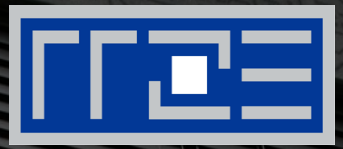
REGIONALES RECHENZENTRUM ERLANGEN [RRZE]



Netzwerkspeicher und Dateisysteme

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

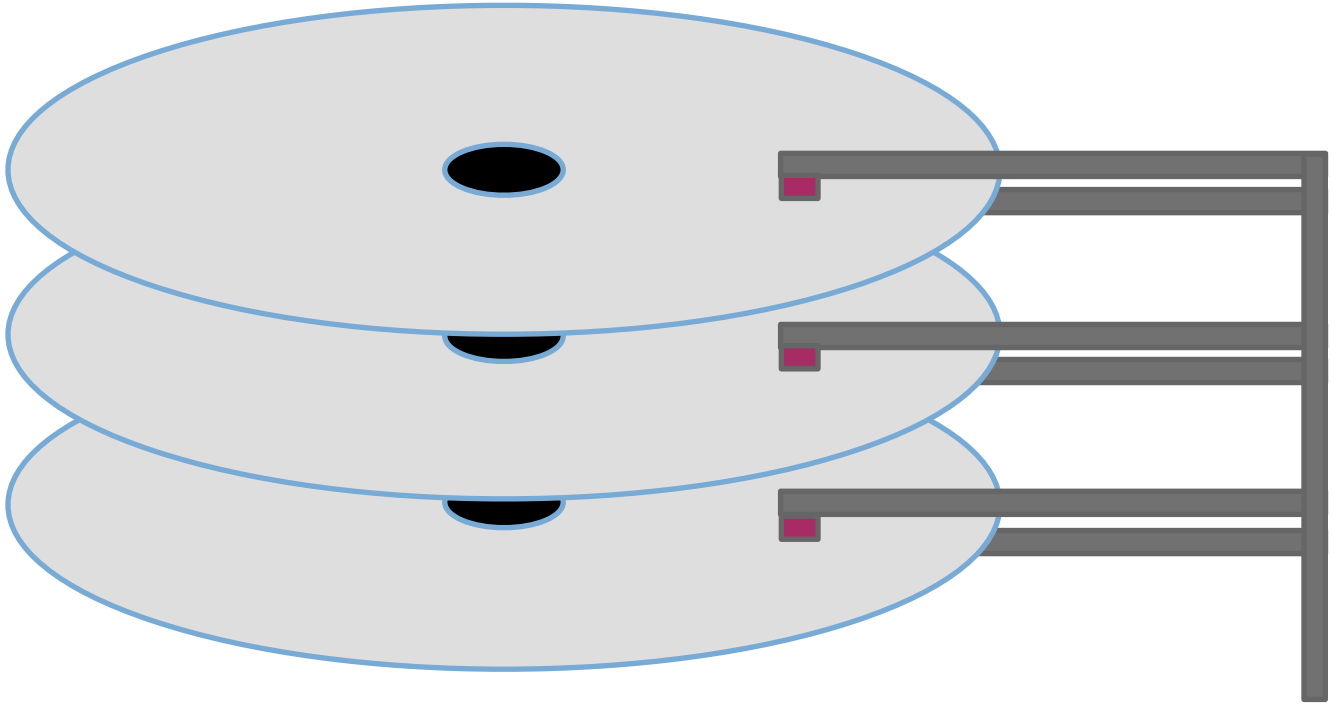
Marcel Ritter, Gregor Longariva, 03.06.2015

Agenda

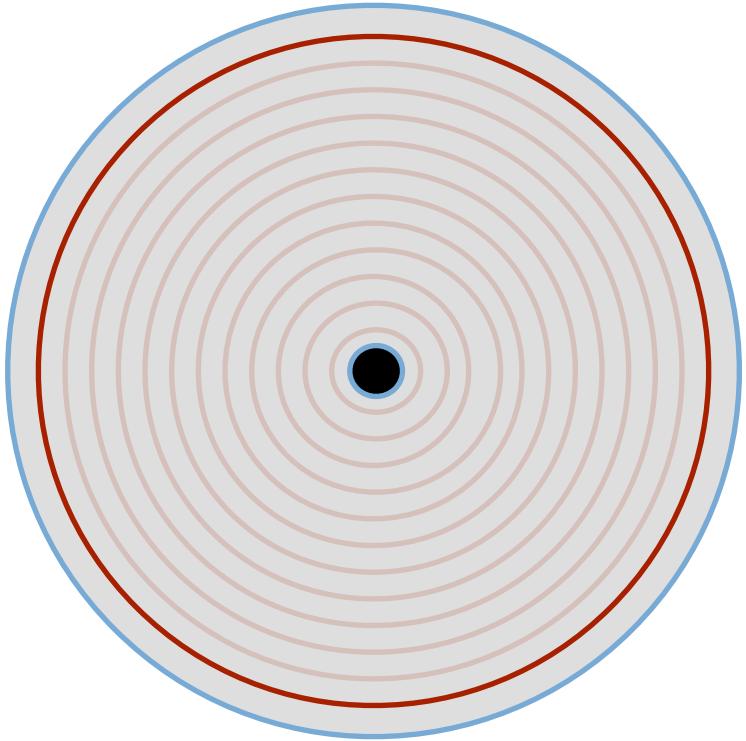
FESTPLATTEN

Prinzipieller Aufbau

Aufbau einer Festplatte

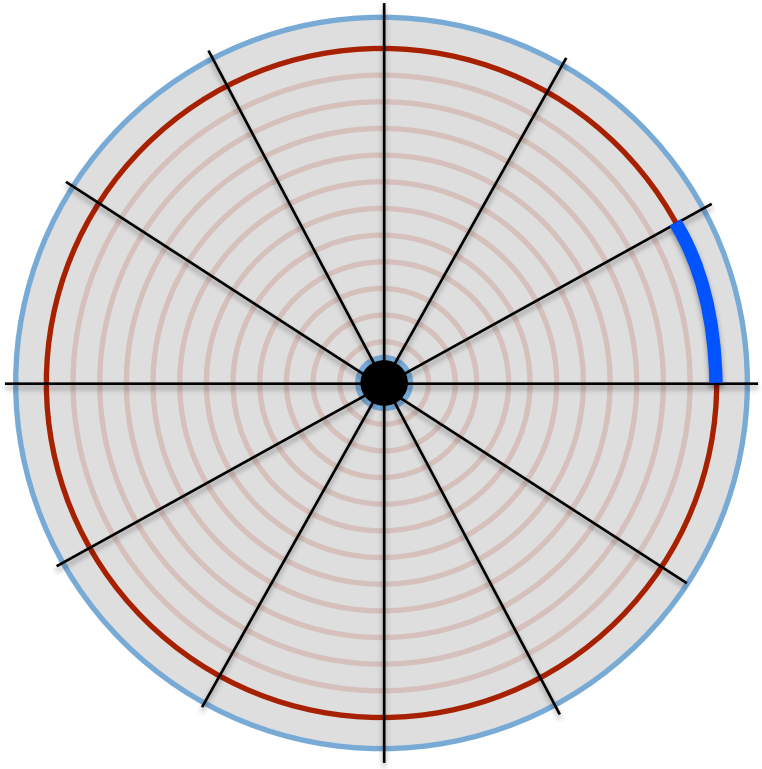


Aufbau einer Festplatte



Spur

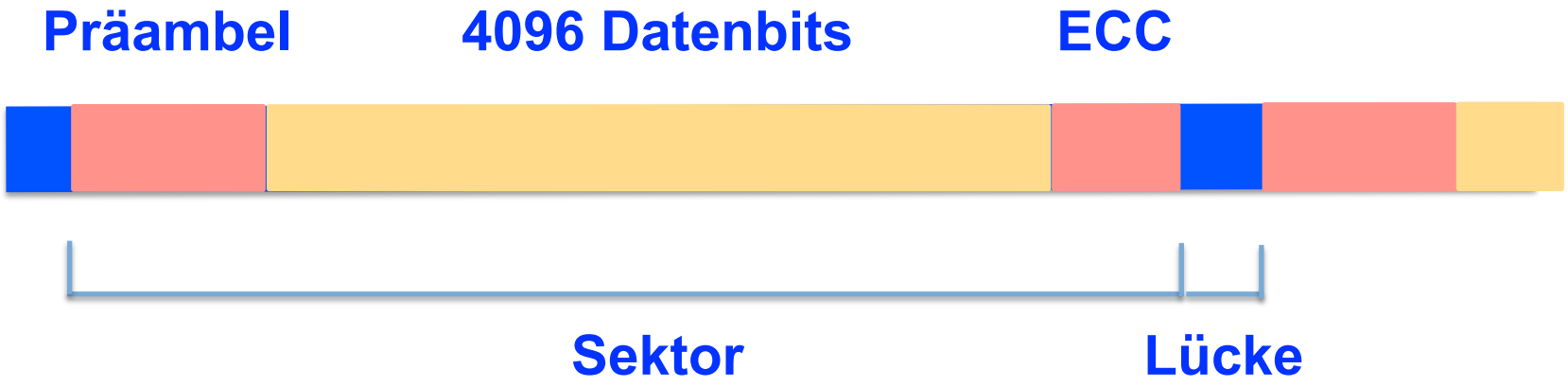
Aufbau einer Festplatte



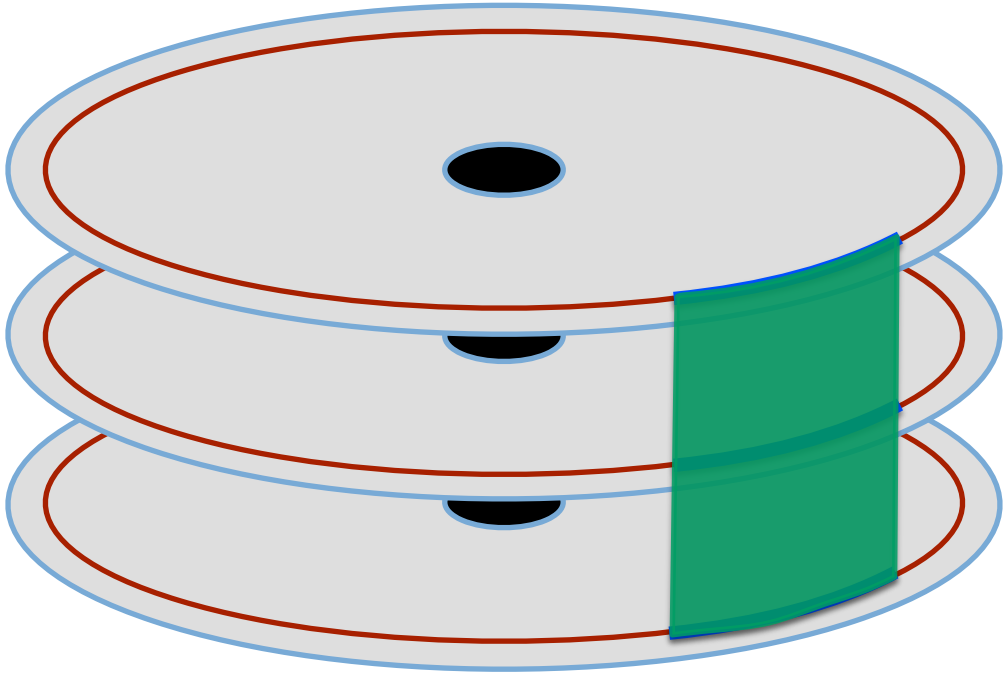
Spur

Sektor

Aufbau einer Festplatte



Aufbau einer Festplatte

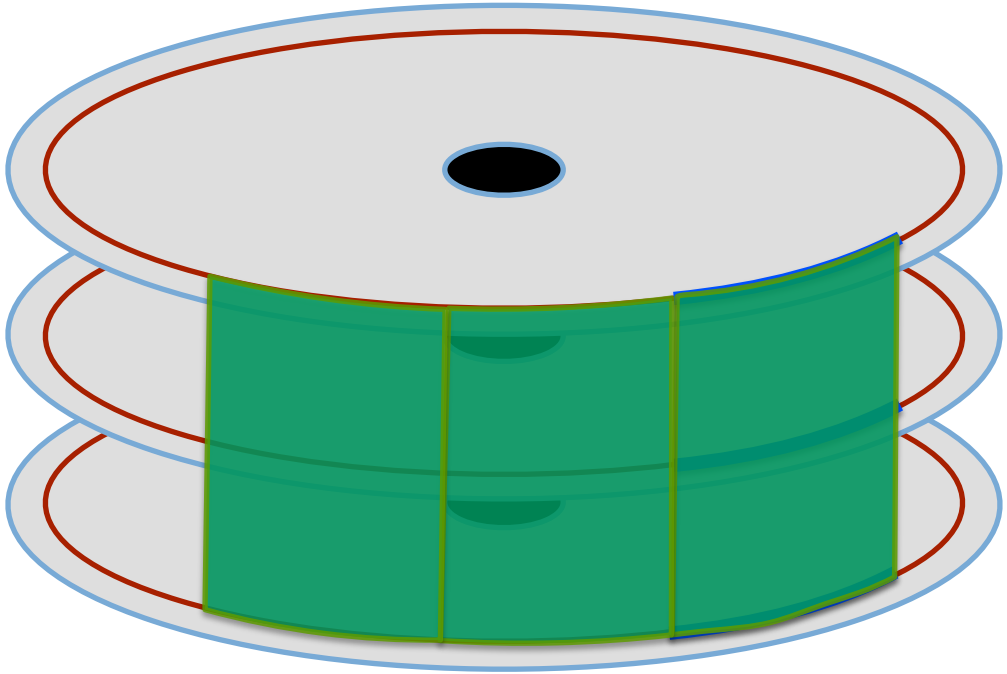


Spur

Sektor

Zylinder

Aufbau einer Festplatte

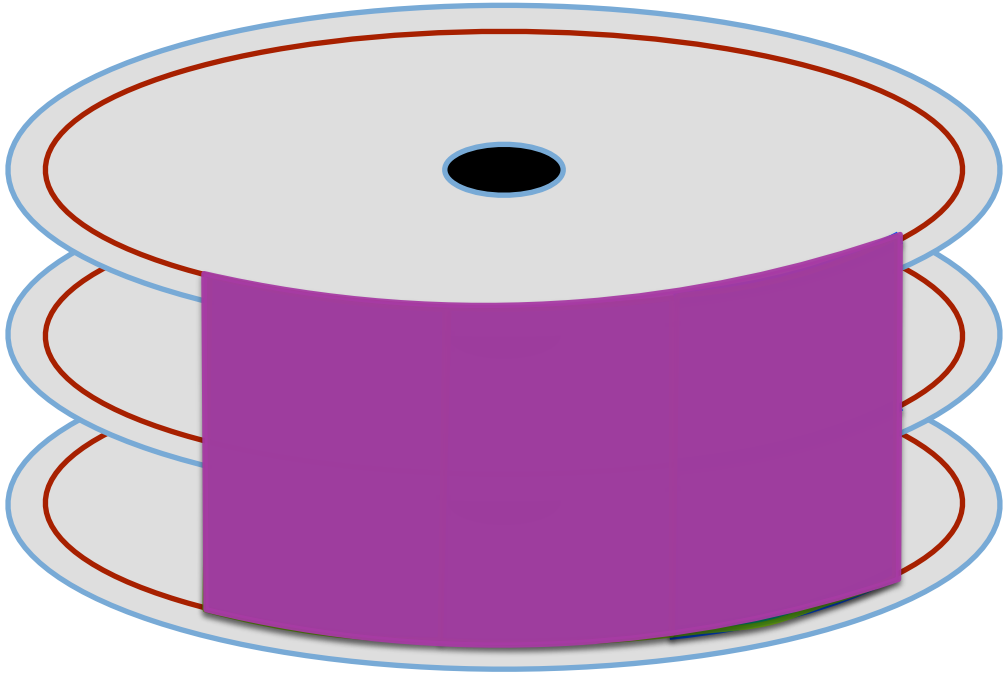


Spur

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

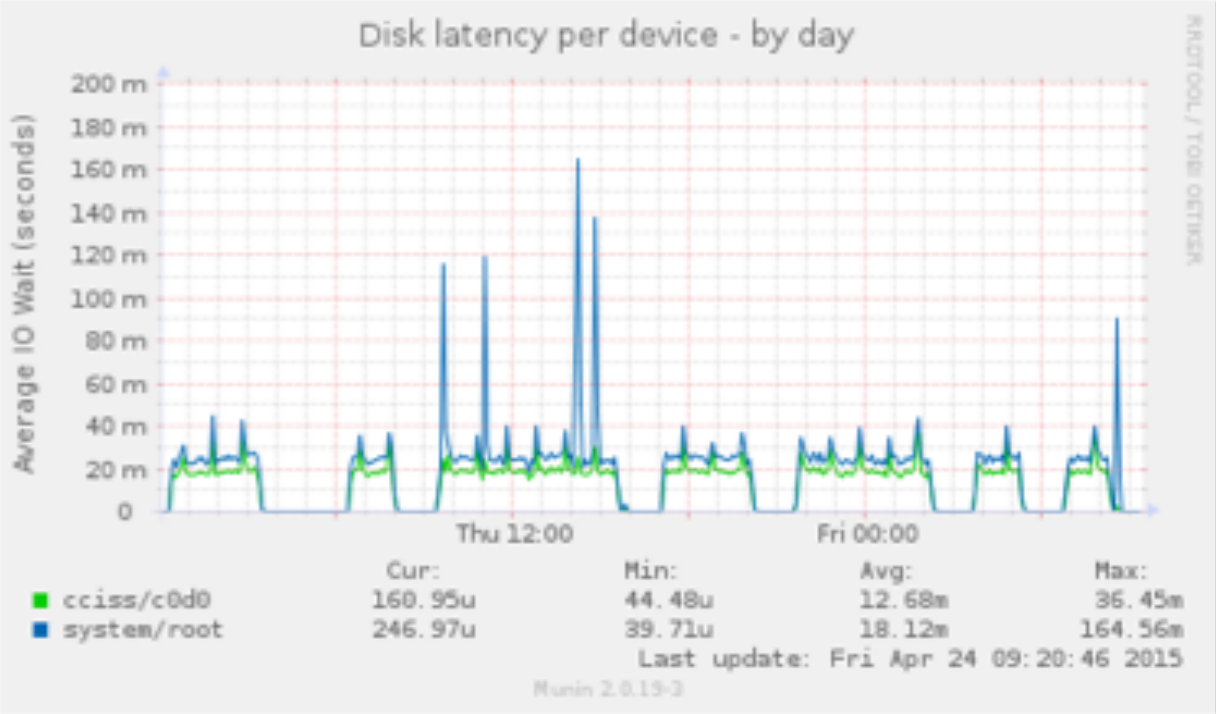
also ca. 180 MByte pro Sekunde

(aber immer noch ohne Plattencaches)

Theoretische Werte vs. Herstellerangaben

Type	Latenz (Theorie)	IOPs (Theorie)	R/W IOPs
3,5“ 15k SAS	4	250	180 / 165
2,5“ 15k SAS	4	250	200 / 190
2,5“ 10k SAS	6	166	150 / 140
2,5“ 7.2k SATA	8.3	120	80 / 74
2,5“ 5.4k SATA	11	90	52 / 50
2,5“ eMLC SSD	0.5	2000	100000 / 40000

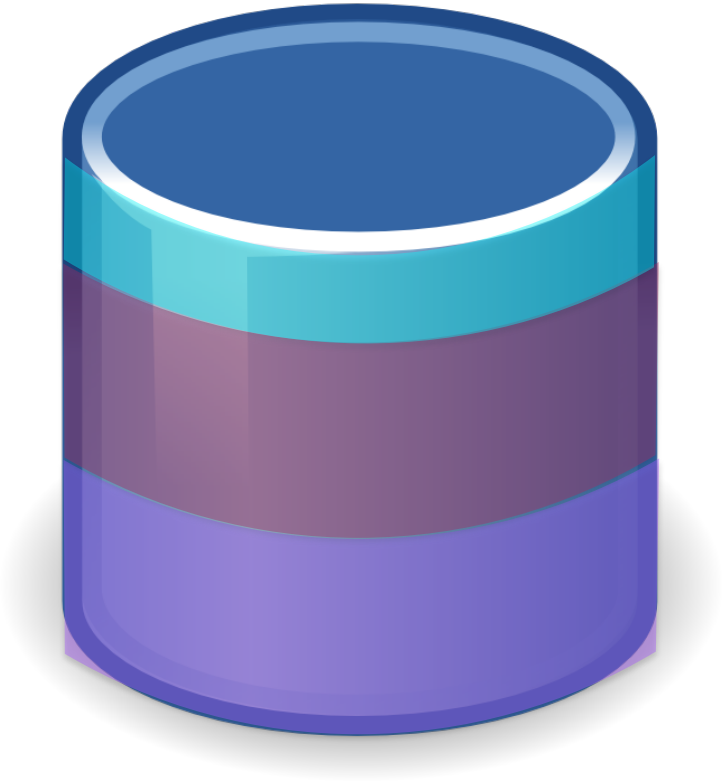
Plattenzugriffe beschleunigen - Cache



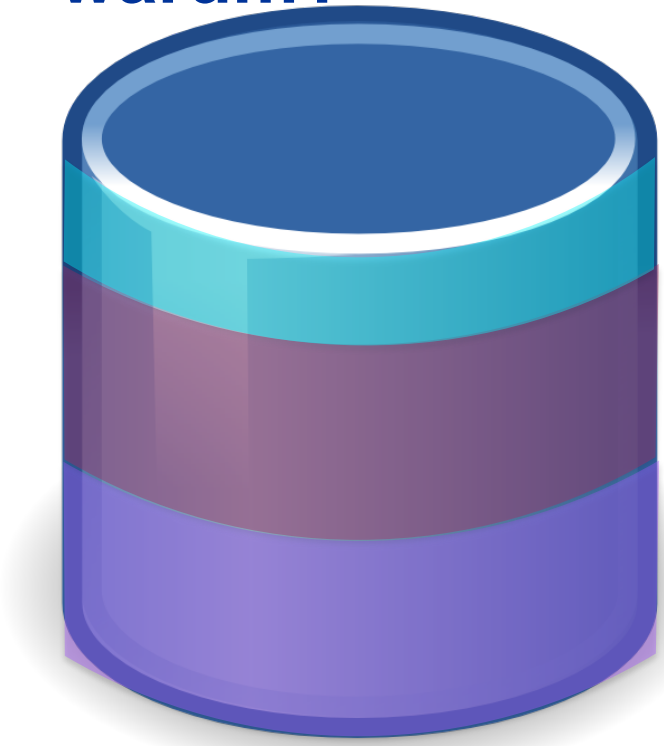
FESTPLATTEN

Partitionierung

Partitionieren



Partitionieren - warum?



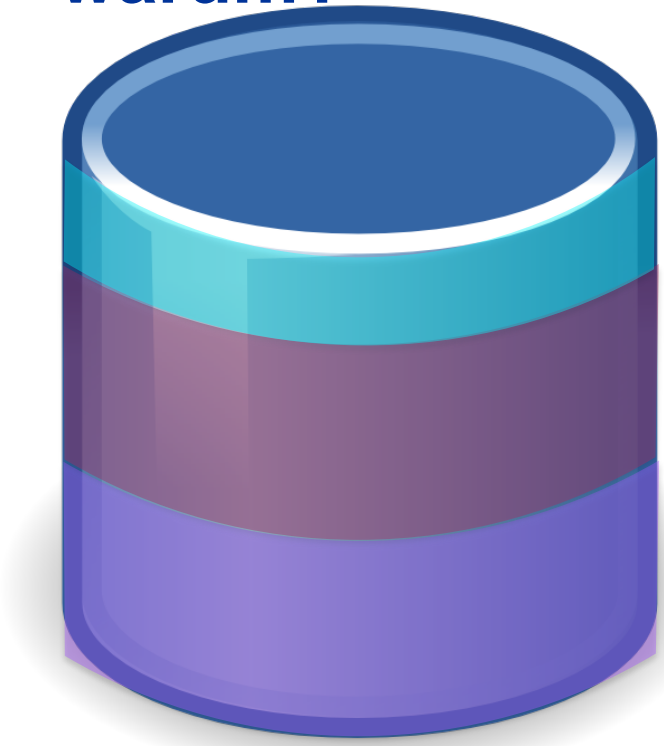
FreeBSD

Linux

Windows

verschiedene Betriebssysteme

Partitionieren - warum?



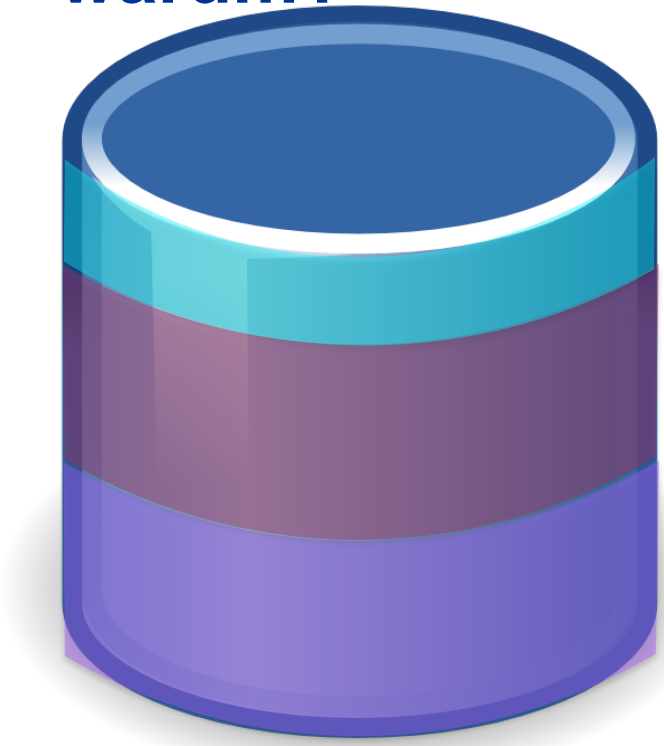
Fotos

Filme

Windows

Trennung Daten und Betriebssystem

Partitionieren - warum?



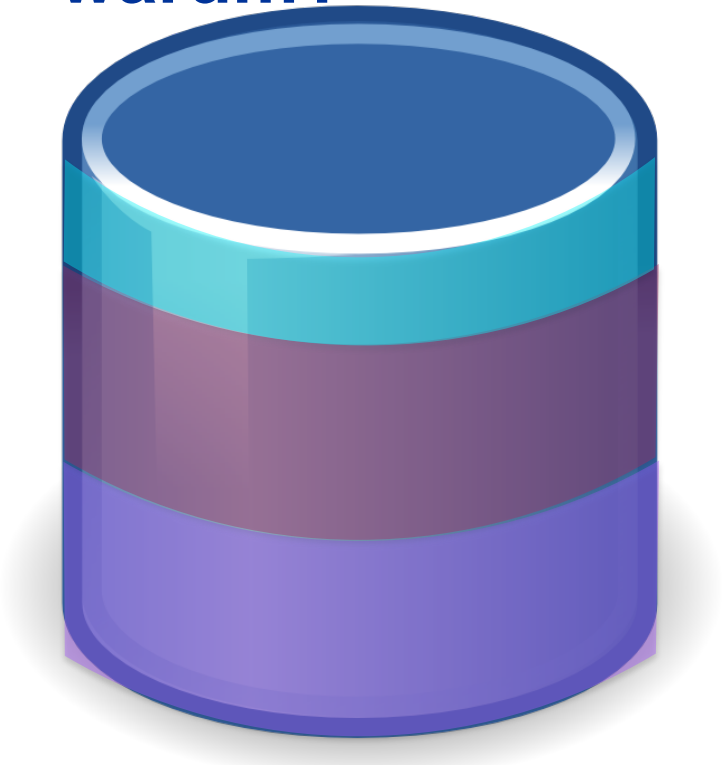
root

swap

/home

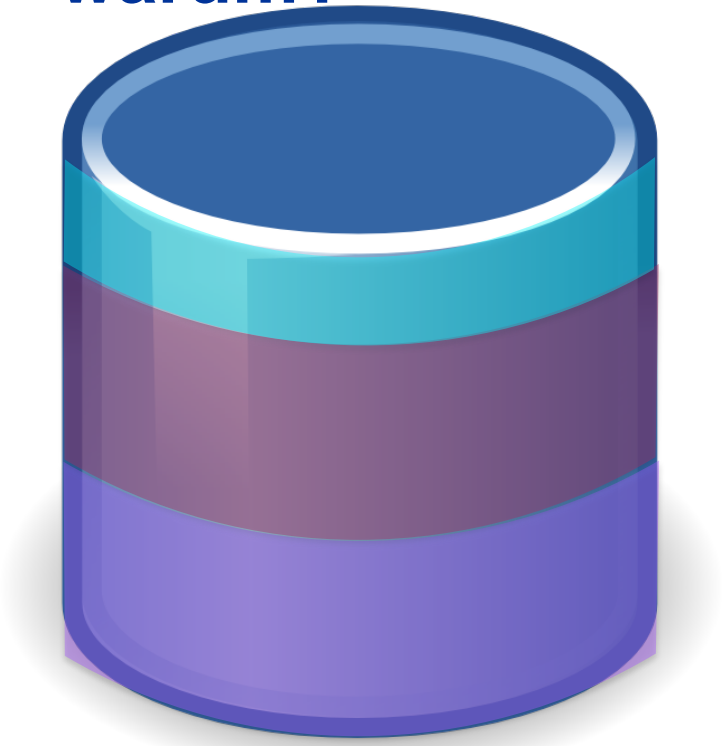
verschiedene Bereiche eines OS

Partitionieren - warum?



Arbeitskopien und Backups

Partitionieren - warum?



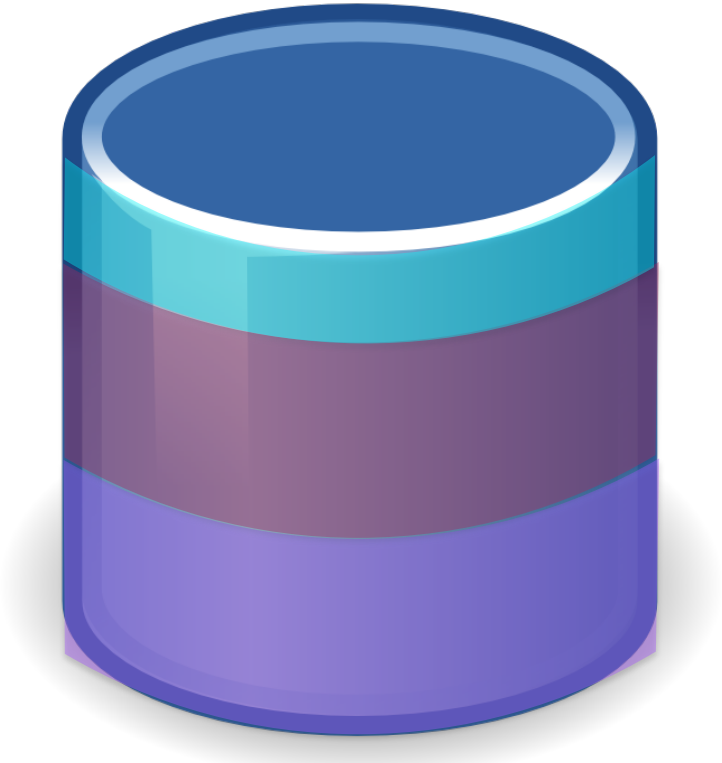
MS-DOS

Win 95a

Windows 8

Verkleinern der Platte

Partitionieren



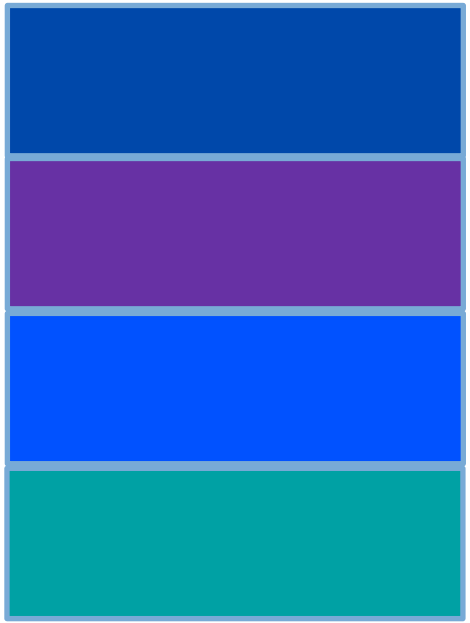
87 - NTFS

bf - Solaris

83 - Linux

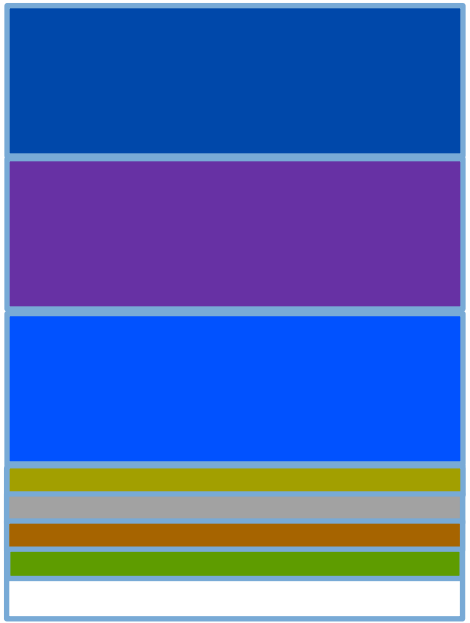
System-ID

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

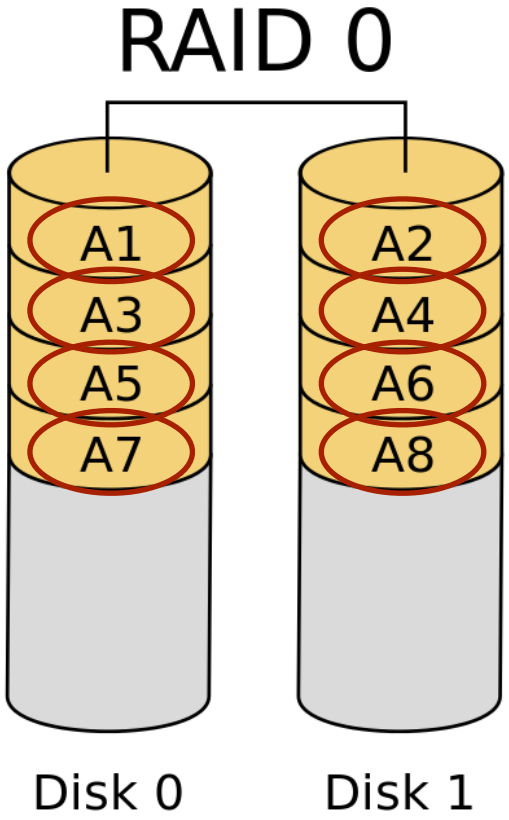
RAID -
Redundant Array of Independent Disks

Warum RAID

mehr Speicherplatz

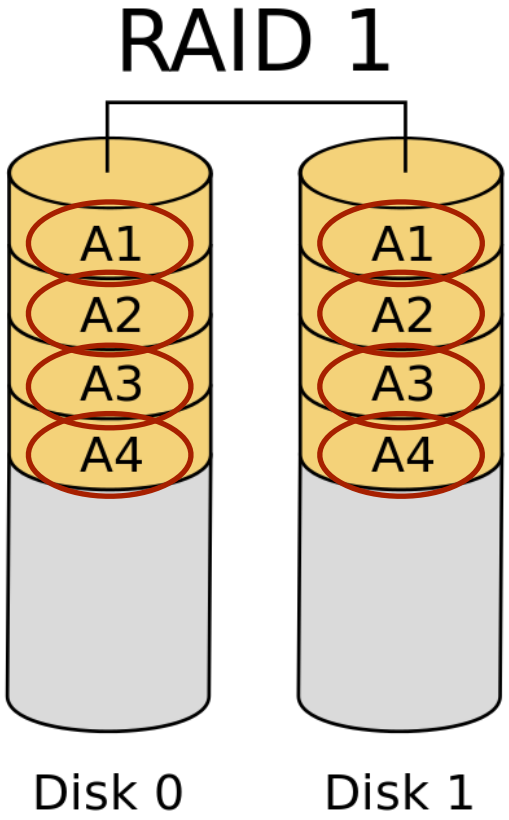
Sicherheit gegen Datenverlust*

RAID 0



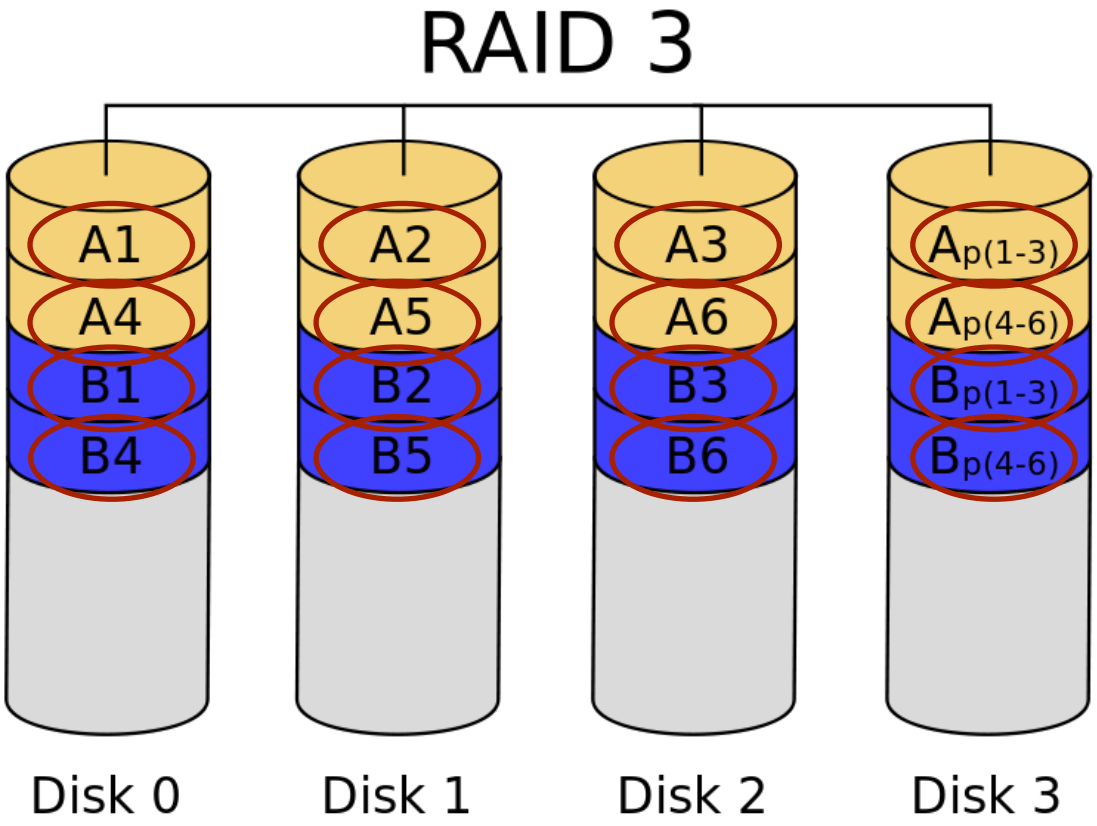
Quelle: Wikimedia

RAID 1 - Mirror



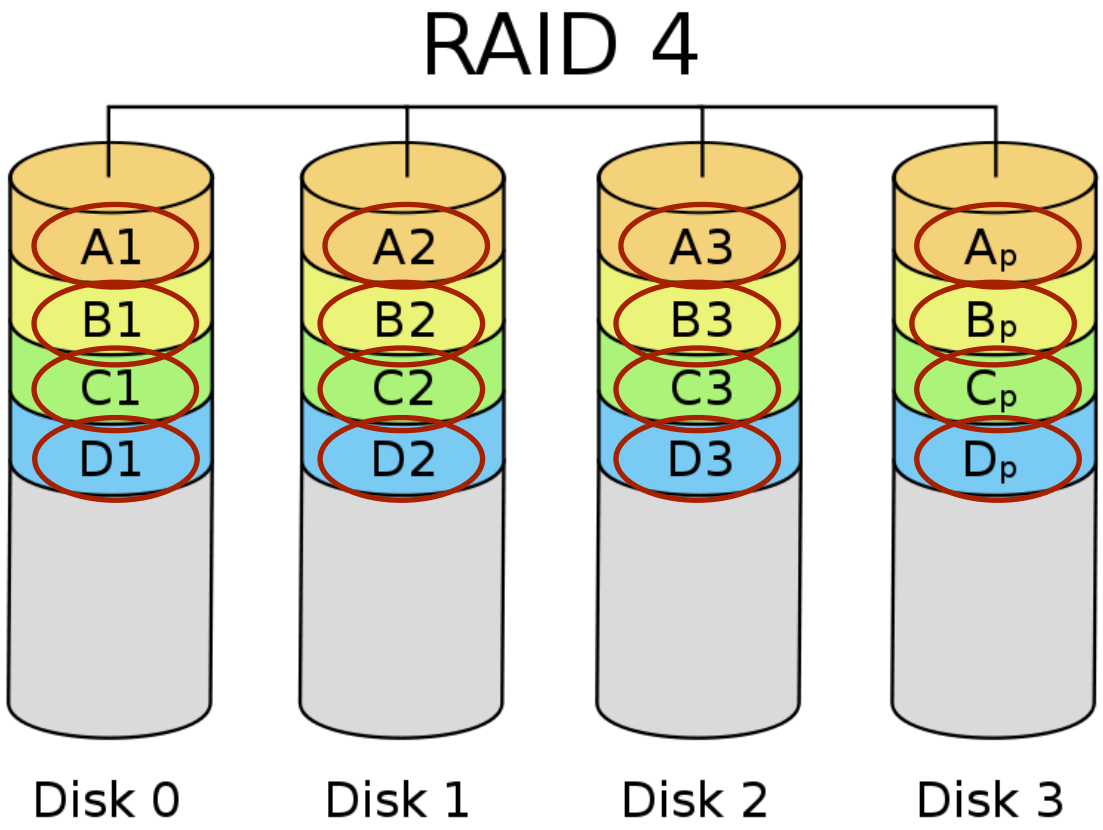
Quelle: Wikimedia

RAID 3



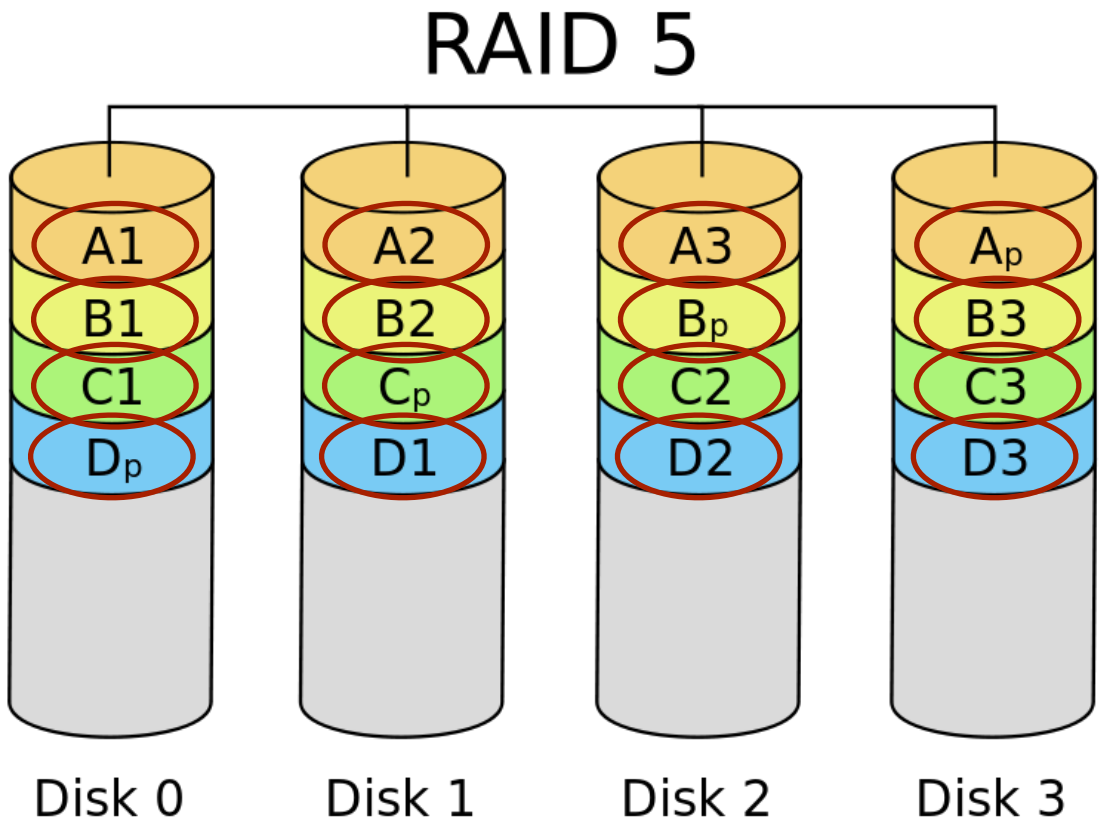
Quelle: Wikimedia

RAID 4



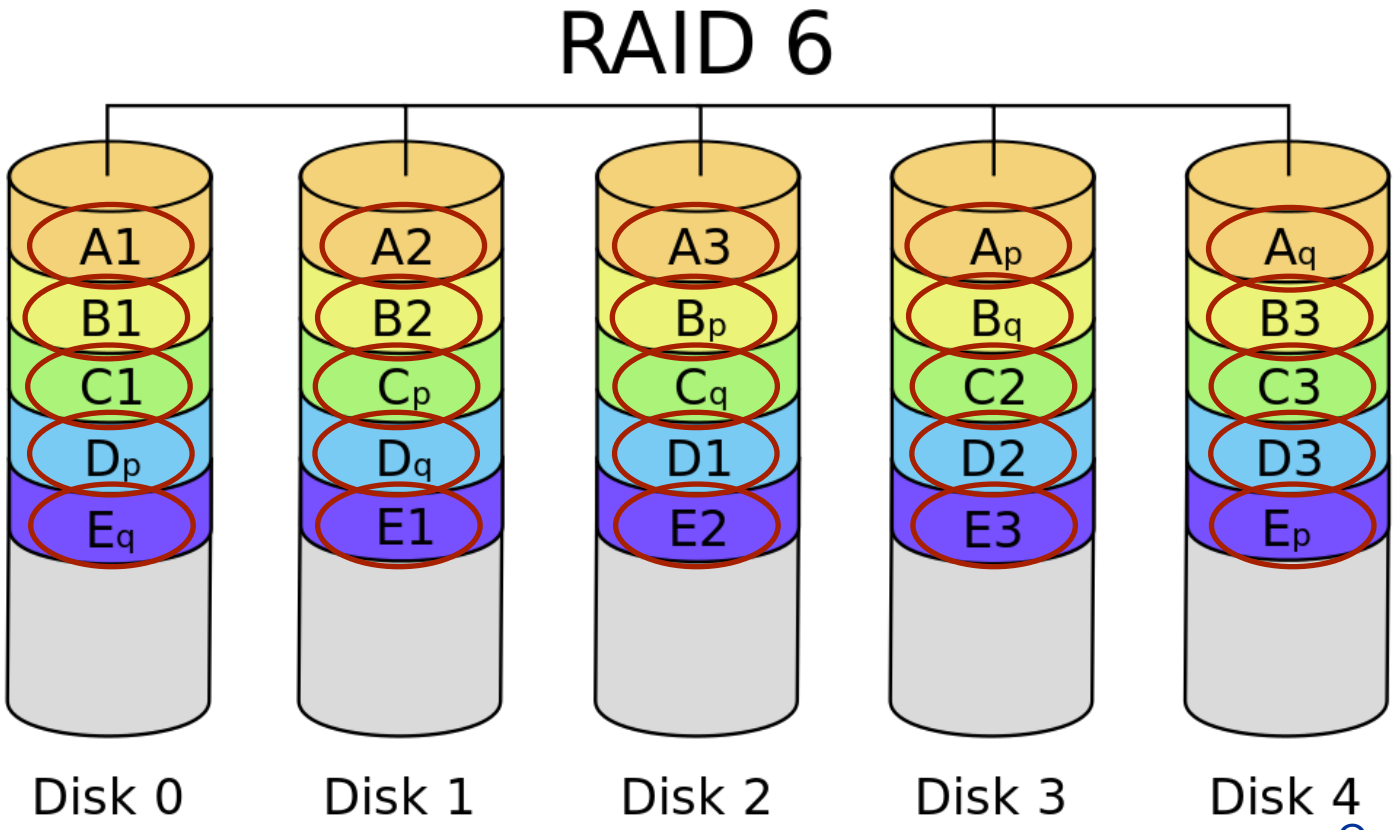
Quelle: Wikimedia

RAID 5



Quelle: Wikimedia

RAID 6



Quelle: Wikimedia

RAID 5 + HotSpare oder RAID 6?

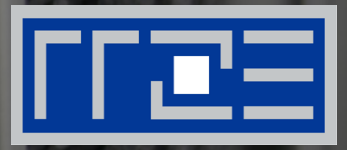
- Verschnitt an Speicherplatz ist gleich
- RAID5: Hotspare wird „geschont“
- Im Fall eines Plattendefekts:
  - RAID5 keine Redundanz (entspricht langsames Raid0)
  - Nach Einspringen der HotSpare werden alle Daten von allen verbliebenen, intakten Platte gelesen um Parity neu zu berechnen
  - Treten Lesefehler 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

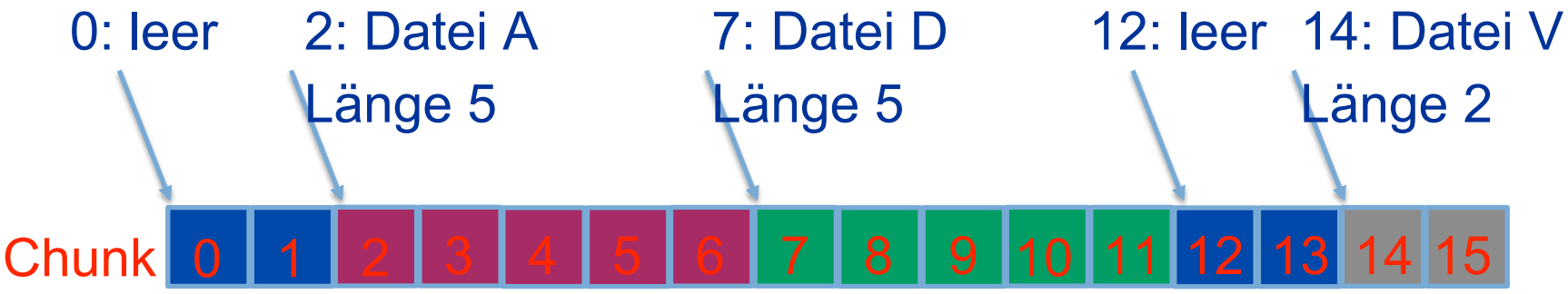
Probleme beim Speichern von Daten

Dateisysteme verwenden Cluster

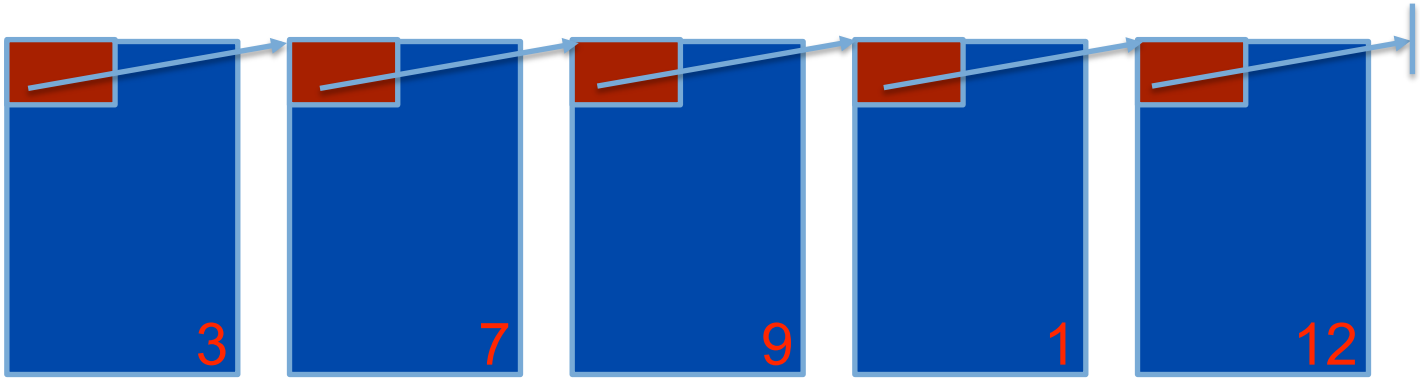
Dateien sind oft größer (oder kleiner) als ein Cluster

Wie kann man nun gespeicherte Daten adressieren?

Kontinuierliche Speicherung

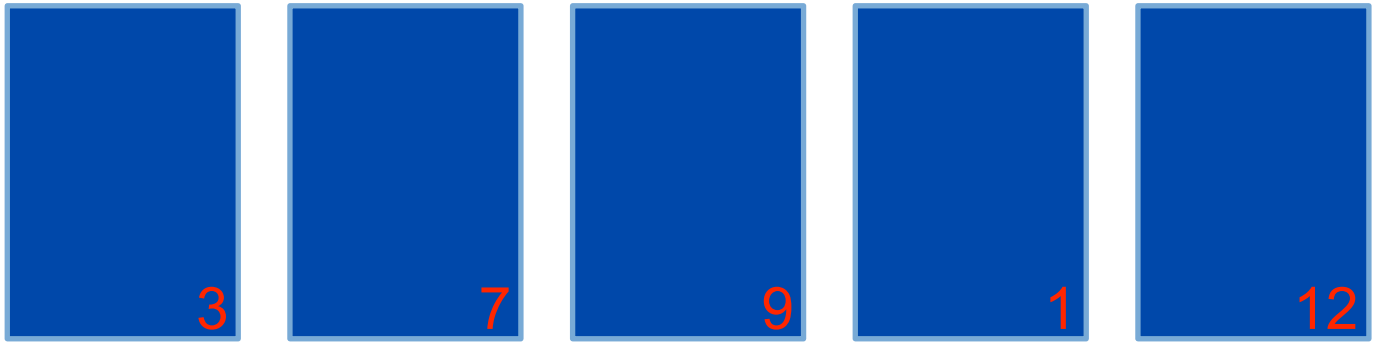


Verkettete Speicherung



Indizierte Speicherung

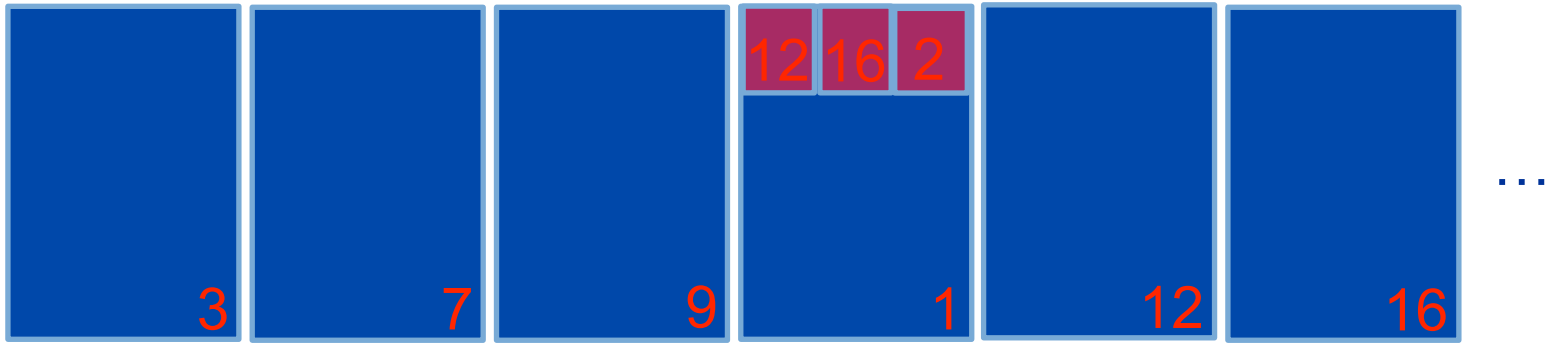
Index-Cluster



Daten-Cluster der Datei

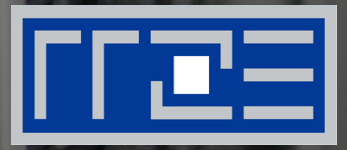
Indizierte Speicherung, mehrstufige Indizierung

Index-Cluster



Daten-Cluster mit einem zusätzlichen Index Cluster

DATEISYSTEME



Beispiele anhand gängiger Dateisysteme

FAT

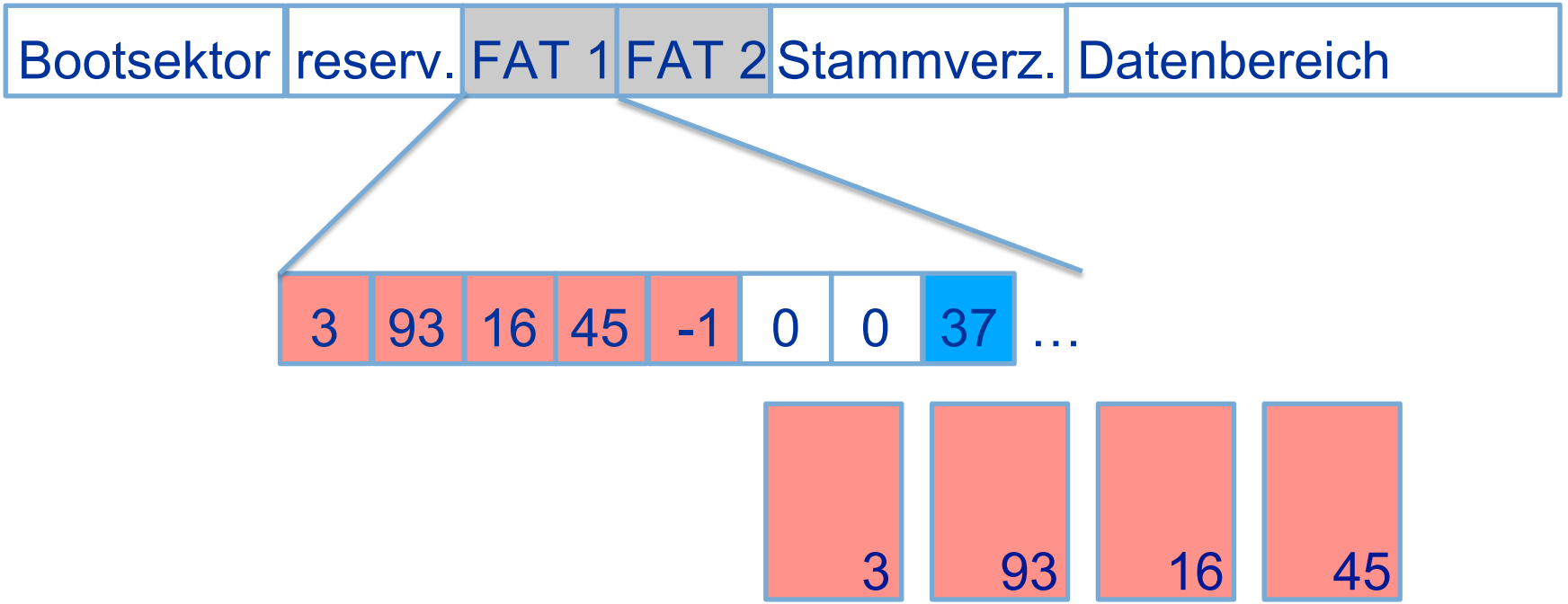
FAT

FAT

FAT

FAT

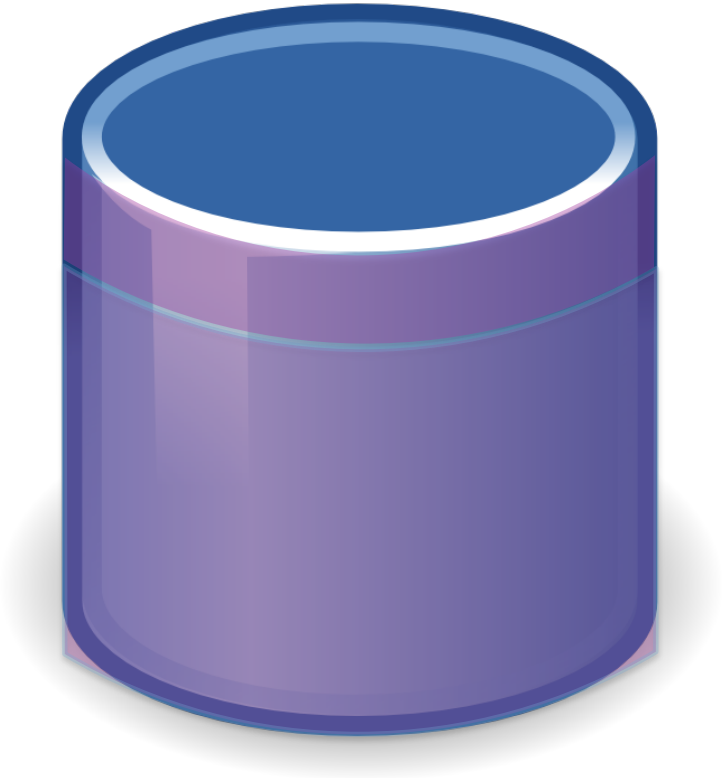
FAT


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

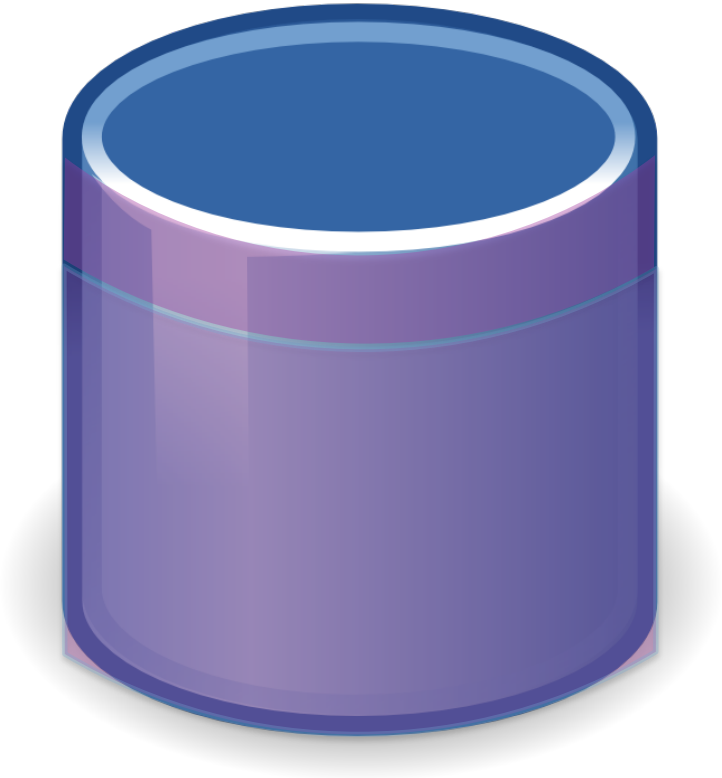
FAT


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

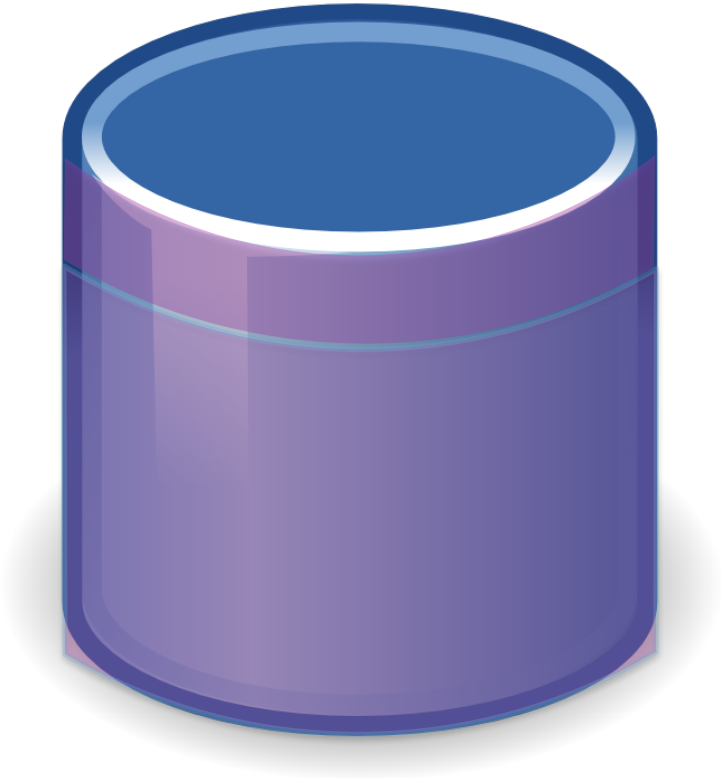
NTFS - Next Technology File System



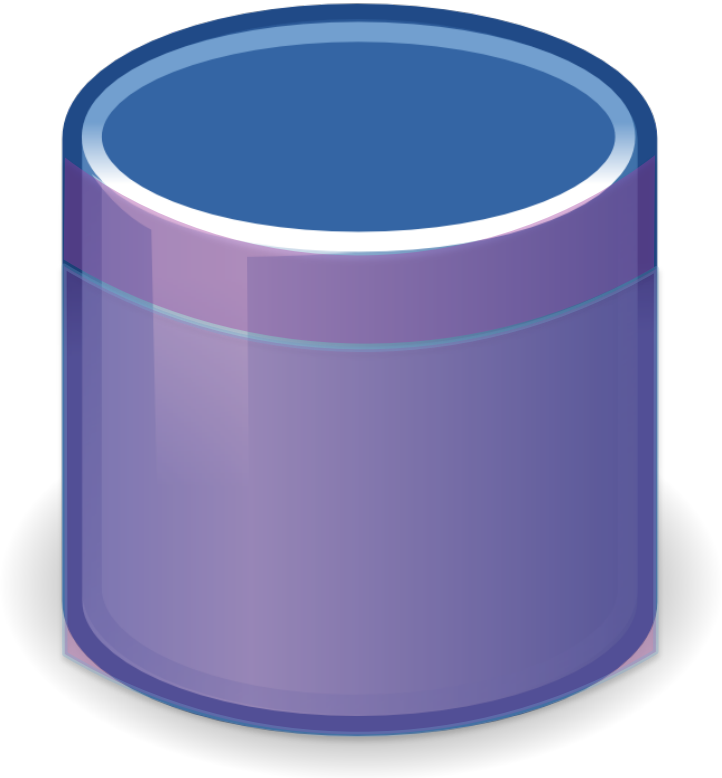
NTFS - Next Technology File System



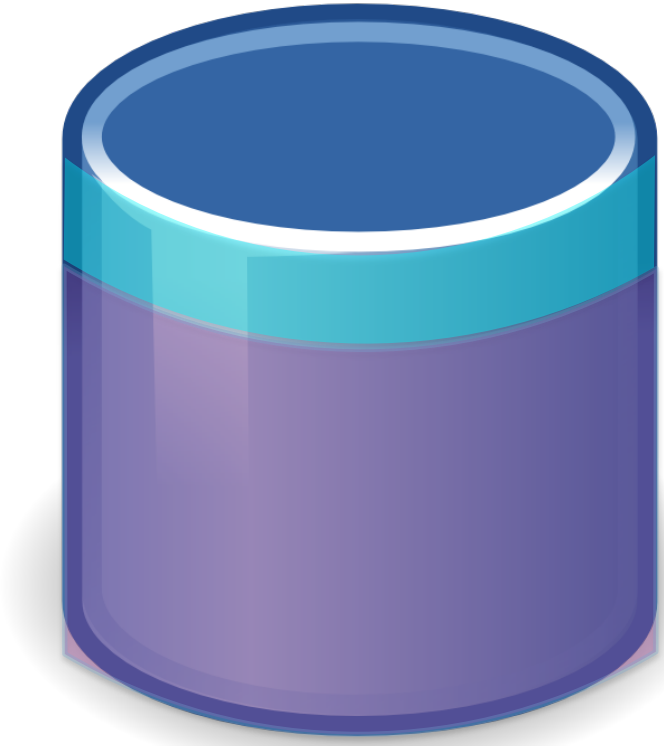
NTFS - Next Technology File System



NTFS - Next Technology File System



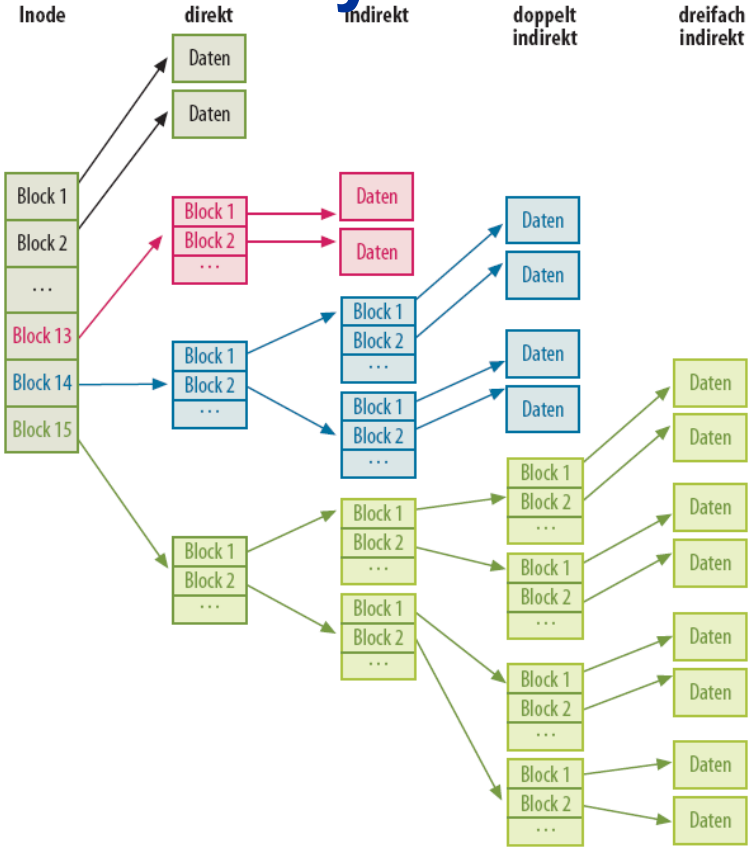
NTFS - Next Technology File System



Master File Table (12,5%)

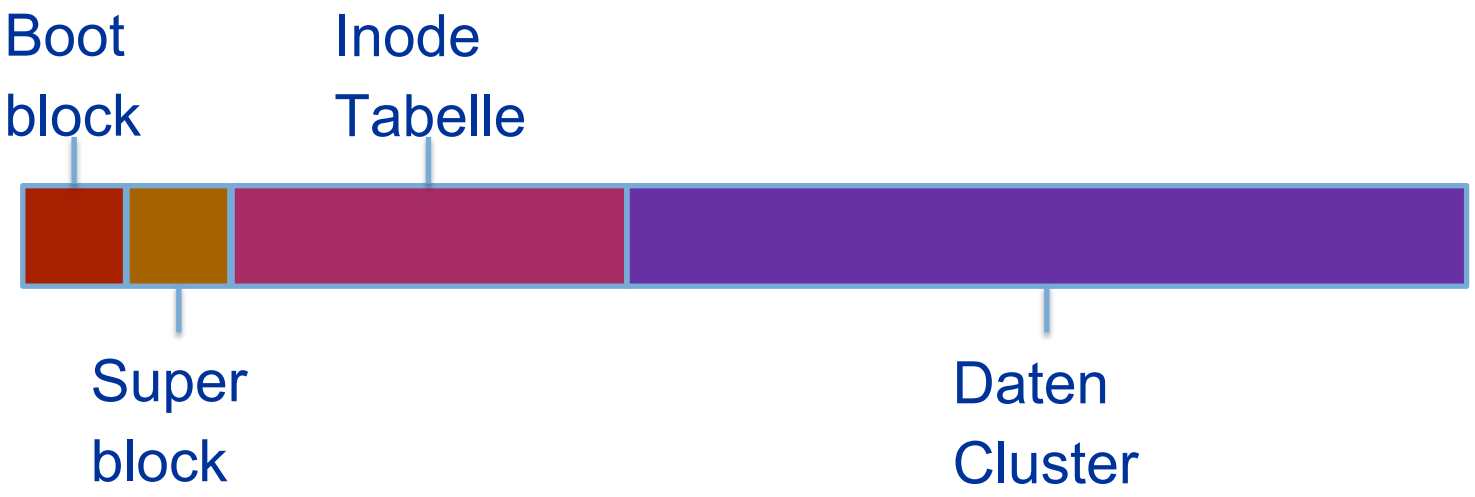
Datenbereich

Klassische Unix Dateisysteme



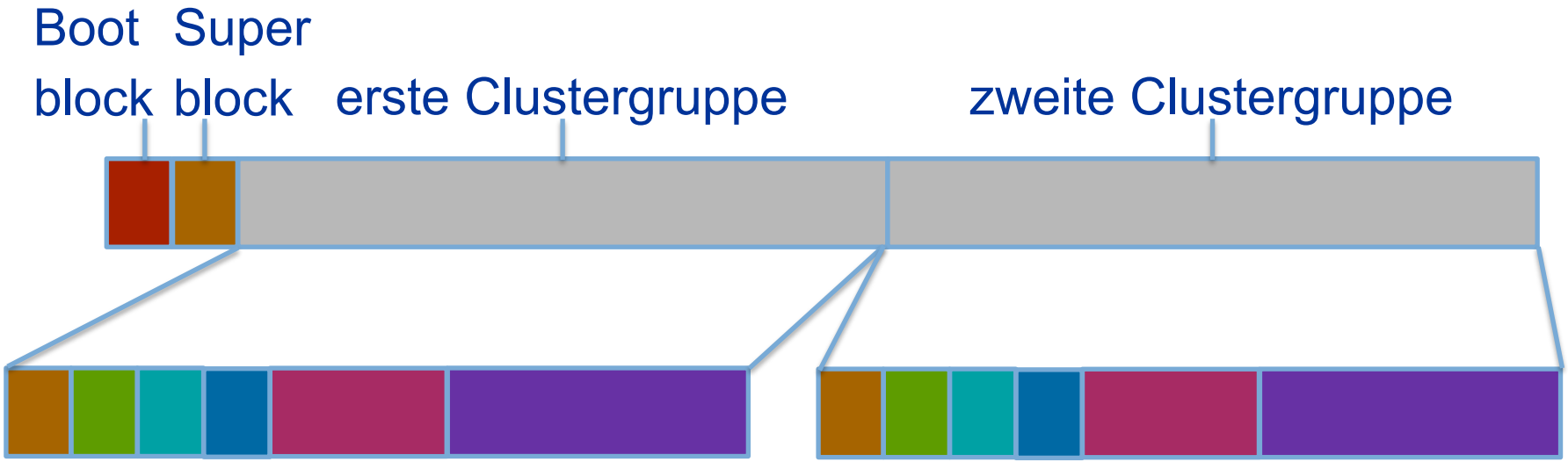
Quelle: heise.de

System V File System

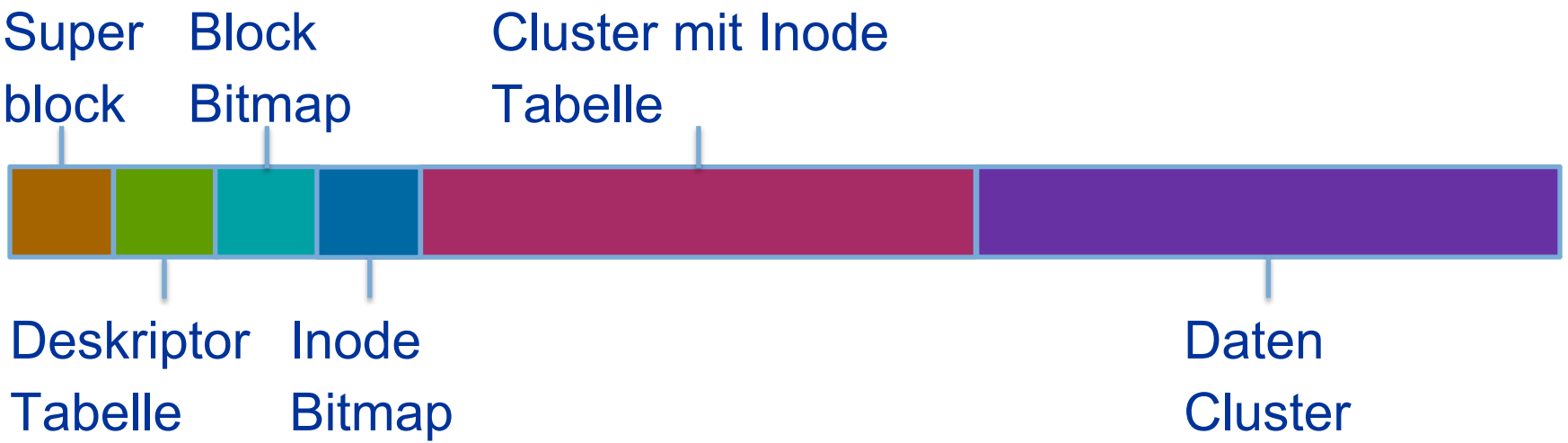


Linux ext2 / ext3 Dateisystem

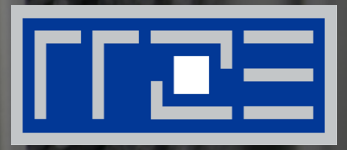
Linux ext2 / ext3 Dateisystem



Linux ext2 / ext3 Dateisystem



DATEISYSTEME



Konzepte um Datenintegrität zu garantieren

Journaling

Metadaten - Journaling

Metadaten

Daten

Vollständiges Journaling

Metadaten

Daten

Ordered - Journaling

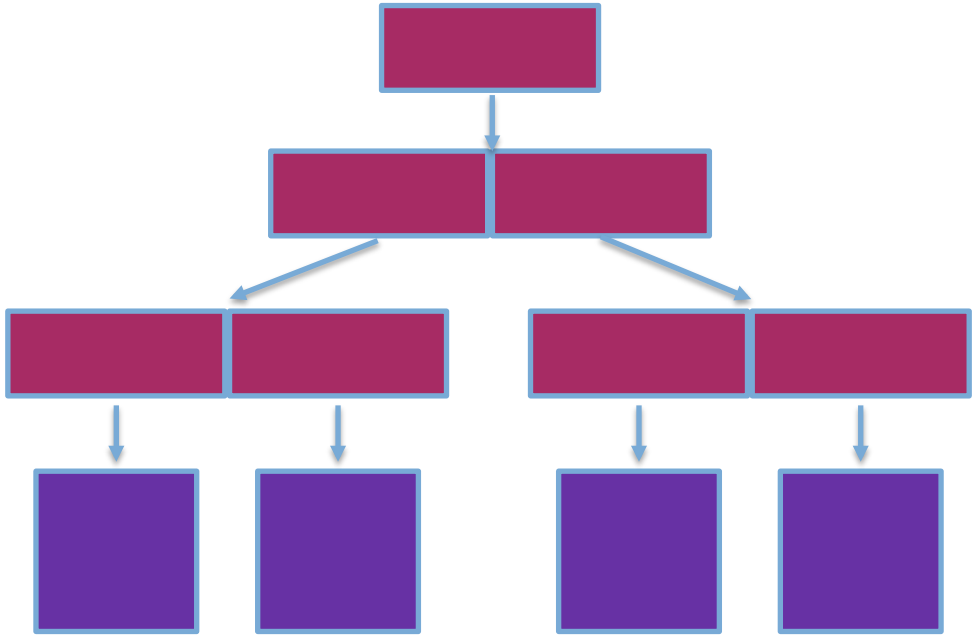
Metadaten

Daten

copy on write

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

ZFS - Beispiel für copy on write



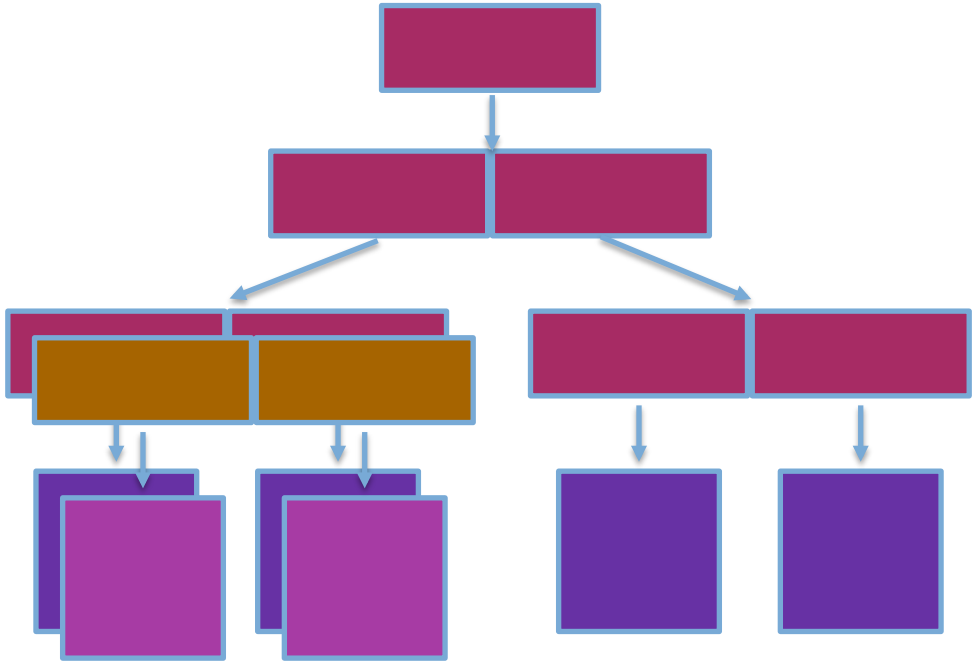
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



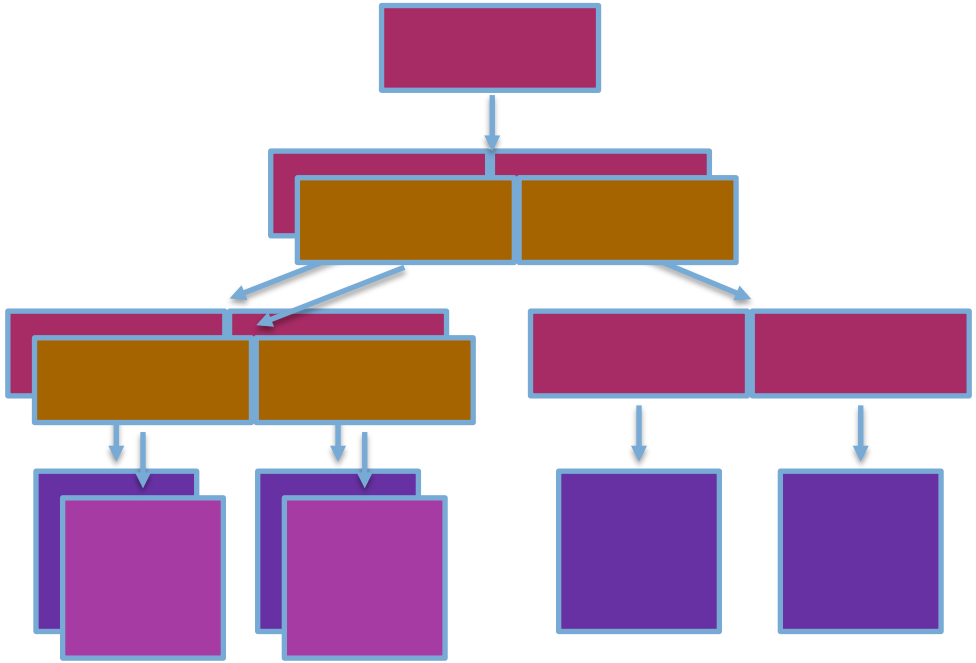
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



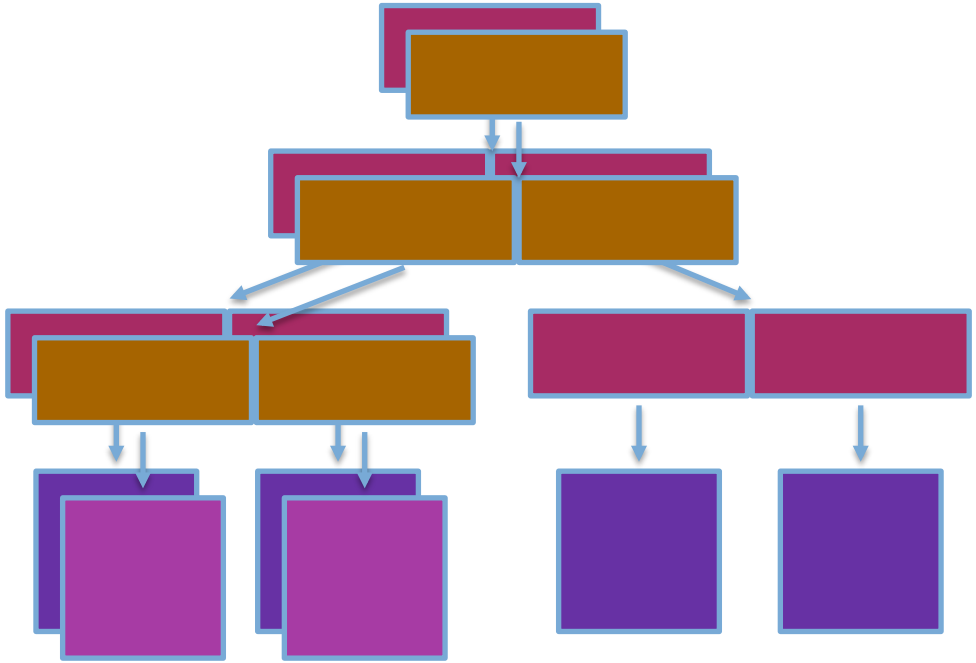
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



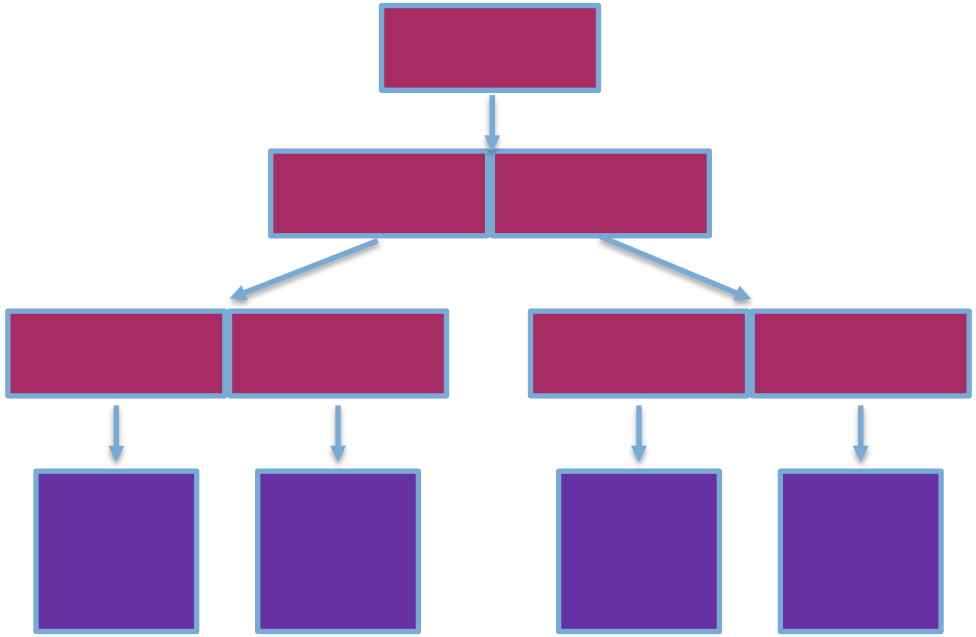
Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

Daten

ZFS - Beispiel für copy on write



Uberblock

Zeiger/Metadaten

Zeiger/Metadaten

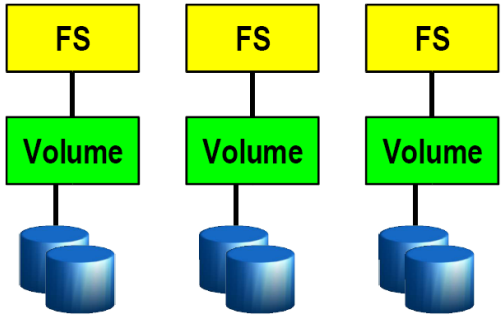
Daten

ZFS - mehr als nur ein Dateisystem

FS/Volume Modell vs. ZFS

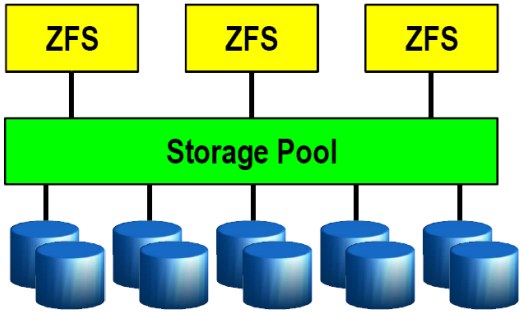
Traditionelle Volumes

- 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

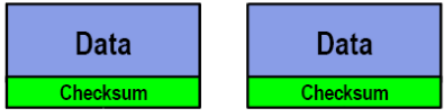


Quelle: Sun / RRZE

ZFS - Datenintegrität

Disk Block Prüfsummen

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

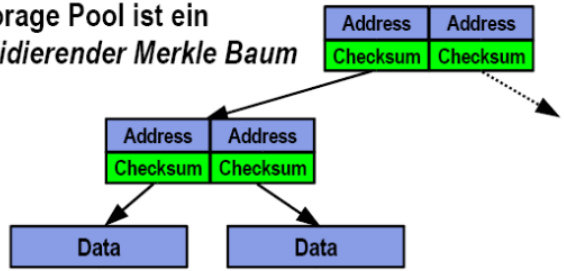


Nur Fehler auf Medium erkennbar

✔	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*



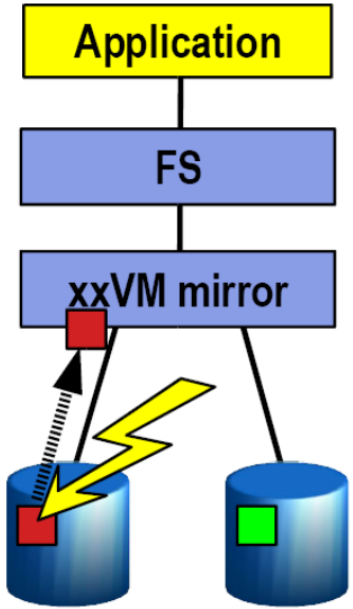
ZFS validiert alle Blöcke

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

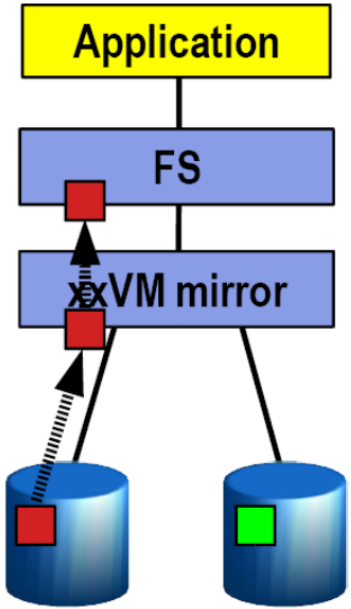
Quelle: Sun / RRZE

ZFS - Datenintegrität

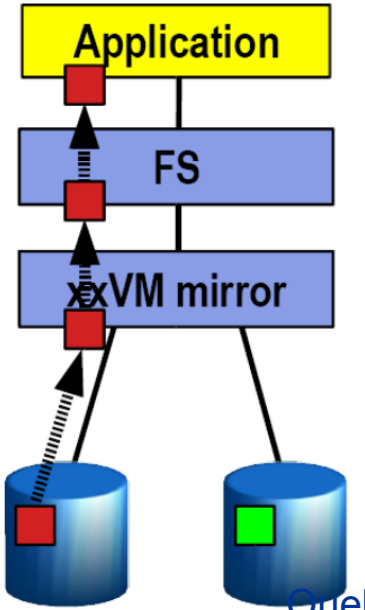
1. read liefert defekten Block



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



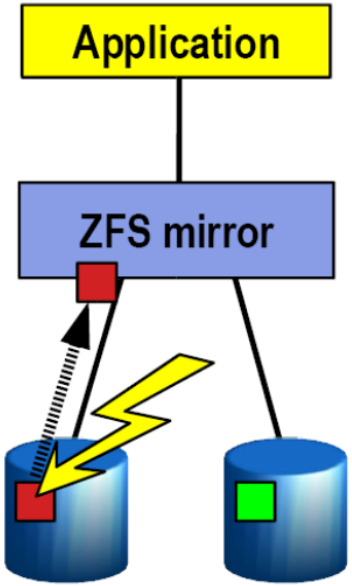
3. Falsche Daten:
Applikation bekommt Probleme
oder rechnet falsch
(ggf. unbemerkt!!!)



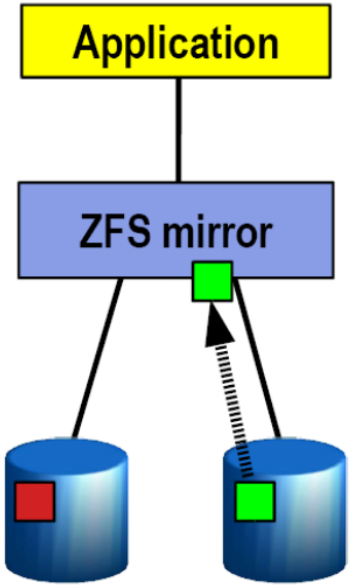
Quelle: Sun / RRZE

ZFS - Datenintegrität

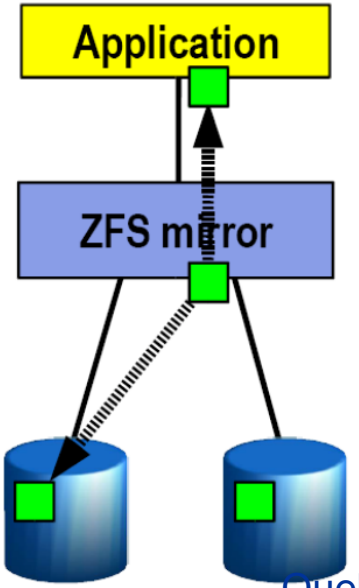
1. read liefert defekten Block



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

ZFS - weitere Features

- 128bit Dateisystem, theoretischer Adressbereich von 2^{128} (und damit 1.84×10^{19} mal mehr als z.B. btrfs)
- max Volume Size 2^{78} bytes
- max Dateigröße 2^{64} bytes
- max Anzahl an Dateien 2^{48}
- max Dateilänge 255 ASCII-Zeichen (oder entspr. weniger Unicode Zeichen)

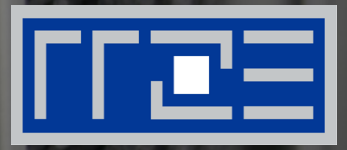
ZFS - weitere Features

- Verschlüsselung
- Komprimierung
- Snapshots (read only)
- Clone
- Online Deduplizierung (RAM intensiv!)
- NFSv4 ACLs
- NFS und SMB Freigaben (Solaris)
- snapshots send/receive ermöglicht Konzepte wie räumlich getrenntes clustering

ZFS - wo kann ich das bekommen?

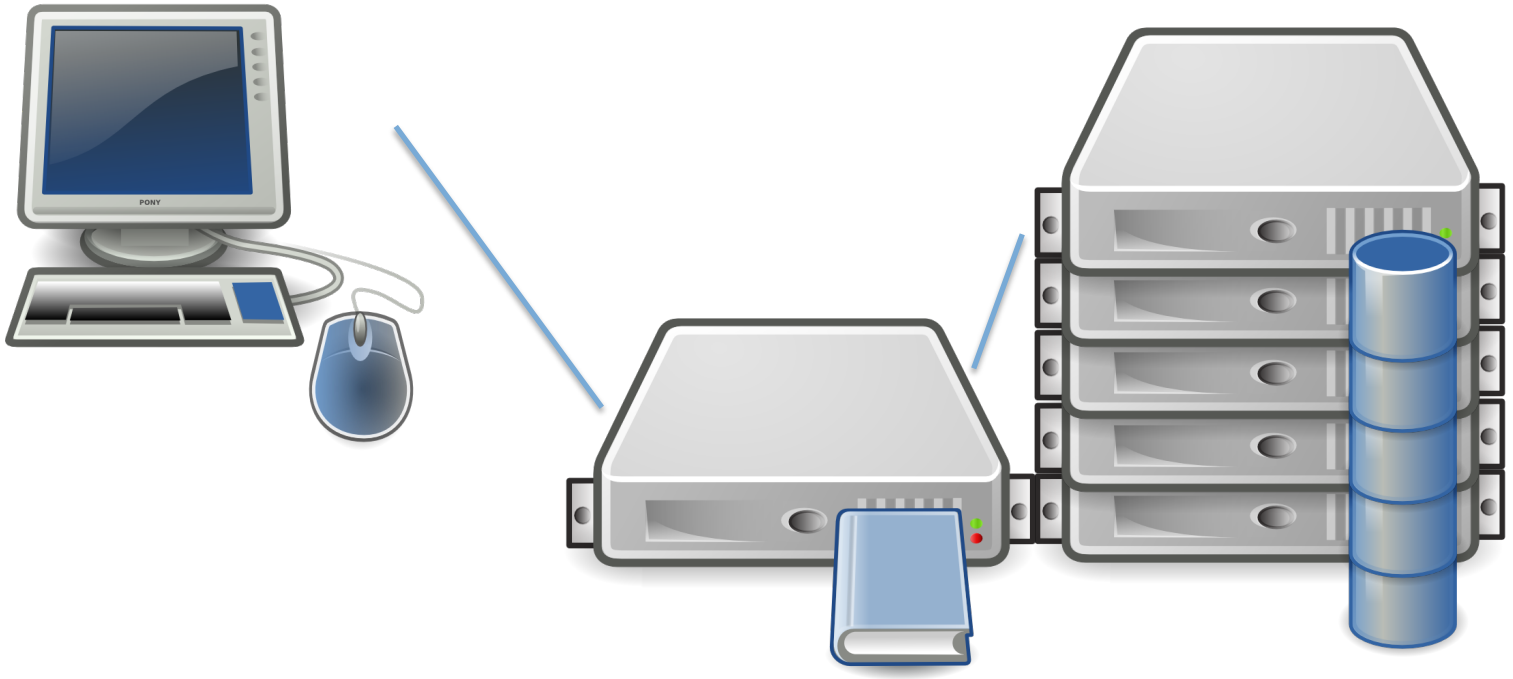
- Solaris, OpenSolaris, OpenIndiana
- versch. BSD Varianten (DragonFly BSD, NetBSD, FreeBSD, OS X, MidnightBSD, PC-BSD)
- NAS OS Distributionen wie FreeNAS, ZFS-Guru, NAS4Free, NexentaStor, EON NAS und andere
- Linux (FUSE, LLNL Implementierung, native Implementierung in Arbeit …)

DATEISYSTEME IM NETZ

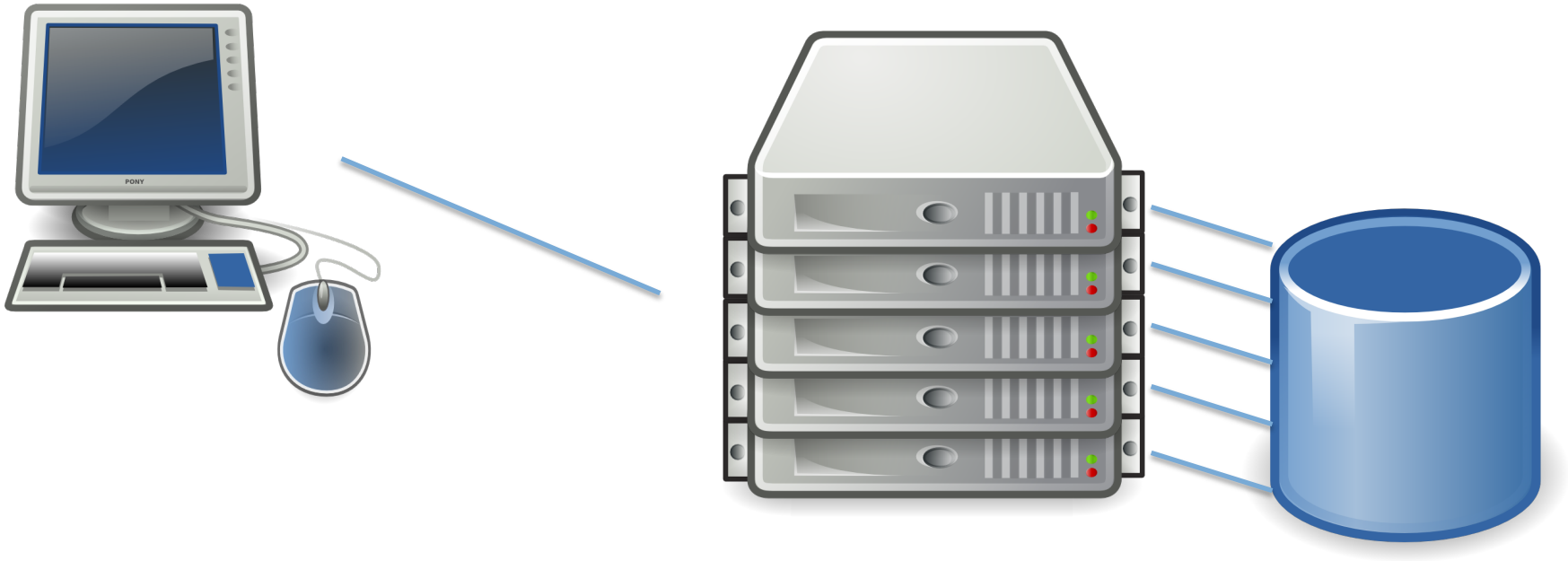


Verteilte- und Cluster- Dateisysteme

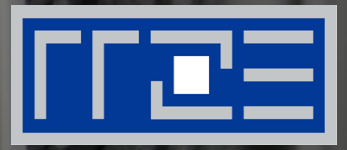
Verteilte- und Cluster- Dateisysteme



Verteilte- und Cluster- 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

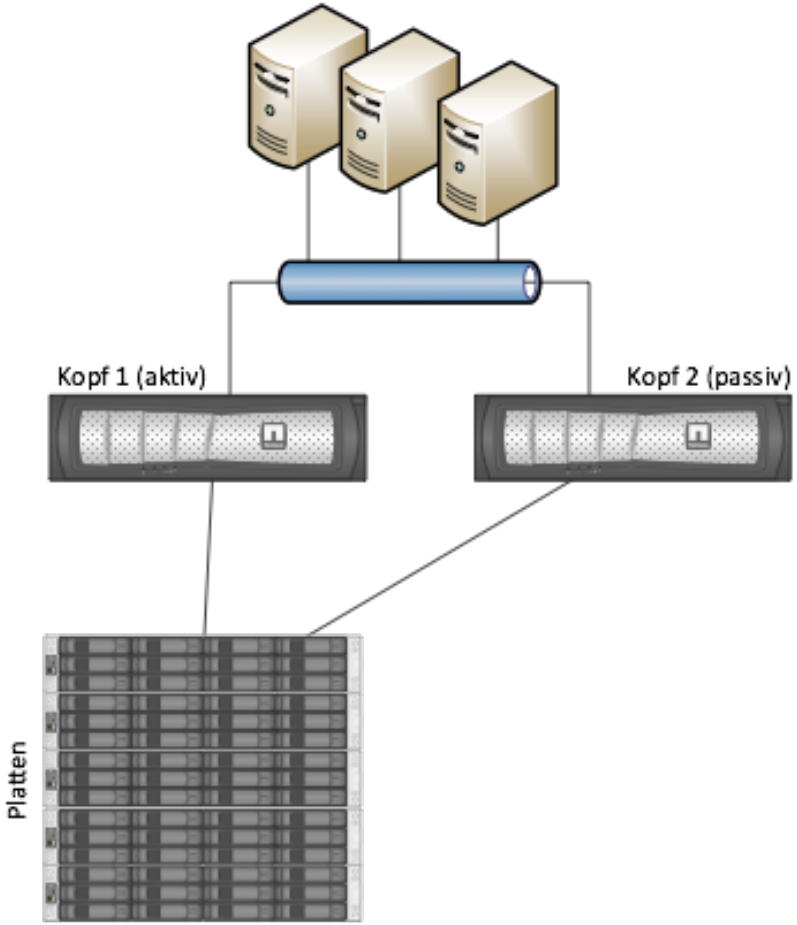
Netzwerk-Filesystem-Protokolle – NFS

- 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

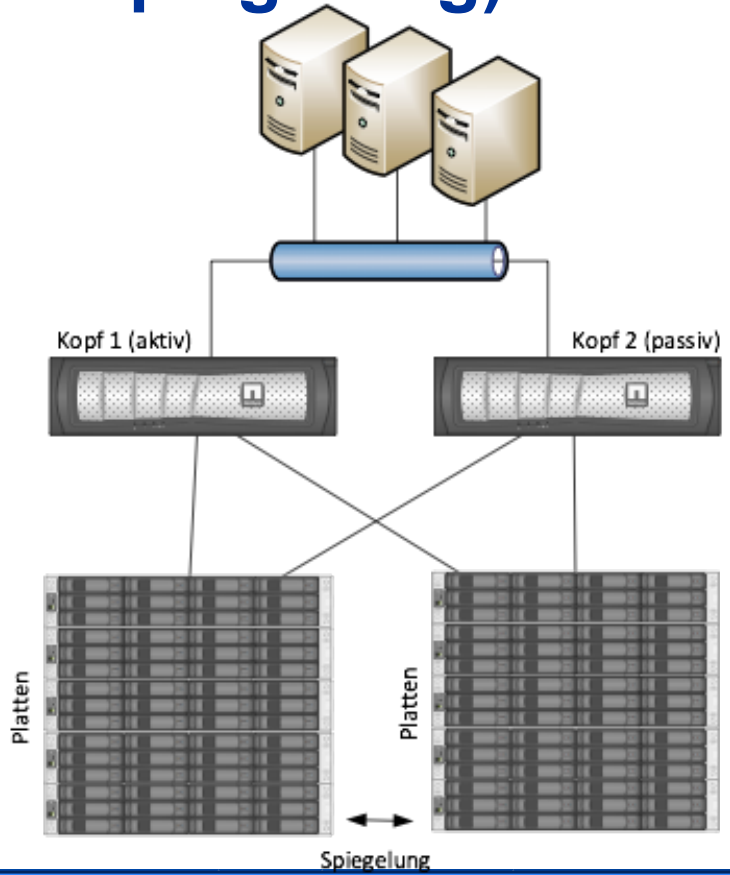
NAS - FILER

Fileserver Appliances

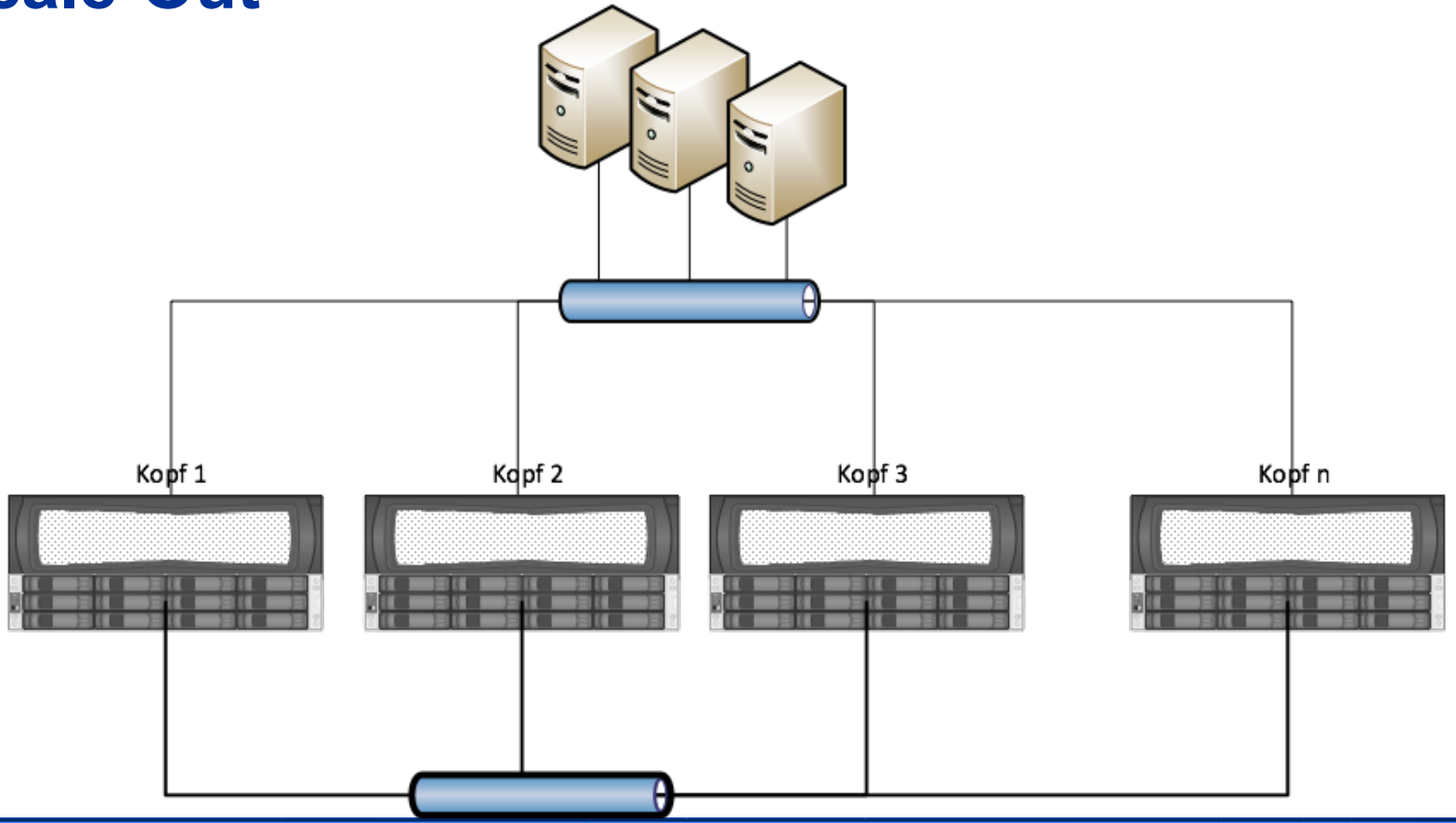
Klassisch



Klassisch (inkl. Spiegelung)



Scale-Out



Vielen Dank für Ihre Aufmerksamkeit!

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

Viel Spaß in den kommenden Wochen bei
den nächsten RRZE - Veranstaltungen!