAS)

- Zugriffsprotokolle

Netzwerk-Filesystem-/NAS-Protokolle

- 2 „Klassiker:

- **Windows**-Welt: **CIFS/SMB**
  - Common Internet Filesystem / System Message Block
  - Ursprung: IBM / Microsoft

- **Unix**-Welt: **NFS**
  - Network Filesystem
  - Ursprung: Sun Microsystems

Netzwerk-Filesystem-Protokolle – CIFS/SMB

- **SMB** (Server Message Block)
  - Version 1.0

- **CIFS** (Common Internet FileSystem)
  - **Version 2.0** (2006) (>= Windows Vista / Server 2008)
    - › Vereinfachung (Subcommands: > 100 => 19)
    - › Neu: Symbolische Links, Größere Blockgröße, Unicode
  - **Version 2.1** (>= Windows 7 / Server 2008 R2)
    - › Performance

Netzwerk-Filesystem-Protokolle – CIFS/SMB

- **Version 3.0** (ehemals 2.2, >= Windows 8 / Server 2012)
  - SMB Direct (SMB over RDMA)
  - SMB Multichannel
  - End-to-End encryption

- **Version 3.0.2** (auch 3.02, >= Windows 8.1 / Server 2012R2)
  - SMB1 abschaltbar (Sicherheit!)

- **Version 3.1.1** (>= Windows 10 / Server 2016)
  - Secure negotiation Pflicht für SMB >= 2.x

Netzwerk-Filesystem-Protokolle – NFS

- **Version 2** (RFC 1094, 03/1989)
  - Basierend auf RPC (Remote Procedure Call)
  - Portmapper (Port 111):
    - Vermittelt Dienste auf dynamischen Ports (Firewall!)
    - UDP (später erst: auch TCP)
  - 32 bit (max. 2 GB Filegröße)
- **Version 3** (RFC 1813, 06/1995)
  - UDP + TCP
  - 64 bit Support
  - Asynchrones Schreiben

Netzwerk-Filesystem-Protokolle – NFS

- **Version 4** (RFC 3010, 12/2000, Rev.: RFC 3530/7530)
  - Single Standard Port 2049 (kein Portmapper!)
  - NFSv4 ACLS (ähnlich Windows/CIFS ACLs)
  - RPCSEC_GSS (Kerberos)
- **Version 4.1** (RFC 5661, 01/2010)
  - pNFS
- **Version 4.2** (RFC 7862, 11/2016)
  - Sparse File Support
  - Server Side Copy
  - Space Reservation

Netzwerk-Filesystem-Protokolle – Warum NFS 4.x nutzen?

- Bis inkl. Version 3
  - Beschränkung Host-basiert (AUTH_SYS / AUTH_UNIX)
  - ro / rw, (no_)root_squash, (in)secure (NAT VMs!)
  - Client-Server Mapping uid/gid-basiert (Sicherheit!)
  - Posix ACLs (nur RFC, kein Standard!)
- Ab Version 4.0:
  - Client-Server Mapping „String“-basiert (idmap!)
  - Starke Verschlüsselung / Authentifizierung
    - krb5 (Authentication Only), krb5i (Integrity), krb5p (Privacy)

Vielen Dank für Ihre Aufmerksamkeit!

Regionales RechenZentrum Erlangen [RRZE]
Martensstraße 1, 91058 Erlangen
http://www.rrze.fau.de